



Manual

Models **1232E / 34E / 36E / 38E**
and **1232SE / 34SE / 36SE / 38SE**

Enhanced AC Controllers
for PMAC and Induction Motors

» **Software Version OS 37.0*** «



* 53096 Manual Rev C August 2021 includes updates to os37.49

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



Read Instructions Carefully!

Specifications are subject to change without notice.

© 2021 Curtis Instruments, Inc. © Curtis is a registered trademark of Curtis Instruments, Inc.

© The design and appearance of the products depicted herein are the copyright of Curtis Instruments, Inc.

53096, Rev D Oct. 2021

TABLE OF CONTENTS

CHAPTERS

1: INTRODUCTION	1
2: INSTALLATION AND WIRING	3
MOUNTING THE CONTROLLER	3
HIGH POWER CONNECTIONS.....	7
LOW POWER 35-PIN CONNECTIONS	9
CONTROLLER WIRING: BASIC CONFIGURATION	12
SWITCH INPUT WIRING	13
LOW POWER CIRCUIT SPECIFICATIONS	13
3: APPLICATION-SPECIFIC FEATURES	20
THROTTLE WIRING.....	20
MOTOR SPEED CONSTRAINTS	24
VOLTAGE LIMITS	25
BATTERY DISCHARGE INDICATOR.....	25
4: PROGRAMMABLE PARAMETERS	26
PROGRAMMING MENUS.....	26
CLONING CONTROLLERS	75
5: MONITOR MENU	76
6: CONTROLLER INFORMATION MENU	86
7: INITIAL SETUP	87
8A: AUTOMATED ACIM MOTOR CHARACTERIZATION PROCEDURE	93
PART 1: TRACTION AND HYDRAULIC SYSTEMS.....	93
PART 2A: TRACTION SYSTEMS ONLY	94
PART 2B: HYDRAULIC SYSTEMS ONLY.....	97
8B: AUTOMATED PMAC MOTOR COMMISSIONING PROCEDURE.....	99
PMAC CONSIDERATIONS.....	99
9: TUNING GUIDE	102
0 – SPEED MODE EXPRESS TUNING	102
1 – SPEED MODE TUNING	103
2 – TORQUE MODE TUNING	105

TABLE OF CONTENTS cont'd

10: VEHICLE CONTROL LANGUAGE (VCL).....	106
VARIABLE TYPES.....	107
VCL RUNTIME RATES	109
I/O CONTROL WITH VCL	110
INTERFACING THE THROTTLE AND BRAKE COMMANDS.....	112
INTERFACING THE PROPORTIONAL CURRENT DRIVER	118
USING THE FAULT HANDLER IN VCL.....	119
CANBUS EMERGENCY MESSAGES	121
OEM-DEFINED USER FAULTS	122
CAN INDEXES FOR THE USER FAULT VARIABLES	125
VCL FUNCTIONS SPECIFIC TO 1232E/SE, 1234E/SE, 1236E/SE, & 1238E/SE CONTROLLERS	126
11: DIAGNOSTICS AND TROUBLESHOOTING.....	142
DIAGNOSTICS	142
TROUBLESHOOTING	143
12: MAINTENANCE	154
CLEANING.....	154
FAULT HISTORY.....	154
APPENDIX A: VEHICLE DESIGN CONSIDERATIONS.....	155
ELECTROMAGNETIC COMPATIBILITY (EMC)	155
ELECTROSTATIC DISCHARGE (ESD) IMMUNITY	156
DECOMMISSIONING AND RECYCLING THE CONTROLLER.....	156
APPENDIX B: EN13849 COMPLIANCE.....	157
APPENDIX C: PROGRAMMING DEVICES	159
PROGRAMMING DEVICES.....	159
APPENDIX D: CONTROLLER SPECIFICATIONS	160



TABLE OF CONTENTS cont'd

TABLES

TABLE 1: HIGH POWERED CONNECTIONS	7
TABLE 2: LOW POWER CONNECTIONS	10
TABLE 3: PROGRAMMABLE PARAMETERS MENUS: 1313/1314 PROGRAMMER	27
TABLE 4: MONITOR MENU: 1313/1314 PROGRAMMER	76
TABLE 5: TYPES OF LED DISPLAY	143
TABLE 6: TROUBLESHOOTING CHART	146
TABLE D-1: SPECIFICATIONS: 1232E/SE, 1234E/SE, 1236E/SE, 1238E/SE CONTROLLERS ¹	160

FIGURES

FIGURE 1: CURTIS AC INDUCTION MOTOR CONTROLLERS:	1
FIGURE 2A: MOUNTING DIMENSIONS, 1232E AND 1232SE.....	3
FIGURE 2B: MOUNTING DIMENSIONS, 1234E AND 1234SE.....	4
FIGURE 2C: MOUNTING DIMENSIONS, 1236E/SE AND 1238E/SE.	5
FIGURE 3: BASIC WIRING DIAGRAM	12
FIGURE 4: WIRING FOR TYPE 1 THROTTLES.....	20
FIGURE 5: WIRING FOR TYPE 2 THROTTLES.....	22
FIGURE 6: WIRING FOR TYPE 3 THROTTLES.....	23
FIGURE 7: ACCELERATION RESPONSE RATE DIAGRAM.....	36
FIGURE 8: BRAKING RESPONSE RATE DIAGRAM	37
FIGURE 9: THROTTLE MAPPING (TORQUE CONTROL MODE).	43
FIGURE 10: EFFECT OF GEAR SOFTEN PARAMETER (TORQUE CONTROL MODE).	43
FIGURE 11: EFFECT OF BRAKE TAPER SPEED PARAMETER (TORQUE CONTROL MODE).....	43
FIGURE 12: DRIVE CURRENT LIMITING MAP.....	45
FIGURE 13: REGEN CURRENT LIMITING MAP.....	46
FIGURE 14: EFFECT OF THROTTLE ADJUSTMENT PARAMETERS.....	46
FIGURE 15: MOTOR COMMAND DIAGRAM.....	114
FIGURE 16: CONTROL MODE PROCESSING.....	117
FIGURE 17: PROPORTIONAL DRIVER PROCESSING.....	118
FIGURE B-1: SUPERVISORY SYSTEM.	157

1 – INTRODUCTION

Curtis 1232E/SE, 1234E/SE, 1236E/SE, and 1238E/SE AC motor controllers provide accurate, dependable, and highly efficient control of speed and torque of AC induction motors (ACIM) and permanent magnet AC motors (PMAC).

These AC controllers contain two microprocessors to provide exceptional capability and functional safety. The primary microprocessor runs an advanced field-oriented AC motor control while simultaneously running VCL software in an embedded logic controller. The second microprocessor continuously monitors the operation of the system, redundantly measuring inputs, crosschecking results, and verifying critical timing and operations.

VCL (Vehicle Control Language) is an innovative software programming language developed by Curtis. Many electric vehicle functions are uniquely built into the VCL code, and additional functions can be created by OEMs as required. VCL opens new avenues of customization, allowing specific vehicle application functions to be created quickly and easily within the motor controller itself, often eliminating the need to use separate vehicle manager modules.

The CANbus communications included within these controllers allow them to function as either the system CAN manager or as a CAN ancillary device. Inputs and outputs can be optimally shared throughout the system, minimizing wiring and creating integrated functions that often reduce the cost of the system.

These controllers are the ideal solution for traction, hoist, dual drive, and other motor drive and vehicle control needs.



Figure 1

Curtis AC induction motor controllers:

from left to right, models 1232SE, 1234E, 1236E, and 1238E.

The E and SE models look similar, and share the same standard features.

Like all Curtis controllers, the E and SE models offer superior operator control of motor drive performance. Features include:

- Closed-loop speed and torque control for induction (ACIM) motors.
- Full field-weakening capability with ACIM motors.
- Built-in auto-characterization routines for effective in-vehicle optimization of motor performance and efficiency.
- Powerful operating system allows parallel processing of vehicle control tasks, motor control tasks, and user configurable programmable logic (VCL).
- Built-in Dual Drive software allows easy setup and control of typical dual-drive vehicles, without VCL.
- CANopen compatible CANbus connection; other CANbus protocols are configurable through VCL.
- Easily programmable through the Curtis 1313 handheld programmer and 1314 PC Programming Station.
- Rugged sealed housing and connectors meet IP65 environmental sealing standards for use in harsh environments.
- Compliance with Machinery Directives 2006/42/EC and EN13849-1.

What is New in os37.49

OS37.49 updates the available ACIM Motor Types to 536. It updates the PMAC Motor Types to 127. This Revision D clarifies the fact that the 1232E/1232/SE controllers do have the analog output.

os37.49 includes the updates for os37.0:

The Digital Out 6 Open/Short and Digital Out 7 Open/Short were updated to Digital Out 6 Overcurrent and Digital Out 7 Overcurrent to more accurately describe the feature as the open driver checks were not performed. The current limit remains at 1 Amp. This name change also applies to the VCL_BIT definitions,

- DIGITAL_OUT_6_OPEN_SHORT_FAULT = DIGITAL_OUT_6_OVERCURRENT_FAULT.
- DIGITAL_OUT_7_OPEN_SHORT_FAULT = DIGITAL_OUT_7_OVERCURRENT_FAULT.

The VCL_CONST definitions were also updated,

- DIGITAL_OUT_6_OPEN_SHORT = DIGITAL_OUT_6_OVERCURRENT.
- DIGITAL_OUT_7_OPEN_SHORT = DIGITAL_OUT_7_OVERCURRENT.

Getting the most out of the Curtis controller

Thoroughly read and refer to this manual to apply and configure the controller. Understanding the installation & wiring guidelines, the parameter settings, the VCL functions, the initial setup & commissioning, and using the diagnostic and troubleshooting guide are critical to a successful application.

This manual describes the typical usage of the controller. If the intended application is not illustrated or discussed within this manual, contact the Curtis distributor or the regional Curtis sales-support office for additional technical support. The Curtis sales and support office-contact information is available from the Curtis Instruments website under the International tab: <https://www.curtisinstruments.com/international/>

2 — INSTALLATION AND WIRING

MOUNTING THE CONTROLLER

The outline and mounting hole dimensions for the 1232E/SE controller are shown in Figure 2a, for the 1234E/SE controller in Figure 2b, and for the 1236E/SE and 1238E/SE controllers in Figure 2c. When an Ampseal plug housing is mated with the 35-pin logic receptacle, these controllers meet the IP65 requirements for environmental protection against dust and water. Nevertheless, in order to prevent external corrosion and leakage paths from developing, **the mounting location should be carefully chosen to keep the controller as clean and dry as possible.**

Mount the controller to a flat surface devoid of protrusions, ridges, or a curvature that can cause damage or distortion to its heatsink (the base plate). Secure the controller using four 6 mm (1/4") diameter bolts evenly torqued to the vehicle's mounting surface. These controller's heatsink (bottom surface) have a typical roughness grade of N8 (ISO 1302), with a flatness tolerance of < 5 mm (0.13 per 25 mm). A thermal joint compound is recommended to improve heat conduction from the controller heatsink to the vehicle's mounting surface. Typically, when properly mounted to a larger metal surface, additional heat-sinking or fan-cooling is not necessary to meet the application's peak and continuous current ratings.

Figure 2a

*Mounting dimensions,
Curtis 1232E and 1232SE
motor controllers.*

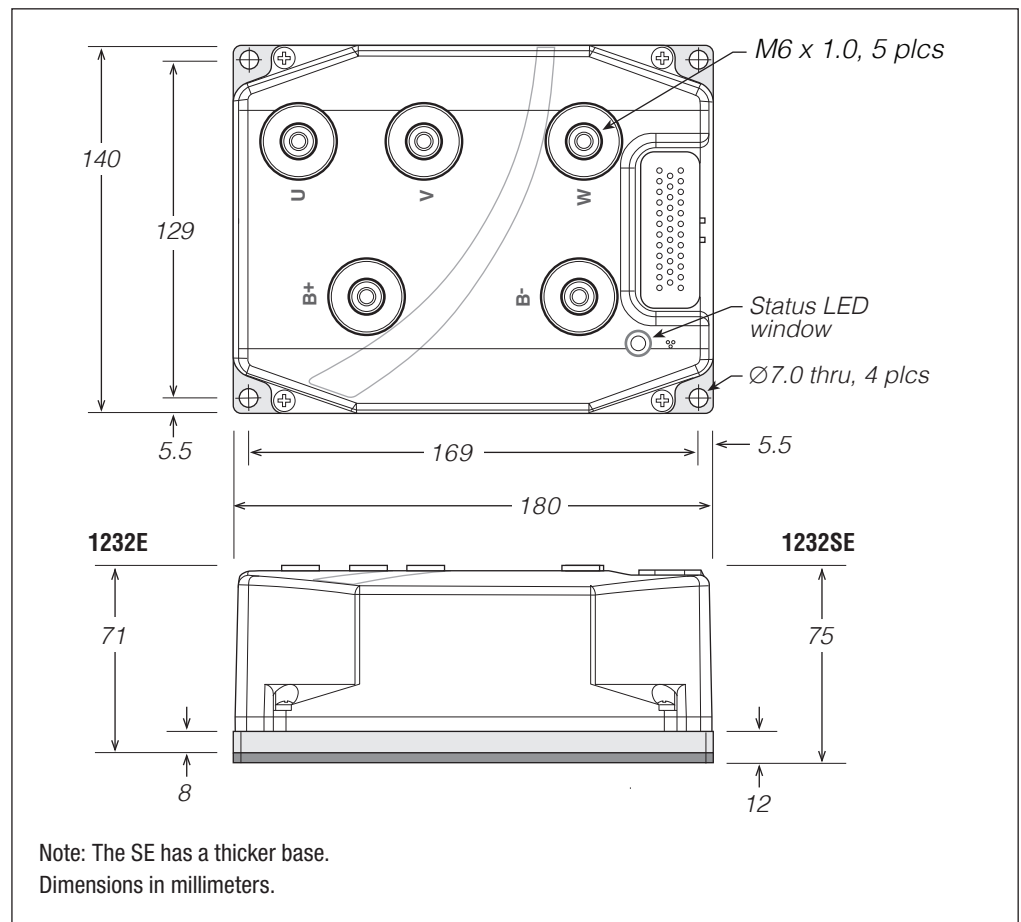


Figure 2b

Mounting dimensions,
Curtis 1234E and 1234SE
motor controllers.

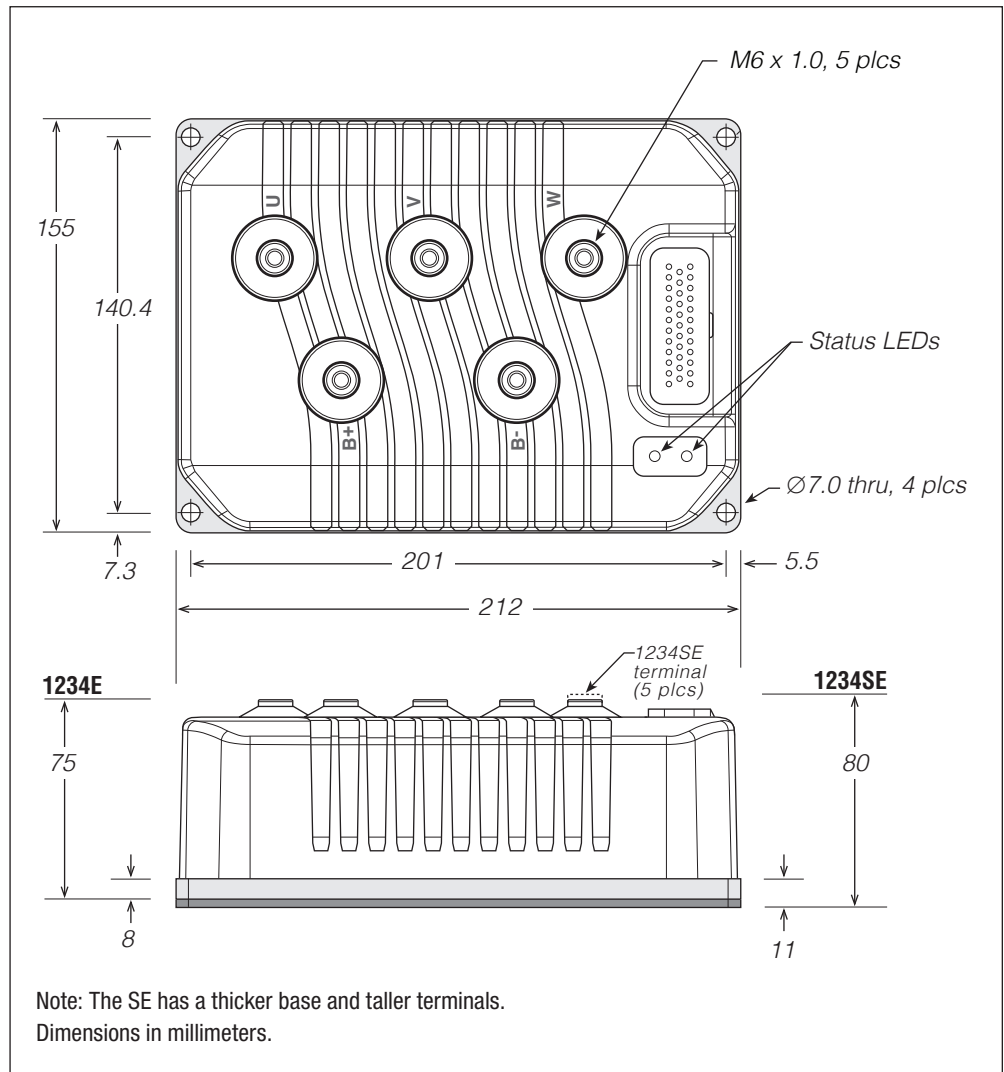
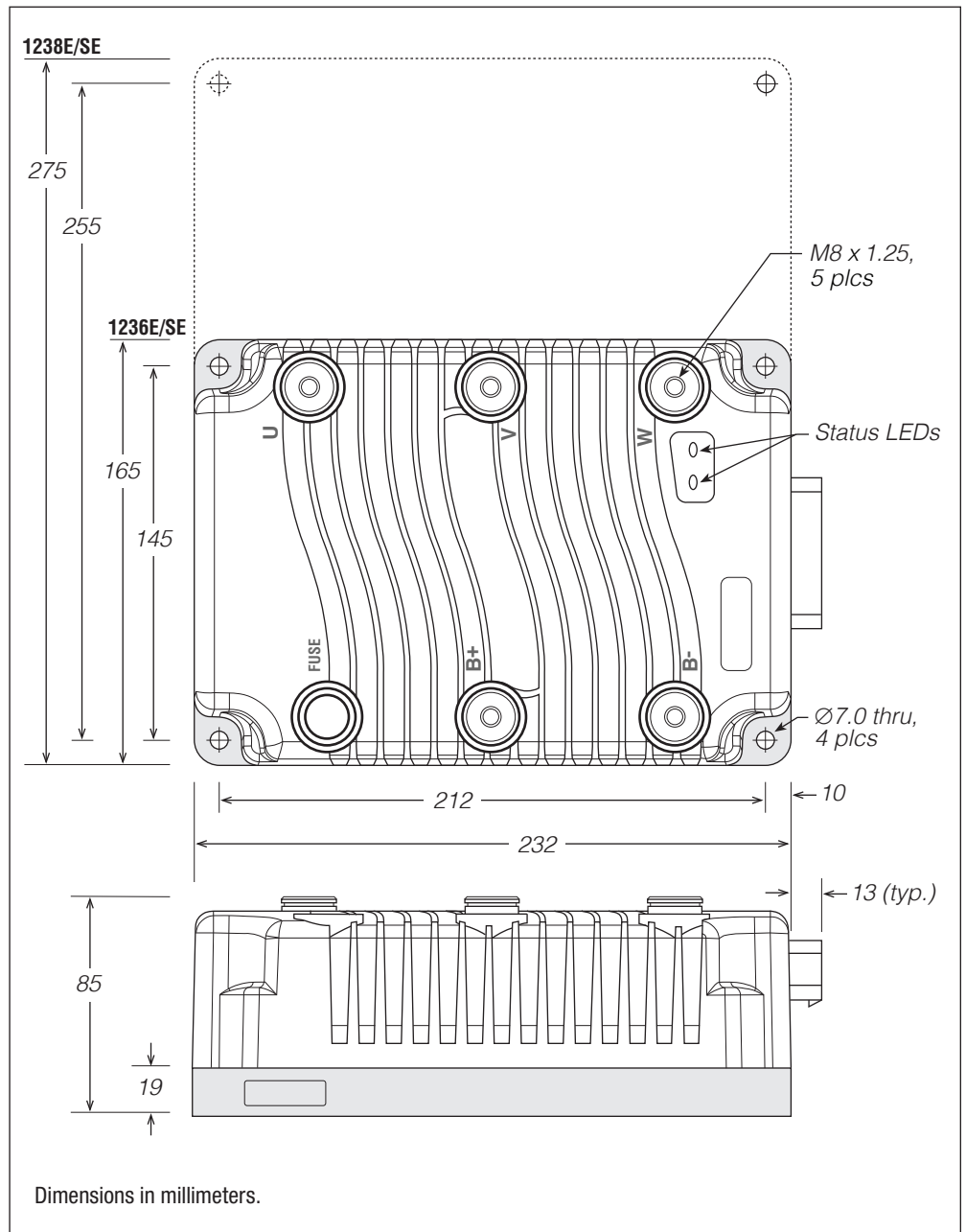


Figure 2c

Mounting dimensions,
Curtis 1236E/SE and
1238E/SE motor controllers.



 WARNING

Working on electrical systems is potentially dangerous. Protect all personnel 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.



Take the required steps during the design and development of the end product to ensure that its EMC performance complies with applicable regulations; suggestions are presented in Appendix A.

These controllers contain **ESD-sensitive components**. Use appropriate precautions in connecting, disconnecting, and handling the controller. See installation suggestions in Appendix A for protecting the controller from ESD damage.

HIGH POWER CONNECTIONS

There are five high power terminals, identified on the controller housing as **B+**, **B-**, **U**, **V**, and **W**.

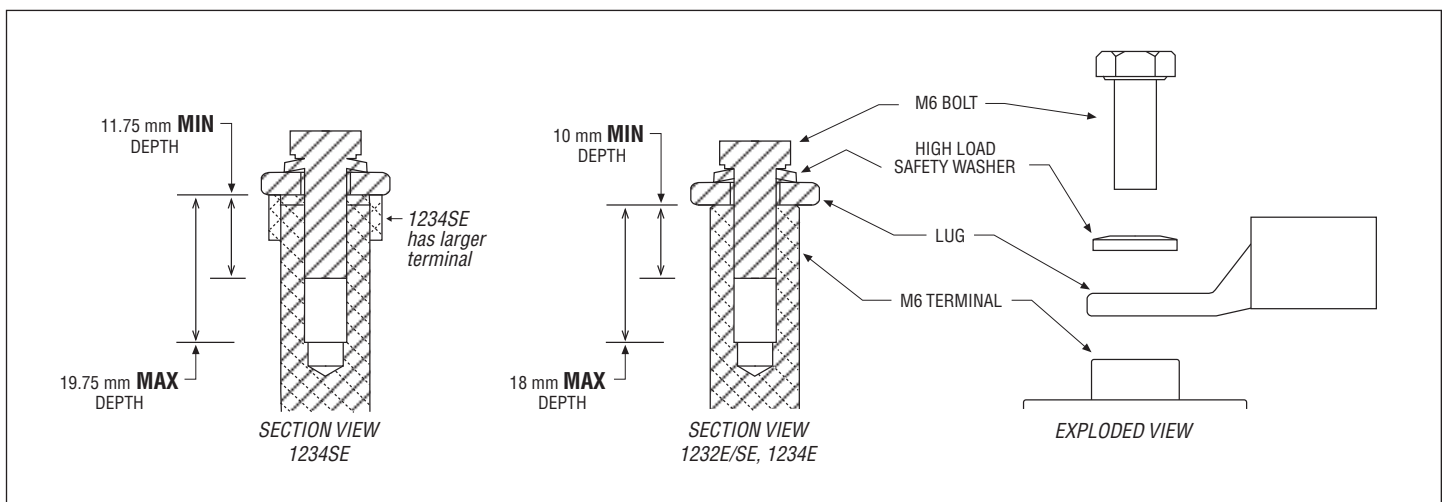
Table 1 High Powered Connections

Terminal	Function
B+	Positive battery to controller
B-	Negative battery to controller
U	Motor phase U
V	Motor phase V
W	Motor phase W

Lug Assembly: 1232E/SE and 1234E/SE models

Five aluminum M6 terminals are provided. Lugs should be installed as follows, using M6 bolts sized to provide proper engagement (see diagram):

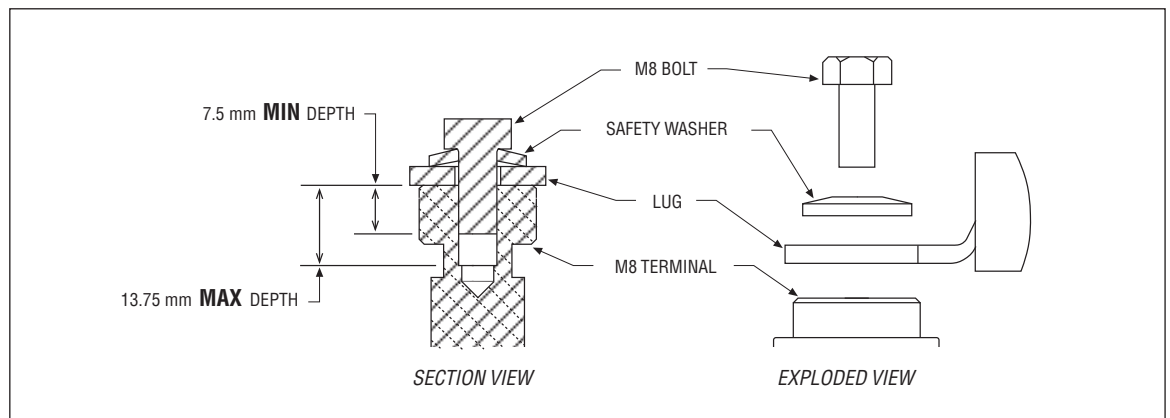
- Place the lug on top of the aluminum terminal, followed by a high-load safety washer with its convex side on top. The washer should be a SCHNORR 416320, or equivalent.
- If two lugs are used on the same terminal, stack them so the lug carrying the least current is on top.
- Tighten the assembly to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



Lug assembly: 1236E/SE and 1238E/SE models

Five M8 terminals are provided. Lugs should be installed as follows, using M8 bolts sized to provide proper engagement (see diagram):

- Place the lug on top of the terminal, followed by a safety washer with its convex side on top. The washer should be a SCHNORR 700800, or equivalent.
- If two lugs are used on the same terminal, stack them so the lug carrying the least current is on top.
- Tighten the assembly to $9.6 \pm 0.9 \text{ N}\cdot\text{m}$ ($85 \pm 8 \text{ in}\cdot\text{lbs}$).



High Power Wiring Guidelines: All Models

Battery cables (B+, B–)

These two cables should be run 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 the cables should not run across the center section of the controller. With multiple high current controllers, use a star ground from the battery B– terminal.

Motor wiring (U, V, W)

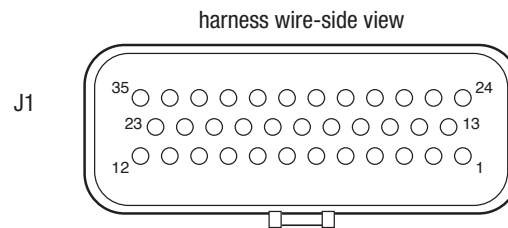
The three phase wires should be close to the same length and bundled together as they run between the controller and the motor. The cable lengths should be kept as short as possible. Use high quality copper lugs and observe the recommended torque ratings.

For optimum noise immunity, the motor cables should not run across the center section of the controller. In applications that seek the lowest possible emissions, a shield can be placed 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. Low current signal wires should not be run parallel to the motor cables. When necessary they should cross the motor cables at a right angle to minimize noise coupling. Refer to Appendix A for further information about Electromagnetic Compatibility (EMC).

LOW POWER 35-PIN CONNECTIONS

All low power connections are made through a single 35-pin AMPSEAL connector. The mating plug housing is AMP p/n 776164-1 and the gold-plated socket terminals are AMP p/n 770520-3 (Strip form) and 770854-3 (loose piece). The connector will accept 0.5 – 1.25 mm (20 – 16 AWG) wire with a 1.7 – 2.7 mm diameter (thin-wall insulation). Seal any non-used connector positions that have the silo-diaphragm pierced with seal plug 770678-1.

The 35 individual pins are characterized in Table 2.



Low Power Wiring Guidelines

Position feedback (Pins 7, 26, 31, 32)

All four wires (+5V, Feedback A, Feedback B, and I/O ground) should be bundled together as they run between the motor and controller logic connector. These can often be run with the rest of the low current wiring harness. The encoder cables should not be run near the motor cables. In applications where this is necessary, shielded cable should be used with the ground shield connected to the I/O ground (pin 7) at only the controller side. In extreme applications, common mode filters (e.g. ferrite beads) could be used.

CANbus (Pins 21, 23, 34, 35)

It is recommended that the CAN wires be run as a twisted pair. However, many successful applications at 125 kbit/s are run without twisting, simply using two lines bundled in with the rest of the low current wiring. The CANbus wiring should be kept away from the high current cables and cross them at right angles when necessary.

All other low power wiring

The remaining low power wiring should be run according to standard practices. When designing the vehicle's wiring and routing, keep the input lines such as throttle, brake, temperature, and the above mentioned encoder signals separate from controller's output lines such as the coil driver outputs. Avoid routing the low-power wiring parallel to the high power (and current) battery and motor cables.

Table 2 Low Power Connections

Pin	Name	Description	Related VCL*	
			Function	References
1	KSI	Keyswitch input. Provides logic power for the controller and power for the coil drivers.		Keyswitch_Voltage
2	Prop. Driver	Proportional driver. This is a coil driver with current control capability typically used for a proportional valve on a hydraulic manifold. Can also be used as a digital input.	Automate_PWM() Put_PWM() Automate_Frequency_Output() Frequency_Output_Duty_Cycle()	Sw_13 PWM5 PD_Current PD_Output PD_Throttle VCL_PD_Throttle
3	Driver 4	Generic driver #4; can also be used as a digital input. Has low frequency PWM capabilities.	Automate_PWM() Put_PWM()	Sw_12 PWM4 PWM4_Output
4	Driver 3	Generic driver #3; can also be used as a digital input. Has low frequency PWM capabilities. Typically used for pump contactor.	Automate_PWM() Put_PWM()	Sw_11 PWM3 PWM3_Output
5	Driver 2	Generic driver #2; can also be used as a digital input. Has low frequency PWM capabilities and a slightly higher current rating. Typically used for electromagnetic brake.	Automate_PWM() Put_PWM()	Sw_10 PWM2 PWM2_Output
6	Driver 1	Generic driver #1; can also be used as a digital input. Has low frequency PWM capabilities. Typically used for main contactor.	Automate_PWM() Put_PWM() Set_Interlock() Clear_Interlock()	Sw_9 PWM1 PWM1_Output Interlock_State Main_State
7	I/O Ground	Input and output ground reference.		
8	Switch 2 Analog 2	Can be used as generic switch input #2 or as generic analog input #2. Typically used as the motor temperature analog input.		Sw_2 Analog2_Input Motor_Temperature
9	Switch 3	Generic switch input #3. Typically used as the interlock switch.		Sw_3
10	Switch 4	Generic switch input #4.		Sw_4
11	Switch 5	Generic switch input #5.		Sw_5
12	Switch 6	Generic switch input #6.		Sw_6
13	Coil Return	This is the coil return pin (at B+ potential) for all the contactor coils.		
14	Switch 16	Generic switch input #16.		Sw_16
15	Throttle Pot High	Pot high connection for a 3-wire throttle pot.		
16	Throttle Pot Wiper	Pot wiper connection for the throttle pot.	Setup_Pot() Setup_Pot_Faults()	Throttle_Pot Throttle_Pot_Output
17	Pot2 Wiper	Pot wiper connection for the brake pot.	Setup_Pot() Setup_Pot_Faults()	Brake_Pot Brake_Pot_Output

* The related VCL columns are vital when writing VCL code (see Chapter 10). VCL “functions” are used to access the various I/Os; VCL “references” are predefined names for specific pins. Refer to the OS SysInfo file for specific VCL functions, controller system variables, usage, and CAN Object IDs.

Table 2 Low Power Connections, cont'd

Pin	Name	Description	Related VCL*	
			Function	References
18	Pot Low	Common pot low connection for the throttle and brake pots.		Pot_Low_Output
19	Digital Out 6	An On/Off output driver. Can also be used as a digital input.	Set_DigOut() Clear_DigOut()	Sw_14 DigOut6 Dig6_Output
20	Digital Out 7	An On/Off output driver. Can also be used as a digital input.	Set_DigOut() Clear_DigOut()	Sw_15 DigOut7 Dig7_Output
21	CAN Term H	High connection for the CAN termination jumper.		
22	Switch 7	Generic switch input #7. Typically used as the Forward switch.		Sw_7
23	CAN H	CANbus high.	Setup_CAN() Setup_Mailbox() Send_Mailbox() etc.	
24	Switch 1 Analog 1	Can be used as generic switch input #1 or as generic analog input #1. Typically used for emergency reverse switch (if applicable).		Sw_1 Analog1_Input
25	+12V Out	Unregulated low power +12V output.		Ext_Supply_Current
26	+5V Out	Regulated low power +5V output.		5_Volts_Output Ext_Supply_Current
27	Pot2 High	Pot high connection for a 3-wire brake pot.		
28	Serial TX	Serial transmit line for display or flash update.	Setup_Serial()	
29	Serial RX	Serial receive line for flash update.	Setup_Serial()	
30	Analog Output	Low power, low frequency 0V to 10V analog output.	Automate_PWM() Put_PWM()	PWM6 Analog_Output
31	Position Feedback A	Quadrature encoder input phase A (ACIM motors).		Motor_RPM MotorspeedA Encoder_Sin_Input_Compensated
32	Position Feedback B	Quadrature encoder input phase B (ACIM motors).		Motor_RPM MotorspeedB Encoder_Cos_Input_Compensated
33	Switch 8	Generic switch input #8. Typically used as the Reverse switch.		Sw_8
34	CAN Term L	Low connection for the CANbus termination jumper.		
35	CAN L	CANbus low.	Setup_CAN() Setup_Mailbox() Send_Mailbox() etc.	

* The related VCL columns are vital when writing VCL code (see Chapter 10). VCL "functions" are used to access the various I/Os; VCL "references" are predefined names for specific pins. Refer to the OS SysInfo file for specific VCL functions, controller system variables, usage, and CAN Object IDs.

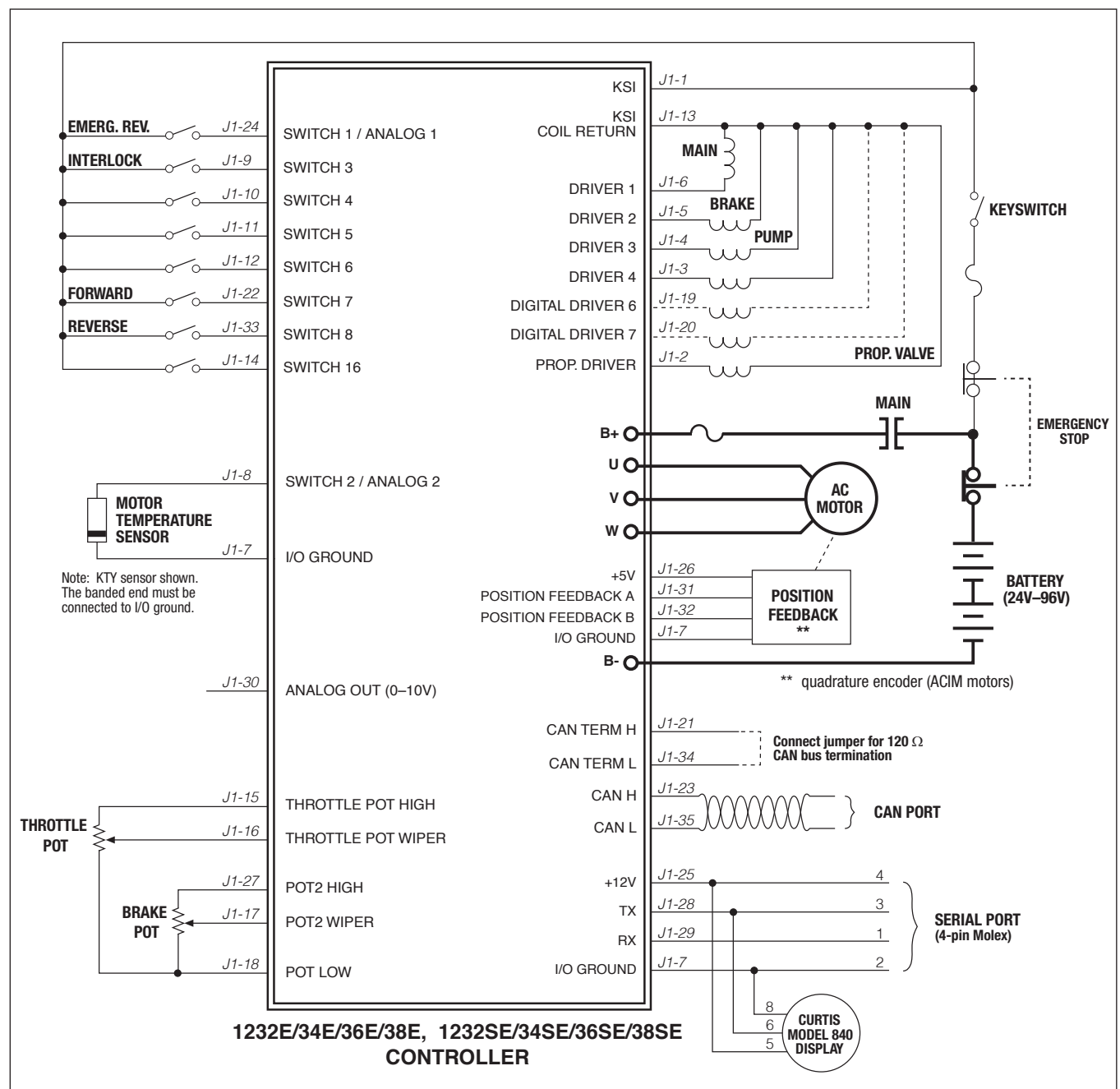
CONTROLLER WIRING: BASIC CONFIGURATION

A basic wiring diagram is shown in Figure 3. Throttle and brake are shown in the diagram as 3-wire potentiometers; other types of throttle and brake inputs are easily accommodated, and are discussed in the following throttle wiring section.

The main contactor coil must be wired directly to the controller as shown in Figure 3 to meet EEC safety requirements. The controller can be programmed to check for welded or missing contactor faults and uses the main contactor coil driver output to remove power from the controller and motor in the event of various other faults. If the main contactor coil is not wired to Pin 6 of the 35-pin

Figure 3

Basic wiring diagram, Curtis 1232E/SE, 34E/SE, 36E/SE, and 38E/SE motor controllers.



connector as shown, 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.

Note that the basic wiring diagram is designed for generic applications and may not fully meet the requirements of every vehicle system. These controllers have very flexible I/O and wiring configurations; Please contact the Curtis distributor or support engineer to discuss a particular application.

SWITCH INPUT WIRING

The following inputs are dedicated to specific functions when the parameter settings are as shown:

Switch 1: Emergency Reverse input if the EMR Enable = On and EMR Type = 0 or 2.

Switch 3: Interlock input if Interlock Type = 0.

Switch 5: Lift input (depends on VCL program).

Switch 6: Lower input (depends on VCL program).

Switch 7: Forward input if Throttle Type = 1–3.

Switch 8: Reverse input if Throttle Type = 1–3.

LOW POWER CIRCUIT SPECIFICATIONS

The input/output circuits wired to the 35-pin connector can be grouped by type as follows; their electrical characteristics are discussed below.

- Digital Inputs.
- Digital and PWM Outputs.
- Analog Inputs.
- Analog Output.
- Power Supply Outputs.
- KSI and Coil Return Inputs.
- Throttle and Brake Inputs.
- Communications Ports I/O.
- Position Feedback Inputs.

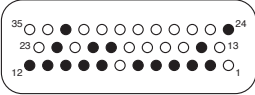
Digital Inputs

These control lines can be used as digital (on/off) inputs. Normal “on” connection is direct to B+; “off” is direct to B–. Input will pull low (off) if no connection is made. All digital inputs are protected against shorts to B+ or B–.

Nine of these lines (Switches 1–8, 16) are designed to pull current to keep switch contacts clean and prevent leakage paths from causing false signals.

The remaining lines are digital inputs associated with driver outputs; note that they have much higher input impedances. The two digital output lines, Digital Out 6 and 7, can also be read as inputs, and are therefore included in this group.

The digital inputs at pins 24 and 8 can also be used as analog inputs, and are included in that group as well.

**DIGITAL INPUT SPECIFICATIONS**

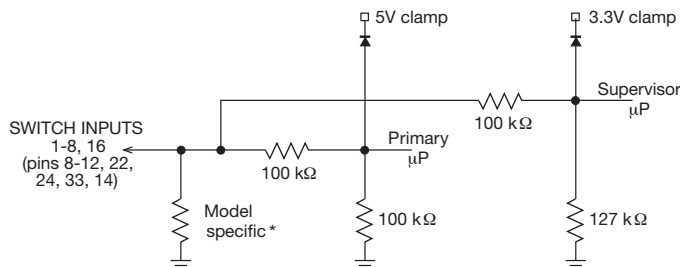
Name	Pin	Logic Thresholds	Input impedance*	Voltage range†	ESD Tolerance
Switch 1	24	Rising edge = 4.4V max Falling edge = 1.5V min	24-36V models: 7.0 kΩ, 7.2 kΩ 36-48V models: 10.8 kΩ, 11.2 kΩ 48-80V models: 25.2 kΩ, 27.3 kΩ 72-96V models: n/a, 29.4 kΩ	-10V to (MaxV + 10V)	± 8 kV (direct strike)
Switch 2	8				
Switch 3	9				
Switch 4	10				
Switch 5	11				
Switch 6	12				
Switch 7	22				
Switch 8	33				
Switch 16	14				
Digital Out 6	19	Rising edge = 4.4V max — = 10V on the 48-80V and 72-96V models Falling edge = 1.5V min	See the Driver Circuit diagrams, below	-5V to (MaxV + 10V)	
Digital Out 7	20				
Driver 1	6				
Driver 2	5				
Driver 3	4				
Driver 4	3				
Prop Driver	2				

Quick Link:
Figure 3 wiring diagram [p.12](#)

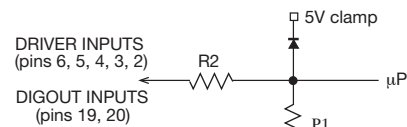
*The first value is for 1232E/SE and 1234E/SE controllers, and the second value is for 1236E/SE and 1238E/SE controllers.

†“MaxV” in this and the following tables is the controller’s maximum voltage; see Table D-1 for the maximum voltage of each model.

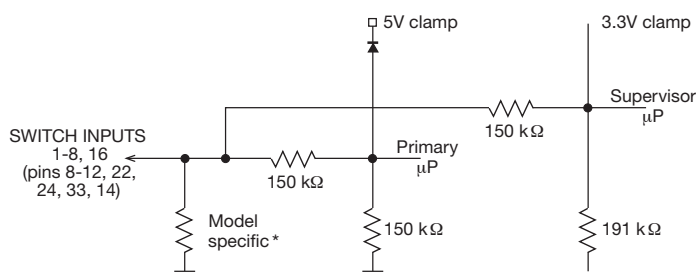
NOTE: The voltage at the switch inputs 3–8 and 16 must be above the high threshold or below the low threshold for proper operation. Allowing these inputs to fall between these thresholds for more than 100 milliseconds could result in a Supervisor Fault (fault code 77).

DIGITAL INPUT IMPEDANCE CIRCUITS**1232E/SE and 1234E/SE controllers**

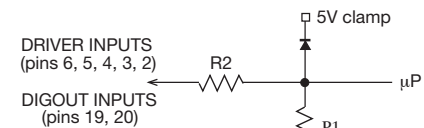
* 24-36V models = 7.5k Ω , 36-48V = 12k Ω , 48-80V = 33k Ω , 72-96V = 36k Ω



Model Voltage Specific	R1 Ω	R2 Ω
24V (also all 1239E models)	100k	100k
24-36V	100k	100k
36-48V	100k	100k
48-80V	49.9k	150k
72-96V	49.9k	150k

1236E/SE and 1238E/SE controllers

* 24-36V models = 7.5k Ω , 36-48V = 12k Ω , 48-80V = 33k Ω , 72-96V = 36k Ω



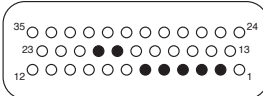
Model Voltage Specific	R1 Ω	R2 Ω
24V (also all 1239E models)	100k	100k
24-36V	100k	100k
36-48V	100k	100k
48-80V	49.9k	150k
72-96V	49.9k	150k

Digital and PWM Outputs

Seven digital (on/off) and PWM output drivers are available. One of these, the proportional driver, can be operated in a current control mode for driving a proportional valve or similar load. The frequency of this driver is normally 18 kHz, but this output can also serve to drive an electronic speedometer or tachometer using the VCL function Automate_Frequency_Output().

Each output can be independently turned on continuously (low level) or pulse width modulated to set the average output voltage. These outputs are intended to drive inductive loads such as contactors and electromagnetic brakes but could also be used to drive resistive loads if peak current ratings are not exceeded. All these outputs are protected against shorts to B+ or B-. All inductive loads should be connected to the coil return (pin 13), which provides flyback diode protection.

These lines can also be used as digital inputs, and are included in that group as well.



DIGITAL and PWM OUTPUT SPECIFICATIONS

Name	Pin	PWM	PV Current	Frequency	Output Current	Protected Voltage	ESD Tolerance		
Driver 1	6	0 to 100% Duty Cycle	N/A	120 to 1000 Hz *	2A Max	-0.5V to (MaxV + 10V)	± 8 kV (direct strike)		
Driver 2	5				3A Max				
Driver 3	4				2A Max				
Driver 4	3								
Prop Driver	2		0 to 2A in 607 nominal steps	18 kHz					
Digital Out 6	19	On / Off	N/A	N/A	1A Max				
Digital Out 7	20								

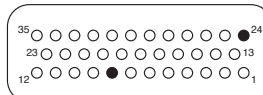
*Drivers 1–4 Frequency is set by the PWM Frequency parameter.

Analog Inputs

Two control lines can be used as analog inputs. Both inputs are protected against shorts to B+ or B-.

Typically Analog 2 is used as the input for the motor temperature sensor. This input provides a constant current appropriate for a thermistor sensor. Some standard predefined motor temperature sensors are supported in software (see the motor's Sensor Type parameter). Note: The industry standard KTY temperature sensors are silicon temperature sensors with a polarity band; the polarity band of a KTY sensor must be the end connected to I/O Ground (pin 7).

These lines can also be used as digital inputs, and are included in that group as well.



ANALOG INPUT SPECIFICATIONS

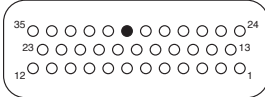
Signal Name	Pin	Operating Voltage	Input impedance*	Protected Voltage	ESD Tolerance
Analog 1	24	0 to 10V in 1024 steps	24-36 V models: 6.9 kΩ, 7.1 kΩ	-10V to (MaxV + 10V)	± 8 kV (direct strike)
Analog 2	8		36-48 V models: 10.5 kΩ, 11.0 kΩ 48-80 V models: 23.8 kΩ, 28.1 kΩ 72-96 V models: n/a, 28.1 kΩ		

* The first value is for 1232E/SE and 1234E/SE controllers, and the second value is for 1236E/SE and 1238E/SE controllers.

Analog Outputs

A single line is available as a low power analog output and is intended to drive instrumentation such as a battery discharge indicator. This output is generated from a filtered PWM signal and has about 1% ripple. The 2% settling time is <25 ms for a 0–5V step and <30 ms for a 0–10V step. This output line is protected against shorts to B+ or B–.

ANALOG OUTPUT SPECIFICATIONS



Signal Name	Pin	Output Voltage	Output Current	Protected Voltage	ESD Tolerance
Analog Out	30	0 to 10V	10 mA	–1V to (MaxV + 10V)	± 8 kV (direct strike)

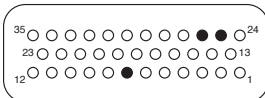
Power Supply Outputs

Quick Link:

[Figure 3 wiring diagram p.12](#)

Two lines provide auxiliary output power for low power circuits such as electronic throttles, LED indicators, displays, position encoder, and remote I/O boards. I/O Ground (at pin 7) is the return line for these low power devices. Both power supply outputs are protected against shorts to B+ or B–.

POWER SUPPLY OUTPUT SPECIFICATIONS



Signal Name	Pin	Output Voltage	Output Current	Protected Voltage	ESD Tolerance
+12V Out	25	11.5 to 14.5V	100 mA max for +12 Out 100 mA max for +5 Out 200 mA max (combined total)	–1V to (MaxV + 10V)	± 8 kV (direct strike)
+5V Out	26	5V ±10%			
I/O Ground	7	n/a		not protected	

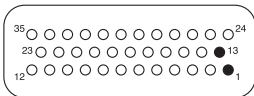
KSI and Coil Return

The **KSI input** provides power for all low power control circuits. This includes the microprocessors, power supply outputs, power for the digital and PWM driver outputs, the power-capacitor precharge (before main contactor closure). Battery voltage is sensed on the input for the VCL battery discharge function.

Coil Return (pin 13) should be wired to the positive battery side of the contactors being driven so that switching noise associated with PWM operation of the contactors is localized to the contactor wiring only.

It is **important** to maintain the division between KSI and coil return in order to ensure reverse polarity protection (e.g., vehicle wiring correct, battery terminals reversed).

KSI and COIL RETURN INPUT SPECIFICATIONS



Signal Name	Pin	Operating Voltage	Input Current	Protected Voltage	ESD Tolerance
KSI	1	Between under- and overvoltage cutbacks	13A max * continuous	± (MaxV + 10V)	± 8 kV (direct strike)
Coil Return	13		10A or 12A max **	(KSI – 0.3V) to (MaxV + 10V)	

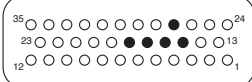
* Including current from the coil return (pin 13).

** 12 A for 1236E/SE and 1238E/SE; 10 A for 1232E/SE and 1234E/SE.

Throttle and Brake Inputs

The two pot inputs are independently programmable to allow use of a voltage throttle or a 2-wire or 3-wire resistance throttle. Voltage throttles require only the Pot Wiper input (with I/O Ground for the return line). Resistance throttles require Pot Wiper and Pot Low (2-wire) or Pot High, Pot Wiper, and Pot Low (3-wire). All throttle I/O is protected against shorts to B+ or B–.

Alternatively, these two inputs can be used for analog signals other than the throttle and brake pot inputs. Configuring the inputs for use with other signals requires VCL programming; see Chapter 10.



Quick Links:

Figure 3 wiring diagram p.12

Throttle & Brake

Types p.20–23

THROTTLE INPUT SPECIFICATIONS

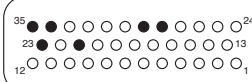
Signal Name	Pin	Operating Voltage	Input Impedance	S/Sink Current	Protected Voltage	ESD Tolerance
Throttle Pot High	15	0V (shorted to Pot Low) 5V (open circuit)	N/A	1 mA nominal (source)	−0.5V to (MaxV + 10V)	± 8 kV (direct strike)
Pot2 High	27					
Throttle Pot Wiper	16	0 to 6.25V	100 kΩ min	0.76 mA nominal (source, 2-wire)		
Pot2 Wiper	17					
Pot Low	18	0 to 0.25V	20 Ω nom.	Faults if above 15 mA (sink)	−1V to (MaxV + 10V)	

Communications Ports

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

The Curtis 1313 handheld programmer and 1314 PC programmer's 1309-serial-interface-device plug into a connector* wired to pins 28 and 29, along with ground (pin 7) and the +12 V power supply (pin 25); see wiring diagram, Figure 3. The Curtis "Spy" Model 840 display will connect the serial-port pins as shown in Figure 3.

Wiring the CAN Term H and CAN Term L pins together provides a local CAN termination of 120 Ω, 0.5 W; keep the length of these wires short. CAN Term H and CAN Term L should never be connected to any external wiring.



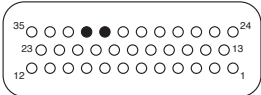
COMMUNICATIONS PORT SPECIFICATIONS

Signal Name	Pin	Supported Protocol / Devices	Data Range	Protected Voltage	ESD Tolerance
CAN H	23	CANopen, other 11-bit or 29-bit identifier protocols	up to 1 Mbit/s	–0.5V to (MaxV + 10V)	± 8 kV (direct strike)
CAN L	35				
CAN Term H	21			(no connection to external wiring)	
CAN Term L	34				
Serial TX	28	Curtis 840 Display, 1313 Handheld Programmer, 1314 PC Programming Station	as required, 9.6 kbit/s to 56 kbit/s	–0.3V to 12V	
Serial RX	29				

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

Position Feedback Input: Quadrature Encoder

Two control lines are internally configured to read a quadrature type position encoder. The encoder is typically powered from the 5V supply (pin 26) or 12V supply (pin 25), but can be powered from any external supply (from 5V up to B+) as long as the logic threshold requirements are met. The quadrature encoder is used in ACIM applications.



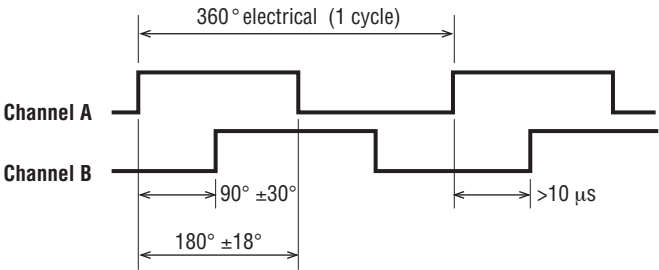
Quick Link:
[Figure 3 wiring diagram p.12](#)

QUADRATURE ENCODER INPUT SPECIFICATIONS

Signal Name	Pin	Logic Threshold	Input Impedance	Max. Frequency	Protected Voltage	ESD Tolerance
Position Feedback A	31	Rising edge= 2.9V max Falling edge= 2.0V min	2 kΩ (internal pull-up to +4.5V)	10 kHz	–5V to (MaxV + 10V)	± 8 kV (direct strike)
Position Feedback B	32					

Phase Shift 90° ±30, Duty Cycle 50% ±10%; no signal edge can be closer than 10 μs to an adjacent edge.

These signal tolerances must be maintained throughout the application’s operating conditions, including voltage, temperature, speed and torque ranges.



Position Feedback Input: Sin/Cos Sensor (encoder)

Use a Sin/Cos sensor in PMAC applications.

A Sine/Cosine encoder is an absolute position sensor that produces two sinusoidal signals, set 90° out of phase. Measurement and comparison of the two signals at any point can determine the absolute position of the sensor. One motor (rotor) mechanical revolution per 360° sin/cos sensor rotation (sweep).

The physical waveforms are not bipolar, but center around an offset voltage (Voff), typically around 2.5 volts. The Peak-Peak voltage (VPP) swing may be as small as 2V or up to 4V, the bigger the range the better the performance for motor control. Both parameters require configuration for the sensor to operate properly.

Configure the Sin Min, Sin Max, Cos Min and Cos Max parameters during commissioning. These normalize the incoming sin/cos signals.

Sine/Cosine Signal Tracking

The values of the inputs are tracked to ensure an accurate position can be obtained if the signal amplitude changes during operation. The amplitude of the signal may alter due to mechanical tolerances, thermal expansion, magnetic field strength changes in the actuator magnet etc.

These signal tolerances must be maintained throughout the application's operating conditions, including voltage, temperature, speed and torque ranges.

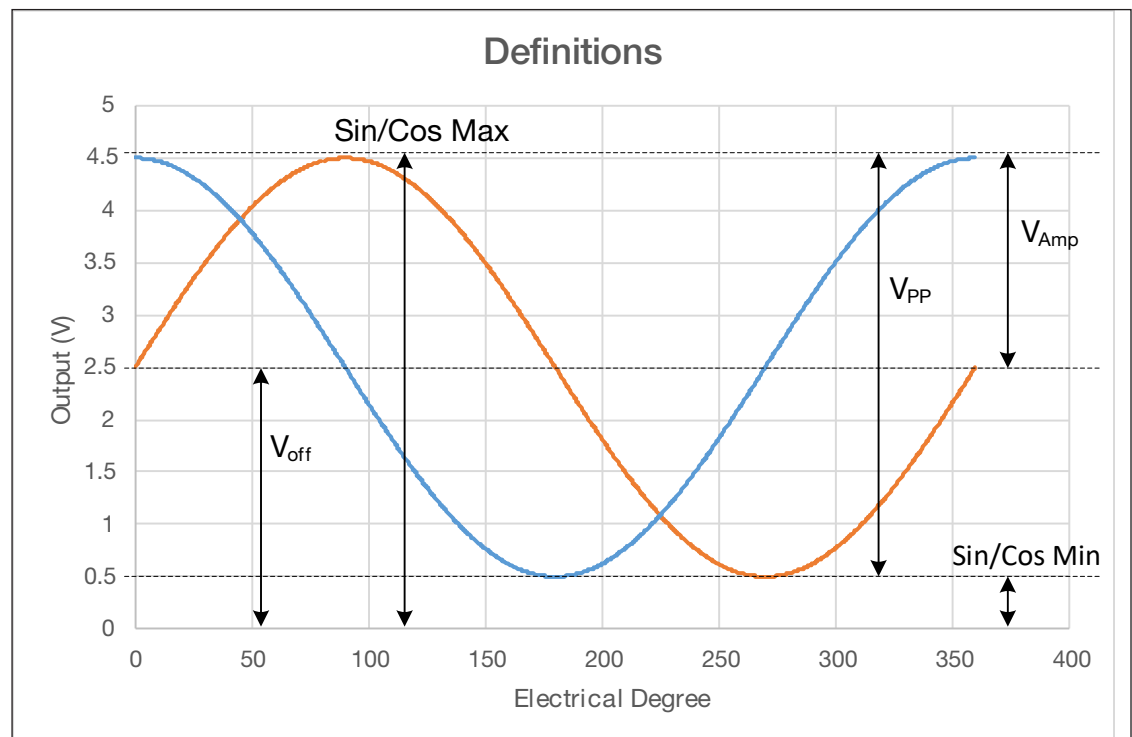
Sin/Cos Sensor Device Electrical Specifications

Signal Name	Pin	Operating Voltage Signal Range	Max Input Frequency ¹	Phase Lag ²	Input Impedance	ESD Tolerance
Position Feedback A	31	0.5-4.5V (+10% / -0%) Optimum: Multiturn sensor. 4Vpp with the signal centered at 2.5V.	500 Hz Sinewave	≤6.1° at 800Hz ±0.5° Max difference between Sine/Cosine inputs	> 50k Ω	± 8 kV (direct strike)
Position Feedback B	32					

¹ Max input frequency may not exceed max controller electrical fundamental frequency.

² Maximum phase lag between sensor and actual rotor position.

This is required to maintain proper PMAC rotor alignment at high speed.



3 — APPLICATION-SPECIFIC FEATURES

Some features of the 1232E/SE – 1238E/SE controllers affect how the specific controller is wired or parameter settings. This chapter provides background information on application-specific features, to assist the vehicle designer in the design process.

THROTTLE WIRING

In this manual, the term *throttle* is used in two senses: (1) as another name for the drive throttle, and (2) as a generic term covering both the drive throttle and the brake throttle. Wiring is the same, whether the throttle in question is used for acceleration or (regen) braking.

Various throttles can be used with these controllers. They are characterized as one of five types in the programming menu of the 1313/1314 programmer.

Type 1: 2-wire 5k Ω –0 potentiometers.

Type 2: single-ended 0–5V throttles, current source throttles, 3-wire potentiometers, and electronic throttles.

Type 3: 2-wire 0–5k Ω potentiometers.

Type 4: wigwag 0–5V throttles and 3-wire potentiometers.

Type 5: VCL input (VCL_Throttle or VCL_Brake).

The two throttle inputs (drive throttle and brake throttle) are programmed independently.

For potentiometers, the controller provides complete throttle fault protection that meets all applicable EEC regulations. **For voltage throttles**, the controller protects against out-of-range wiper values, but does not detect wiring faults; it is therefore the responsibility of the OEM to provide full throttle fault protection in vehicles using voltage throttles.

Throttle types 1–3 use the forward and reverse inputs (switches 7 and 8) in addition to the throttle pot input to define the throttle command (see Figure 15). Throttle types 4 and 5 do not use the forward and reverse inputs.

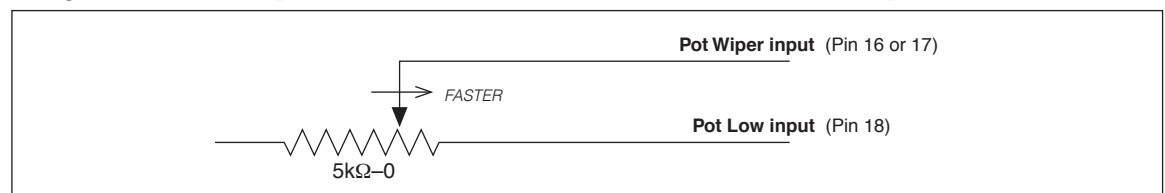
Wiring for the most common throttles is described in the following three pages and shown in the accompanying illustrations. If a throttle the application will use is not covered, contact the Curtis distributor or support engineer.

Quick Link:
Figure 15 [p.114](#)

Throttle Type 1

For these 2-wire resistive potentiometers, shown in Figure 4, full throttle request corresponds to 0 Ω measured between the pot wiper pin and the Pot Low pin. A Type 1 throttle requires the Forward & Reverse Deadbands parameters settings be towards the higher voltage (e.g., 4.50 V) and the Forward & Reverse Max parameters set to the lower voltage (e.g., 0.5 V). Note, this is the opposite of these parameters' default setting. With the 2-wire rheostat in place, the throttle-wiper voltage can be check using the Monitor » Inputs variable Throttle Pot (or Pot2Raw for the brake pot).

Figure 4
Wiring for Type 1
throttles.



Broken wire protection is provided by the controller sensing the current flow from the pot wiper input (pin 16 or 17) through the potentiometer and into Pot Low (pin 18). For Type 1 throttles, if the Pot Low input current falls below 0.65 mA, a throttle fault is generated and the throttle request is zeroed. Note: Pot Low (pin 18) must not be tied to ground (pin 7 or B-).

Throttle Type 2

With these throttles, the controller looks for a voltage signal at the wiper input. Zero throttle request corresponds to 0V and full throttle request to 5V.

A variety of devices can be used with this throttle input type, including voltage sources, current sources, 3-wire pots, and electronic throttles. The wiring for each is slightly different, as shown in Figure 5, and they have varying levels of throttle fault protection.

When a **voltage source** is used as a throttle, it is the responsibility of the OEM to provide appropriate throttle fault detection. For ground-referenced 0–5V throttles, the controller will detect open breaks in the wiper input but cannot provide full throttle fault protection.

To use a **current source** as a throttle, a resistor must be added to the circuit to convert the current source value to a voltage; the resistor should be sized to provide a 0–5V signal variation over the full current range. It is the responsibility of the OEM to provide appropriate throttle fault detection.

When a **3-wire potentiometer** is used, the controller provides full fault protection in accordance with EEC requirements. The pot is used in its voltage divider mode, with the controller providing the voltage source and return. Throttle Pot High (pin 15) provides a current limited 5V source to the 3-wire potentiometer, and Pot Low (pin 18) provides the return path. This is the throttle shown in the basic wiring diagram (Figure 3) for the drive throttle and for the brake throttle.

Complementing the controllers, Curtis offers both Hall-effect and 3-wire potentiometer throttles which are easily integrated into vehicles.

Hall-effect voltage throttles:

The Curtis FP Series of throttles offers multiple pedal angles and mounting configurations (floor, suspended, flush) with 0–5 volt operation with a Idle Validation Switch (IVS).



Curtis FP-SCV-0022 Hall-effect throttle

The ET-XXX electronic throttle is typically used only as a drive throttle (illustrated in Fig. 5).

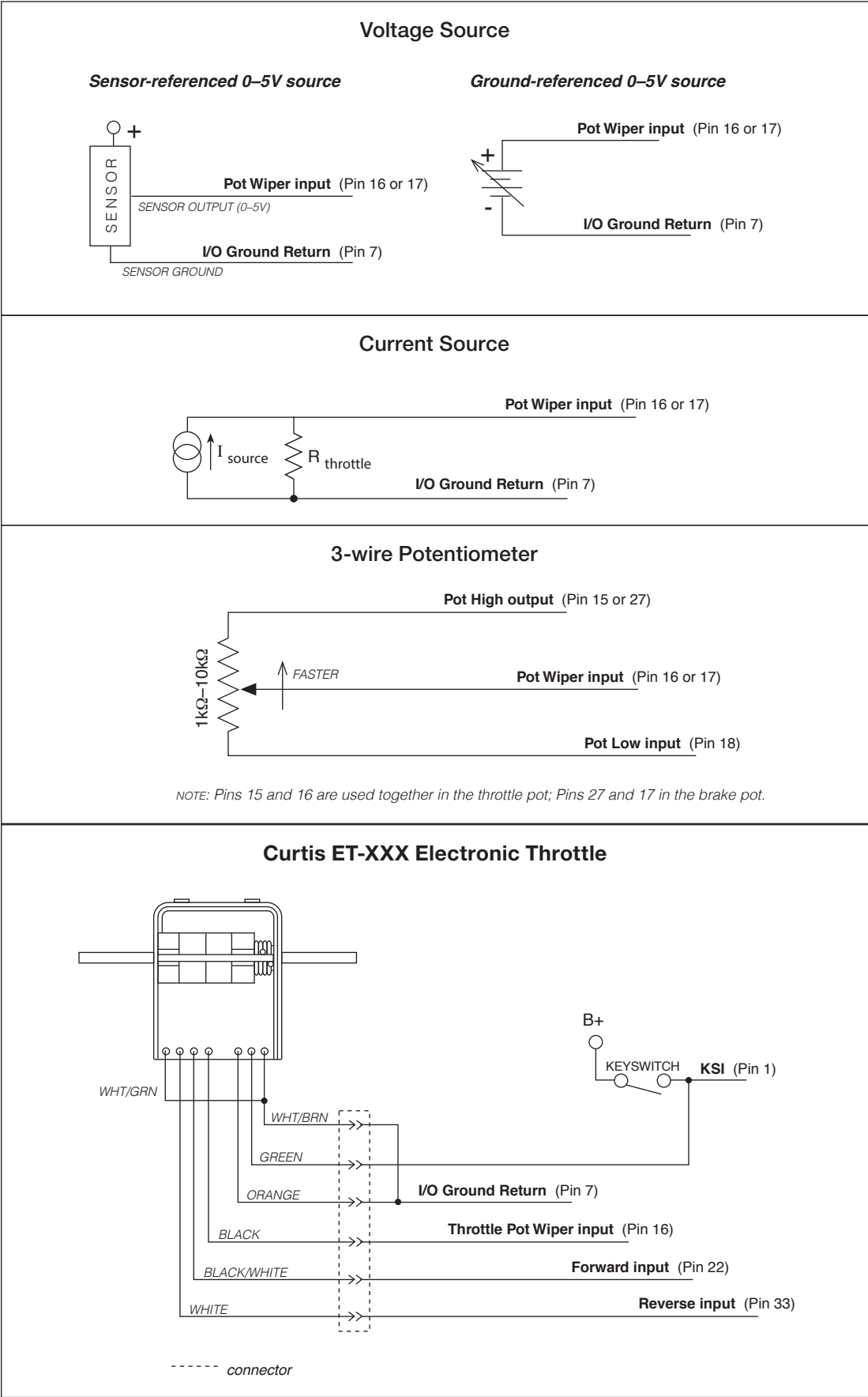
These voltage throttles contains no built-in fault detection, and the controller will detect only open wiper faults. It is the responsibility of the OEM to provide any additional throttle fault detection necessary.

3-wire potentiometer throttle:

The FP-10 model offers, besides the controller's 3-wire fault detection, two throttle spring detection switches and two micro-switches to indicate idle validation and full throttle circuits. This throttle can also be configured from 0-5k (Type 3) or 5k-0 (Type 1) throttles while retaining the detection circuits.

For help with a throttle selection, contact the Curtis distributor or support engineer to discuss the particular throttle requirements and the application of Curtis throttles.

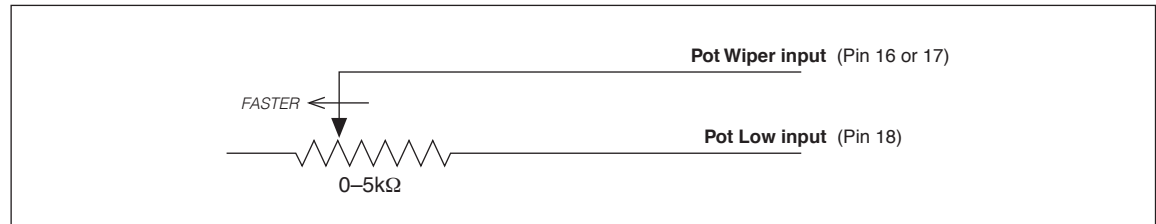
Figure 5
Wiring for Type 2 throttles.



Throttle Type 3

For these 2-wire resistive potentiometers, shown in Figure 6, full throttle request corresponds to 5 k Ω measured between the pot wiper pin and the Pot Low pin.

Figure 6
Wiring for Type 3 throttles.



Broken wire protection is provided by the controller sensing the current flow from the wiper input (pin 16 or 17) through the potentiometer and into Pot Low (pin 18). For Type 3 throttles, if the Pot Low input current falls below 0.65 mA, a throttle fault is generated and the throttle request is zeroed. Note: Pot Low (pin 18) must not be tied to ground (pin 7 or B-).

Throttle Type 4

Type 4 throttles operate in wigwag style. No signals to the controller's forward and reverse inputs are required; the direction is determined by the wiper input value. Only 0–5V voltage sources and 3-wire potentiometers can be used as Type 4 throttles. The controller interface for Type 4 throttles is the same as for the corresponding Type 2 throttles; see Figure 5.

In a Type 4 throttle, the neutral point must be set up somewhere in the center of the throw, with increasing voltage beyond this point providing increasing forward command and voltages below this point providing increasing reverse command. For example, to set the Forward Deadband at 2.6V with Forward Max at 4 V, and Reverse Deadband at 2.4V with Reverse Max at 1V.

When a 3-wire pot is used, the controller provides full fault protection. When a voltage throttle is used, the controller will detect open breaks in the wiper input but cannot provide full throttle fault protection.

Throttle Type 5

Throttle Type 5 provides a different way of sending the throttle command to the controller. This throttle type uses VCL to define the throttle signal that will be “input” into the throttle signal chain as VCL_Throttle (see Figure 15).

This throttle type can be used for either the drive throttle or the brake throttle by using the VCL variables VCL_Throttle or VCL_Brake (see Brake menu). How the VCL program is written will determine the source of the throttle signal, making this a very flexible throttle input method. VCL can be written to use the throttle or brake pot inputs, switch inputs, or CAN communication messages as the source of the throttle signals.

Setting the Throttle Type to Type 5 also allows the throttle pot input (Pin 16) to be redefined by a VCL program for uses other than throttle input.

Note: The option also applies to the Brake Type, which when set to Type 5 uses VCL_Brake as signal chain for the Brake_Command (see the Brake parameter menu and Figure 15).

For questions regarding this throttle type, contact the Curtis distributor or support engineer.

Quick Links:

Figure 15 [p.114](#)

Throttle Type parameter [p.47](#)

Brake Type parameter [p.49](#)

MOTOR SPEED CONSTRAINTS

The maximum motor speed is a programmable parameter in each control mode. Regardless of which control mode is used, the maximum motor speed the controller will allow is constrained by the number of motor poles, the encoder pulses per motor revolution, and the maximum speed constraint imposed by the firmware.

NOTE: The overall maximum motor speed* allowed is the least of the following three constraints:

1. Electrical Frequency Constraint

The controller is designed to output fundamental electrical frequencies up to 450 Hz. It accomplishes this by clamping the Max Speed allowed, using the equation:

$$\text{Max Speed Frequency Limit} = 54000 / \text{Number of Motor Poles}$$

Thus, for example, an 8-pole motor running synchronously at 450 Hz would rotate at $54000/8 = 6750$ rpm (max). Therefore the internal control software will limit the max speed to 6750 rpm for an 8-pole motor. Limited over-speed is allowed, for example if the motor were to go over this speed going down a hill, the controller will still attempt to produce the correct frequency for maximized torque and proper control; it will not simply clamp to 450 Hz.

2. Encoder Pulses/Revolution Constraint (quadrature encoder)

The maximum encoder frequency the controller will accept is 10 kHz. To determine how fast this constraint will allow the motor to spin, use the equation

$$\text{Max Speed Encoder Limit} = 600000 / \text{Encoder Size}$$

(e.g., a motor with a 128-pulse encoder can run up to 4687 rpm).

3. Firmware Max Speed Constraint

The maximum motor speed the controller will allow is 10,000 rpm.

$$\text{Max Speed RPM Limit} = 10,000^*$$

Quick Link:
Max Speed Controller
Limit [p.80](#)

** This maximum allowed speed is displayed in the Monitor » Motor » Max Speed Controller Limit variable.*

Note: In the case where the Max Speed parameter is the prevailing constraint, greater RPM may be possible.

Contact the Curtis distributor or support engineer to discuss the application.

VOLTAGE LIMITS

The controller establishes both hardware-based voltage limits and parameter-based user defined limits. Overvoltage protection cuts back regen braking to prevent damage to batteries and other electrical system components due to overvoltage. Undervoltage protection prevents systems from operating at voltages below their design thresholds.

The four threshold points are calculated from the Nominal Voltage, Undervoltage Kp and Ki, User Overvoltage, and User Undervoltage parameter settings and the controller's minimum voltage and maximum voltage ratings. Note that both the KSI (pin 1) and the B+ terminal (when the main is closed) are at battery voltage, and the capacitor bank is precharged via KSI prior to the main closure.

Overvoltage = Either Max Voltage (see Table D-1)
or $\text{User Overvoltage} \times \text{Nominal Voltage}$, whichever is lower.

Severe Overvoltage = Overvoltage (see previous item) + 10V.

Undervoltage = Either Min Voltage (see Table D-1)
or $\text{User Undervoltage} \times \text{Nominal Voltage}$, whichever is higher.

Severe Undervoltage = Either drive current cut back to 0% for 64 ms or Brownout Voltage * (see Table D-1) is reached, whichever comes first.

* The Brownout Voltage is determined by the controller base type and cannot be changed. When the controller's capacitor voltage falls below the Brownout voltage the bridge is switched off (i.e., motor current is switched off). If the capacitor voltage stays below the Brownout voltage for > 64 milliseconds the controller will reset (equivalent to cycling the keyswitch). If the capacitor voltage rises above the Brownout voltage before 64 ms have passed the bridge will be re-enabled. The Severe Undervoltage point cannot be set lower than the Brownout voltage.

BATTERY DISCHARGE INDICATOR

The lead-acid battery discharge indicator (BDI) algorithm continuously calculates the battery state-of-charge from the KSI voltage. The result of the BDI algorithm is the variable BDI Percentage, which is viewable in the 1313/1314 menu Monitor » Battery. When KSI is turned off, the present BDI Percentage is stored in nonvolatile memory.

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

	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 the standard values for the application's type of batteries as the starting point in setting the reset, full, and empty volts-per-cell parameters. Note: For non lead-acid batteries, including Lithium-Ion battery packs, use the pack's or cell manufacturer's approved Battery Management System (BMS) for determining BDI.

Quick Links:
Controller voltage
ratings [p.160](#)
User/parameter voltage
limits [p.68](#)

4 — PROGRAMMABLE PARAMETERS

These controllers have a number of parameters that can be programmed using a Curtis 1313 handheld programmer or 1314 PC Programming Station. The programmable parameters allow the vehicle's performance to be customized to fit the needs of specific applications.

PROGRAMMING MENUS

The programmable parameters are grouped into nested hierarchical menus, as shown in Table 3.

Motor response tuning

Motor response characteristics can be tuned through speed control or through torque control, depending on the application. Use the Control Mode Select parameter ([page 32](#)) to select which tuning mode to use:

- Speed Mode Express.
- Speed Mode.
- Torque Mode.

Speed Mode Express is a simplified version of Speed Mode with a reduced set of parameters that is adequate for most speed-controlled applications.

Use *Speed Mode* or *Speed Mode Express* for applications where throttle input corresponds to motor speed output.

Use *Torque Mode* for applications where throttle input corresponds to motor torque output.

Note: Tune using torque control or speed control, but not both. For example, if adjusting a torque control parameter while Speed Mode or Speed Mode Express has been selected as the tuning mode, the programmer will show the new setting but it will have no effect.

We strongly urge reading Chapter 7, Initial Setup, before adjusting any of the parameters.

NOTICE

Even if opting to leave most of the parameters at their default settings, it is imperative that the procedures outlined in Chapter 7 are performed (completed), which sets up the basic system characteristics for an application.

Parameter change faults

Parameters marked **[PCF]** in the menu charts will set a Parameter Change Fault (code 49) if they are changed while the motor bridge is enabled (interlock = On). Although the parameter will be changed, the fault will prevent motor control functions until the fault is 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.

Table 3 Programmable Parameters Menus: 1313/1314 Programmer

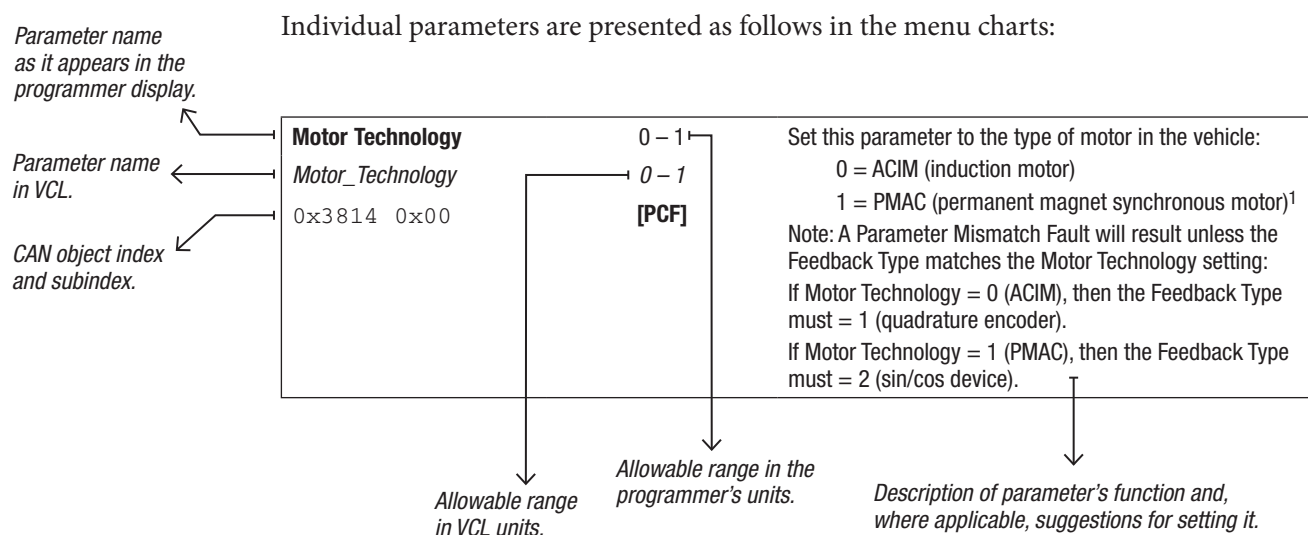
CONTROL MODE SELECT..... p. 32	— RESTRAINT..... p. 38	CURRENT LIMITS MENU..... p. 44
0 - SPEED MODE EXPRESS..... p. 33	— Restraint Forward	— Drive Current Limit
— Max Speed	— Restraint Back	— Regen Current Limit
— Kp	— Soft Stop Speed	— Brake Current Limit
— Ki	— POSITION HOLD..... p. 39	— EMR Current Limit
— Accel Rate	— Position Hold Enable	— Interlock Brake Current Limit
— Decel Rate	— Position Hold Timeout Time	— POWER LIMITING MAP..... p. 44
— Brake Rate	— Kp	— PL Nominal Speed
— Pump Enable	— Kd	— Delta Speed
— Regen Lower Enable	— Zero Speed Threshold	— DRIVE LIMITING MAP..... p. 45
	— Zero Speed Threshold Time	— Nominal
	— Position Hold Settling Time	— Plus Delta
	— Entry Rate	— Plus 2x Delta
	— Exit Rollback Reduction	— Plus 4x Delta
	— Pump Enable..... p. 40	— Plus 8x Delta
	— Regen Lower Enable..... p. 40	— REGEN LIMITING MAP... p. 46
		— Nominal
		— Plus Delta
		— Plus 2x Delta
		— Plus 4x Delta
		— Plus 8x Delta
1 - SPEED MODE MENU..... p. 34	2 - TORQUE MODE MENU..... p. 40	
— SPEED CONTROLLER..... p. 34	— SPEED LIMITER..... p. 40	THROTTLE MENU..... p. 47
— Max Speed	— Max Speed	— Throttle Type
— Kp	— Kp	— Forward Deadband
— Ki LS	— Ki	— Forward Map
— Ki HS	— Kd	— Forward Max
— VEL FEEDFORWARD..... p. 34	— RESPONSE..... p. 41	— Forward Offset
— Kvff	— Accel Rate	— Reverse Deadband
— Build Rate	— Accel Release Rate	— Reverse Map
— Release Rate	— Brake Rate	— Reverse Max
— ACC FEEDFORWARD p. 35	— Brake Release Rate	— Reverse Offset
— Kaff	— Neutral Braking	— Throttle Filter
— Kbff	— Neutral Taper Speed	— HPD SRO Type
— Build Rate	— Forward Full Restraint Speed	— Sequencing Delay
— Release Rate	— Back Full Restraint Speed	— VCL Throttle Enable
— RESPONSE..... p. 36	— FINE TUNING..... p. 42	
— Full Accel Rate HS	— Creep Torque	BRAKE MENU..... p. 49
— Full Accel Rate LS	— Brake Full Creep Cancel	— Brake Pedal Enable
— Low Accel Rate	— Creep Build Rate	— Brake Type
— Neutral Decel Rate HS	— Creep Release Rate	— Brake Deadband
— Neutral Decel Rate LS	— Gear Soften	— Brake Map
— Full Brake Rate HS	— Brake Taper Speed	— Brake Max
— Full Brake Rate LS	— Reversal Soften	— Brake Offset
— Low Brake Rate	— Max Speed Decel	— Brake Filter
— FINE TUNING..... p. 37		— VCL Brake Enable
— Partial Decel Rate		
— HS (High Speed)		
— LS (Low Speed)		
— Reversal Soften		
— Max Speed Accel		
— Max Speed Decel		
— Creep Speed		

Table 3 Programmable Parameters Menus: 1313/1314 Programmer, cont'd

EM BRAKE CONTROL MENU..... p. 50 — Brake Type — Pull In Voltage — Holding Voltage — Battery Voltage Comp. — Set EM Brake On Fault — Zero Speed Threshold — Zero Speed Threshold Time — Position Hold Settling Time — Brake Set Time — Torque Release Time — Brake Release Time — Torque Preload Time — Torque Preload Enable — Save Torque Preload — Torque Preload Cancel Delay — EM Brake Fault Motor Revs	— PD Max Current — PD Min Current — PD Dither % — PD Dither Period — PD Kp — PD Ki — DRIVER 3..... p. 54 — Contactor Enable — Pull In Voltage — Holding Voltage — FAULT CHECKING..... p. 55 — Driver1 Checks Enable — Driver2 Checks Enable — Driver3 Checks Enable — Driver4 Checks Enable — PD Checks Enable — External Supply Max — External Supply Min — PWM FREQUENCY..... p.55	— LOS (LIMITED OPERATING STRATEGY)..... p. 60 — LOS Upon Encoder Fault — LOS Max Speed — LOS Max Current — LOS Max Mod Depth — LOS Accel Rate — LOS Decel Rate — 1 - PMAC (PERMANENT MAGNET MOTOR)..... p. 61 — COMMISSIONING TESTS..... p. 61 — Typical Max Speed — Max Test Speed — Motor Type — Test Enable — Test Throttle — Swap Direction — Max Test Current — FIELD WEAKENING CONTROL.... p. 62 — Field Weakening Rate — FW Current Limit (Apk) — Motor Type — Restricted Operation — COMMISSIONING RESULTS..... p. 63 — Tuning Voltage — Short Circuit Current (RMS) — Phasing Mode — Sensor Position Offset — Sensor Compensation Factor — Kp d Axis Current — Ki d Axis Current — Kp q Axis Current — Ki q Axis Current — Sin Min — Sin Max — Cos Min — Cos Max — Rsys — Commissioning Needed — FEEDBACK OPTIONS..... p. 64 — Feedback Type
— EM BRAKE TEST MENU..... p. 52 — Mode — Direction — Current — Ramp Time — Test Threshold — Hold Time — Max Speed Limit — Hill Compensation Enable	MOTOR MENU..... p. 56 — Typical Max Speed — Motor Technology — 0 - INDUCTION MOTOR (ACIM) p. 57 — CHARACTERIZATION TESTS p. 57 — Test Enable — Test Throttle — Motor Poles — Max Test Speed — Max Text Current — SlipGain — Current Reg Tuning Test Enable — FIELD WEAKENING CONTROL... p. 58 — FW Base Speed — Field Weakening Drive — Weakening Rate Drive — Min Field Current — Swap Two Phases — Swap Feedback Direction — Motor Type	
DRIVERS MENU..... p. 52 — MAIN CONTACTOR..... p. 52 — Main Enable — Main Interlock Type — Pull In Voltage — Holding Voltage — Battery Voltage Comp. — Interlock Type — Open Delay — Weld Check Enable — Main DNC Check Enable — Weld Check On Open Enable — Main DNC Check Threshold — Precharge Enable — PROPORTIONAL DRIVER..... p. 54 — PD Enable — Hyd Lower Enable		

Table 3 Programmable Parameters Menus: 1313/1314 Programmer, cont'd

— 1 - QUADRATURE ENCODER..... p. 64 — Encoder Steps — SPEED FILTERS..... p. 64 — Speed Filter — Speed Filter Zero Speed — 2 - SIN/COS..... p. 65 — Enable Multiturn Sensor — Sin Cos Fault Threshold — Sin Cos fault Threshold High — Sin Cos Fault Time — Sin Cos Startup Time — Sin Min — Sin Max — Cos Min — Cos Max — SPEED FILTERS..... p. 65 — Speed Filter — Speed Filter Zero Speed — FEEDBACK FAULT SETUP..... p. 66 — Fault Detection Enable — Fault Max Speed — Fault Min Speed — Pulse Fault Detect Time — Fault Stall Speed — Fault Stall Time — TEMPERATURE CONTROL..... p. 67 — Sensor Enable — Sensor Type — Sensor Offset — Braking Thermal Cutback Enable — Temperature Hot — Temperature Max — MotorTemp LOS Max Speed	BATTERY MENU..... p. 68 — Nominal Voltage — Kp UV — Ki UV — User Overvoltage — User Undervoltage — Reset Volts Per Cell — Full Volts Per Cell — Empty Volts Per Cell — Discharge Time — BDI Reset Percent — BATTERY CURRENT LIMITING MENU..... p. 69 — Battery Current Limiter Enable — Drive Battery Current Limit — Regen Battery Current Limit — Kp — Kp HS — Ki — Ki HS — Kd — Kd HS — Limit Ramp Rate BATTERY POWER LIMITING MENU..... p. 70 — Drive Power Limit — Regen Power Limit — Voltage Filter DUAL DRIVE MENU see Dual Drive addendum, Document number: 53097 DUAL-DRIVE SUPPLEMENT_os37 VEHICLE MENU..... p. 71 — Metric Units — Speed to RPM — Capture Speed 1 — Capture Speed 2 — Capture Distance 1 — Capture Distance 2 — Capture Distance 3	EMERGENCY REVERSE MENU p. 72 — EMR Enable — EMR Type — EMR Dir Interlock — EMR Time Limit — EMR Speed — EMR Accel Rate — EMR Decel Rate INTERLOCK BRAKING MENU..... p. 73 — Enable — Decel Rate HS — Decel Rate LS — Interlock Brake Timeout — Settling Time CAN INTERFACE MENU..... p. 74 — CANOpen Interlock — CAN Node ID 1 — CAN Node ID 2 — CAN Node ID 3 — CAN Node ID 4 — Supervisor Node ID — Baud Rate — Heartbeat Rate — PDO1 Timeout Period — PDO2 Timeout Period — Emergency Message Rate — Suppress CANOpen Init RESET CONTROLLER..... p. 75
--	---	--



Note: All bit variables have two VCL parameter names. The first is the name of the bit, and the second is the name of the byte containing the bit. The bit position within the byte is indicated in brackets after the byte name.

Examples:

BIT NAME: **Metric_Units**

BYTE NAME: *OptionBits3* [Bit 5]

BIT NAME: **EMR_Dir_Interlock**

BYTE NAME: *EMR_DIR_INTERLOCK_BIT0* [Bit 0]

In the second example, “_Bit0” is part of the byte name, and does not indicate the bit position; this byte, like all bytes, has 8 available bits.

Within the menu charts, each pair of bit variable names is shown as a grouped set, with the bit name appearing first and then the byte name:

Metric Units	On/Off
<i>Metric_Units</i>	<i>On/Off</i>
<i>OptionBits3</i>	[Bit 5]

SDO Write Message

To retain parameter values changed via CANopen SDO write messages following a key-cycle, write a non-zero value to `CAN_EE_Writes_Enabled` (object index 0x332F, subindex 0x00) before changing parameter values. This will cause changes to be written to non-volatile memory immediately. After completing the changes, write the value zero to `CAN_EE_Writes_Enabled`.

CAUTION

Do not leave `CAN_EE_Writes_Enabled` at a non-zero value during normal operation, because damage to the controller's EEPROM may occur.

CAN SDO and PDO message transmission nomenclature:

With regard to SDO messages, the terms “Client”, “Server”, “Request”, and “Response” are used. For SDOs, the Client /Server are:

- Used in SDO transfers, including data uploads and downloads.
- The Client device (controller) initiates the communications, whether it is a read, write, upload or download.
- The Sever must respond to the Client request with data and confirm or issue an abort.
- Uploads are data from the device (controller).
- Downloads are data sent to the device (controller).
- SDO_RX (Receive), or SDO_REQ.
- SDO_TX (Transmit), or SDO_RESP.

PDOs messages are always from the perspective of the ancillary device (controller).

- Transmit PDOs (TPDO) are messages that are sent by the ancillary device (controller).
- Receive PDOs (RPDO) are messages that are received by the ancillary device (controller).

CONTROL MODE SELECT

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Control Mode Select	0 – 2	This parameter determines which control method will be in effect when programming motor response: <u>0 = SPEED MODE EXPRESS.</u> <u>1 = SPEED MODE.</u> <u>2 = TORQUE MODE.</u> Contact the Curtis distributor or support engineer if there is interest in a non-generic control method. NOTE: Such a case will almost always require time, which may delay any immediate project's time-frame over the standard options. NOTE: Do not change this parameter while the controller is powering the motor. Any time this parameter is changed a Parameter Change Fault (fault code 49) is set and must be cleared by cycling power; this protects the controller and the operator.
<i>Control_Mode_Select</i>	0 – 2	
0x3010 0x00	[PCF]	

0 - SPEED MODE EXPRESS — SPEED MODE EXPRESS MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Max Speed <i>Max_Speed_SpdMx</i> 0x3840 0x00	100 – 10000 rpm <i>100 – 10000</i>	Defines the maximum requested motor rpm at full throttle. Partially-applied throttle is scaled proportionately; e.g., 40% applied throttle corresponds to a request for 40% of the set Max Speed Value. If Max_Speed_SpdMx is set <10 rpm (through VCL or CAN), the throttle request is zeroed. NOTE: The maximum motor rpm is subject to the constraints on page 24.
Kp <i>Kp_SpdMx</i> 0x3841 0x00	0 – 100% <i>0 – 8192</i>	Determines how aggressively the speed controller attempts to match the speed of the motor to the commanded speed. Larger values provide tighter control. If the gain is set too high, the application may experience oscillations as the controller tries to control speed. If it is set too low, the motor may behave sluggishly and be difficult to control.
Ki <i>Ki_SpdMx</i> 0x3842 0x00	5 – 100% <i>50 – 1000</i>	The integral term (Ki) forces zero steady state error, so the motor will run at exactly the commanded speed. Larger values provide tighter control. If the gain is set too high, the application may experience oscillations as the controller tries to control speed. If it is set too low, the motor may take a long time to approach the exact commanded speed.
Accel Rate <i>Accel_Rate_SpdMx</i> 0x3843 0x00	0.1 – 30.0 s <i>100 – 30000</i>	Sets the rate (in seconds) at which the speed command increases when throttle is applied. Larger values represent slower response.
Decel Rate <i>Decel_Rate_SpdMx</i> 0x3847 0x00	0.1 – 30.0 s <i>100 – 30000</i>	Sets the rate (in seconds) that is used to slow down the vehicle when the throttle is reduced. Larger values represent slower response.
Brake Rate <i>Brake_Rate_SpdMx</i> 0x3848 0x00	0.1 – 30.0 s <i>100 – 30000</i>	Sets the rate (in seconds) at which the vehicle slows down when brake is applied or when throttle is applied in the opposite direction. Larger values represent slower response.
Pump Enable* <i>AC_Pump_Enable_SpdM</i> <i>AC_Pump_Enable_SpdM_Bit0 [Bit 0]</i> 0x3896 0x00	On / Off <i>On / Off</i>	This parameter should be programmed On to operate a pump motor rather than a vehicle drive motor. The speed-controller responsiveness and stability are enhanced for pump-motor applications.
Regen Lower Enable* <i>Regen_Lower_Enable_SpdM</i> <i>Regen_Lower_Enable_SpdM_Bit0 [Bit 0]</i> 0x3877 0x00	On / Off <i>On / Off</i>	This parameter works together with Pump Enable as follows: 1) When Pump Enable = On and Regen Lower Enable = On, the pump motor can turn in both the forward and the reverse direction. In this case, the pump can be used as the Lower function (reverse). The pump motor, when “driven in reverse by a load” will also regen current in a similar fashion to a drive-motor regenerative braking. 2) When Pump Enable = On and Regen Lower Enable = Off, the pump motor can turn only in the forward direction. In this case, a hydraulic valve is typically used for the Lower function. 3) When Pump Enable = Off, the Regen Lower Enable parameter has no effect on the control system.

* This parameter appears twice in the parameter menu structure. Changing the value of this parameter affects the same parameter in Speed Mode.

Quick Link:

Motor Speed

Constraints [p.24](#)

1- SPEED MODE — SPEED CONTROLLER MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Max Speed <i>Max_Speed_SpdM</i> 0x3011 0x00	100 – 10000 rpm <i>100 – 10000</i>	Defines the maximum requested motor rpm at full throttle. Partially-applied throttle is scaled proportionately; e.g., 40% applied throttle corresponds to a request for 40% of the set Max Speed Value. If Max_Speed_SpdM is set <10 rpm (through VCL or CAN), the throttle request is zeroed. NOTE: The maximum motor rpm is subject to the constraints on page 24.
Kp <i>Kp_SpdM</i> 0x3012 0x00	0 – 100% <i>0 – 8192</i>	Determines how aggressively the speed controller attempts to match the speed of the motor to the commanded speed. Larger values provide tighter control. If the gain is set too high, the application may experience oscillations as the controller tries to control speed. If it is set too low, the motor may behave sluggishly and be difficult to control.
Ki LS <i>Ki_SpdM</i> 0x3015 0x00	5 – 100% <i>50 – 1000</i>	The Ki LS parameter sets the Ki for low vehicle speeds. The integral term (Ki) forces zero steady state error, so the motor will run at exactly the commanded speed. Larger values provide tighter control. If the gain is set too high, the application may experience oscillations as the controller tries to control speed. If it is set too low, the motor may take a long time to approach the exact commanded speed.
Ki HS <i>Ki_HS_SpdM</i> 0x301D 0x00	5 – 100% <i>50 – 1000</i>	The Ki HS parameter sets the integral term (Ki) for high vehicle speeds; see description of Ki LS.

Quick Link:

Motor Speed
 Constraints [p.24](#)

1- SPEED MODE — VELOCITY FEEDFORWARD MENU [OPTIONAL]

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Kvff <i>Kvff_SpdM</i> 0x3014 0x00	0 – 500 A <i>0 – 5000</i>	This velocity feedforward term is designed to improve throttle responsiveness and speed controller performance, especially at low speeds. For traction systems, set it to 50–70% of the current needed to maintain a very low speed, unloaded, on flat ground. For a pump system, set it to the lowest load current (i.e., the current running at the minimum load). Alternatively, the responsiveness of a pump speed control loop can be significantly enhanced by using a VCL program to continuously update this parameter to the appropriate value as each pump load is requested.
Build Rate <i>Vel_FF_Build_Rate_SpdM</i> 0x3093 0x00	0.1 – 5.0 s <i>100 – 5000</i>	Determines how fast the Kvff term builds up. For traction systems, if the mechanical gear-backlash is abruptly felt or heard when moving the throttle from neutral to a very small value, slowing the build rate (i.e., setting it to a higher value) will soften the feel. For a pump system, start with this parameter at the minimum setting. Slowing it down (i.e., setting it to a higher value) will reduce speed overshoot if too much feedforward has been commanded.
Release Rate <i>Vel_FF_Release_Rate_SpdM</i> 0x3094 0x00	0.1 – 5.0 s <i>100 – 5000</i>	Determines how fast the Kvff term releases. If the release seems too abrupt, slowing the release rate (i.e., setting it to a higher value) will soften the feel. It should be set fast enough (i.e., at a low enough value) to prevent the vehicle from running on after throttle release.

1- SPEED MODE — ACCELERATION FEEDFORWARD MENU [OPTIONAL]

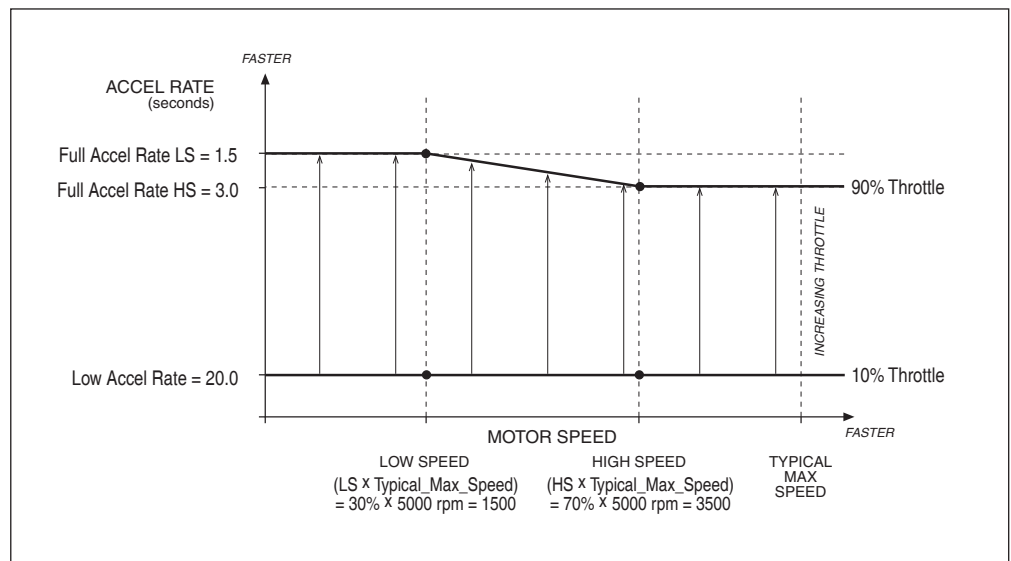
PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Kaff <i>Kaff_SpdM</i> 0x3013 0x00	0 – 500 A <i>0 – 5000</i>	This acceleration feedforward term is designed to improve throttle responsiveness and speed controller performance at all speeds. It can be thought of as a “quick start” function which can enhance responsiveness at all speeds. Using the present accel and decel rates, observe the average current while running at full throttle at low speeds when accelerating without load on flat ground, and set Kaff to 50–70% of that value. NOTE: If any accel rate parameters get changed, this parameter will need to be changed also.
Kbff <i>Kbff_SpdM</i> 0x3019 0x00	0 – 500 A <i>0 – 5000</i>	This braking feedforward term is designed to improve braking responsiveness at all speeds. Using the present decel rates, observe the average current while running at full throttle braking, and set Kbff to that value.
Build Rate <i>Acc_FF_Build_Rate_SpdM</i> 0x3095 0x00	0.1 – 5.0 s <i>100 – 5000</i>	Determines how fast the Kaff and Kbff terms build up. For traction systems, if the mechanical gear-backlash is abruptly felt or heard when moving the throttle from neutral to a very small value, slowing the build rate (i.e., setting it to a higher value) will soften the feel. For a pump system, start with this parameter at the minimum setting. Slowing it down (i.e., setting it to a higher value) will reduce over-shoot if too much feedforward has been commanded.
Release Rate <i>Acc_FF_Release_Rate_SpdM</i> 0x3096 0x00	0.1 – 5.0 s <i>100 – 5000</i>	Determines how fast the Kaff and Kbff terms release. It should be set fast enough (i.e., at a low enough value) to prevent the vehicle from running on after throttle release.

1- SPEED MODE – RESPONSE MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Full Accel Rate HS <i>Full_Accel_Rate_HS_SpdM</i> 0x307F 0x00	0.1 – 30.0 s 100 – 30000	Sets the rate (in seconds) at which the speed command increases when full throttle is applied at high vehicle speeds. Larger values represent slower response. See Figure 7 for relationship between Full Accel Rate HS, Full Accel Rate LS, and Low Accel Rate.
Full Accel Rate LS <i>Full_Accel_Rate_LS_SpdM</i> 0x3016 0x00	0.1 – 30.0 s 100 – 30000	Sets the rate (in seconds) at which the speed command increases when full throttle is applied at low vehicle speeds.
Low Accel Rate <i>Low_Accel_Rate_SpdM</i> 0x3081 0x00	0.1 – 30.0 s 100 – 30000	Sets the rate (in seconds) at which the speed command increases when a small amount of throttle is applied. This rate is typically adjusted to affect low speed maneuverability.
Neutral Decel Rate HS <i>Neutral_Decel_Rate_HS_SpdM</i> 0x3018 0x00	0.1 – 30.0 s 100 – 30000	Sets the rate (in seconds) that is used to slow down the vehicle when the throttle is released to neutral at high vehicle speeds.
Neutral Decel Rate LS <i>Neutral_Decel_Rate_LS_SpdM</i> 0x3070 0x00	0.1 – 30.0 s 100 – 30000	Sets the rate (in seconds) that is used to slow down the vehicle when the throttle is released to neutral at slow vehicle speeds.
Full Brake Rate HS <i>Full_Brake_Rate_HS_SpdM</i> 0x301B 0x00	0.1 – 30.0 s 100 – 30000	Sets the rate (in seconds) at which the vehicle slows down from high speeds when full brake is applied or when full throttle is applied in the opposite direction. See Figure 8 for relationship between Full Brake Rate HS, Full Brake Rate LS, and Low Brake Rate.
Full Brake Rate LS <i>Full_Brake_Rate_LS_SpdM</i> 0x3077 0x00	0.1 – 30.0 s 100 – 30000	Sets the rate (in seconds) at which the vehicle slows down from low speeds when full brake is applied or when full throttle is applied in the opposite direction.
Low Brake Rate <i>Low_Brake_Rate_SpdM</i> 0x301A 0x00	0.1 – 30.0 s 100 – 30000	Sets the rate (in seconds) at which the vehicle slows down at all speeds when a small amount of brake is applied or when a small amount of throttle is applied in the opposite direction. The Low Brake Rate parameter should always be set less than or equal to the Neutral Decel Rate LS, Neutral Decel Rate HS, Full Brake Rate LS and Full Brake Rate HS.

Figure 7

Acceleration response rate diagram. In this example, HS = 70%, LS = 30%, Typ Max Spd = 5000 rpm.



1- SPEED MODE — FINE TUNING MENU

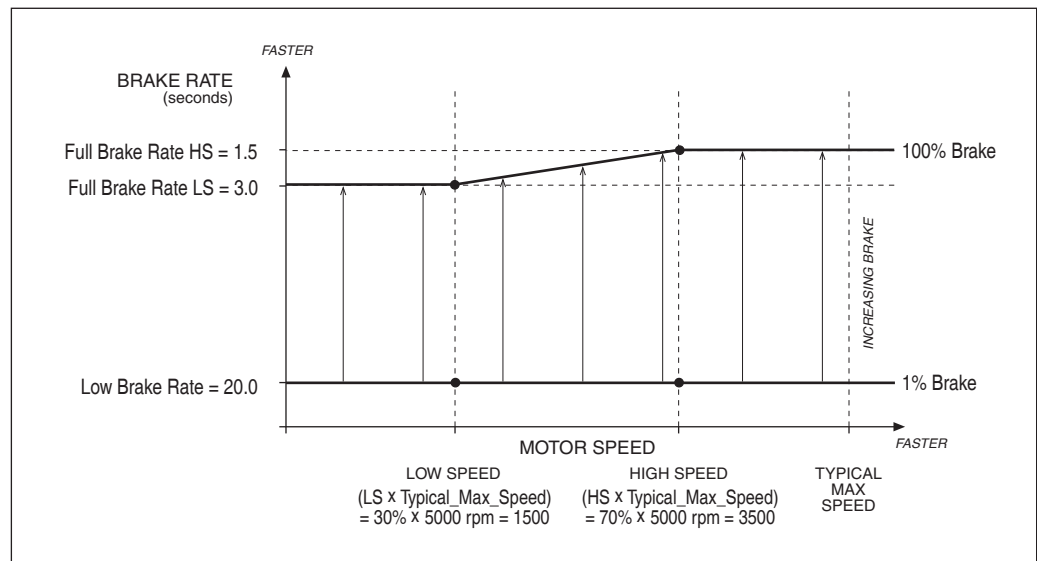
PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Partial Decel Rate <i>Partial_Decel_Rate_SpdM</i> 0x3082 0x00	0.1 – 30.0 s 100 – 30000	Sets the rate (in seconds) that is used to slow down the vehicle when the throttle is reduced without being released to neutral. Larger values represent slower response.
HS (High Speed) <i>HS</i> 0x3076 0x00	0 – 100% 0 – 32767	Sets the percentage of the Typical Max Speed above which the “HS” parameters will be used.
LS (Low Speed) <i>LS</i> 0x3075 0x00	0 – 100% 0 – 32767	Sets the percentage of the Typical Max Speed below which the “LS” parameters will be used.
Reversal Soften <i>Reversal_Soften</i> 0x3074 0x00	0 – 100% 0 – 30000	Larger values create a softer reversal from regen braking to drive when near zero speed. This helps soften the transition when the regen and drive current limits are set to different values. Note: This parameter is not mode-specific and appears in two places. Changing the value of this speed-mode parameter affects the parameter listed in: Program » Torque Mode » Response » Fine Tuning » Reversal Soften.
Max Speed Accel <i>Max_Speed_Accel_SpdM</i> 0x384A 0x00	0.1 – 30.0 s 100 – 30000	In some applications, the Max Speed value is changed frequently, through VCL or over the CANbus. The Max Speed Accel parameter controls the rate at which the maximum speed setpoint is allowed to change when the value of Max Speed is raised. The rate set by this parameter is the time to ramp from 0 rpm to Typical Max Speed rpm. For example, suppose Max Speed is raised from 1000 rpm to 4000 rpm. If Typical Max Speed is 5000 rpm and the rate is 10.0 seconds, it will take $10.0 \times (4000 - 1000) \div 5000 = 6.0$ seconds to ramp from 1000 rpm to 4000 rpm.
Max Speed Decel <i>Max_Speed_Decel_SpdM</i> 0x3846 0x00	0.1 – 30.0 s 100 – 30000	This parameter works like the Max Speed Accel parameter, except that it controls the rate at which the maximum speed setpoint is allowed to change when the value of Max Speed is <u>lowered</u> . For example, suppose the Max Speed is changed from 4500 rpm to 2500 rpm. If Typical Max Speed is 5000 rpm, and the rate is 5.0 seconds, it will take $5.0 \times (4500 - 2500) \div 5000 = 2.0$ seconds to ramp from 4500 rpm to 2500 rpm.
Creep Speed <i>Creep_Speed_SpdM</i> 0x3852 0x00	0 – 1000 rpm 0 – 1000	Sets a minimum speed for steady state operation. The programmed accel and decel rates will still be in affect as the speed slews through the region between this value and zero. A larger value of this parameter will provide a smoother feel, but my impact (increase) brake distance.

Quick Link:

Typical Max Speed [p.56](#)

Figure 8

Braking response rate diagram. In this example,
 HS = 70%,
 LS = 30%,
 Typ Max Spd = 5000 rpm.



1- SPEED MODE — RESTRAINT MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Restraint Forward <i>Restraint_Forward_SpdM</i> 0x3020 0x00	0 – 100% 0 – 32767	Increases torque when on a steep hill in order to limit roll-forward speed. Setting this parameter too high may cause oscillations in the motor as it attempts to limit the roll-forward speed.
Restraint Back <i>Restraint_Back_SpdM</i> 0x301F 0x00	0 – 100% 0 – 32767	Increases torque when on a steep hill in order to limit roll-back speed. Setting this parameter too high may cause oscillations in the motor as it attempts to limit the roll-back speed.
Soft Stop Speed <i>Soft_Stop_Speed</i> 0x3027 0x00	0 – 500 rpm 0 – 500	Defines the speed below which a much slower decel rate is used. A setting of zero disables the function. NOTE: This parameter works only in Speed Mode and Speed Mode Express. Soft Stop Speed is useful for vehicles that have fast deceleration and vehicles operating on ramps using the Position Hold function. With vehicles that have fast deceleration, the driver may find the final speed reduction to zero rpm uncomfortable; the vehicle may even rock back as a result of tire wind-up. Soft Stop Speed allows the vehicle to slow at the same fast rate until it reaches the set threshold, at which point it changes to a slower (softer) deceleration rate. However, if the threshold is set too high, the vehicle will feel like it is “running on.” When throttle is released on a ramp, the vehicle may roll back before Position Hold (see below) takes control. Soft Speed Stop can be used to reduce the amount of rollback, but shouldn’t be set so high the vehicle drives up the ramp after the throttle is released.

1- SPEED MODE — POSITION HOLD MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Position Hold Enable <i>Position_Hold_Enable</i> <i>Position_Hold_Enable_Bit0</i> [Bit 0] 0x3893 0x00	On / Off On / Off [PCF]	Allows the Position Hold mode to be entered at zero throttle when the vehicle comes to a stop. NOTE: EM Brake Type = 2 also enables the Position Hold function.
Position Hold Timeout Time <i>Position_Hold_Timeout_Time</i> 0x388C 0x00	0.0 – 20.0 s 0 – 625	This parameter plus the Position Hold Settling Time parameter sets the maximum time the vehicle will stay in Position Hold before releasing the hold and going into Restraint mode. Setting the parameter to zero disables this timeout function, which means the Position Hold will be held. Activating the interlock resets the timer.
Kp <i>Kp_Position_Hold</i> 0x388A 0x00	2 – 100% 82 – 2048	Determines the stiffness with which position is regulated when in Position Hold mode. High Kp will produce less rollback on a ramp, but more bouncing; see Kd below. Too much Kp will cause instability.
Kd <i>Kd_Position_Hold</i> 0x388D 0x00	0 – 100% 0 – 8192	Determines the damping in Position Hold mode. Some damping must be present in the control system to keep the vehicle from oscillating slowly (“bouncing”). High Kd will improve the dynamic response of the Position Hold controller, but too much Kd will cause fast instability.
Zero Speed Threshold <i>Zero_Speed_Threshold</i> 0x306F 0x00	5 – 100 rpm 5 – 100	Determines the speed below which the EM brake will be commanded to set. Setting this speed too high may cause a jerky stop when the EM brake sets and stops the motor. This parameter appears twice in the menu structure. Changing the value of this parameter affects this (same) parameter in the EM Brake Control menu.
Zero Speed Threshold Time <i>Zero_Speed_Threshold_Time</i> 0x30DA 0x00	0 – 480 ms 0 – 60	Determines how long motor speed must be below Zero_Speed_Threshold to declare zero speed. This parameter appears twice in the menu structure. Changing the value of this parameter affects this (same) parameter in the EM Brake Control menu.
Position Hold Settling Time <i>Position_Hold_Setting_Time</i> 0x3024 0x00	0 – 5000 ms 0 – 156	Determines how long the position hold function is allowed to operate before the EM brake is set. This time should be set long enough for the position hold to settle. This parameter appears twice in the menu structure. Changing the value of this parameter affects this (same) parameter in the EM Brake Control menu.
Entry Rate <i>Entry_Rate_Position_Hold</i> 0x383B 0x00	5 – 100% 50 – 1000	When the vehicle transitions from forward speed to reverse speed or from reverse speed to forward speed (for example when coming to a stop going up a steep ramp) Position Hold is automatically entered immediately at zero speed regardless of this parameter. This parameter applies when the vehicle needs to be brought to a stop without the assistance of gravity (for example when moving forward down a ramp). This rate determines how quickly zero speed is attained after the ramped speed request reaches zero. Setting this parameter too high will make the stop seem very abrupt and may even cause the vehicle to roll back slightly. When the parameter is set lower the vehicle take longer to come to a stop and enter Position Hold mode.
Exit Rollback Reduction <i>Exit_Rollback_Reduction</i> 0x3844 0x00	0 – 100% 0 – 2048	This function is applicable when the Torque Preload Enable function has been Disabled (Off) or the Torque Preload Cancel Delay timer has expired (see EM Brake menu). Exit Rollback Reduction is only intended for use with EM Brake Type = 2, for EM Brake Type = 1 (or 0) set Exit Rollback Reduction = 0. Exit Rollback Reduction introduces an additional control function for the speed controller to reduce rollback on a ramp after a throttle command is applied from a stop. For example, suppose the vehicle is on a ramp facing upwards and, after a forward throttle request, the vehicle rolls back slightly before climbing the ramp (again, assuming the torque preload function is inactive). As the vehicle rolls back, this additional term will be added to the torque request until forward speed is sensed, to attempt to minimize rollback. The effect increases with percentage, and is disabled when = 0%. Setting the value too high may introduce temporary oscillations. If the Torque Preload Cancel Delay parameter expires before the throttle is re-engaged, the torque preload [Position Hold] memory will be cleared. In such a case, setting Exit Rollback Reduction to > 0% will reduce unintended roll.

Quick Links:

EM Brake Type [p.50](#)Torque Preload Enable [p.51](#)Torque Preload Cancel Delay [p.51](#)

1- SPEED MODE — PUMP PARAMETERS

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Pump Enable* AC_Pump_Enable_SpdM AC_Pump_Enable_SpdM_Bit0 [Bit 0] 0x3896 0x00	On / Off On / Off	This parameter should be programmed On to operate a pump motor rather than a vehicle drive motor. Speed controller responsiveness and stability are enhanced.
Regen Lower Enable* Regen_Lower_Enable_SpdM Regen_Lower_Enable_SpdM_Bit0 [Bit 0] 0x3877 0x00	On / Off On / Off	This parameter works together with Pump Enable as follows: 1) When Pump Enable = On and Regen Lower Enable = On, the pump motor can turn in both the forward and the reverse direction. In this case, the pump can be used as the Lower function (reverse). The pump motor, when “driven in reverse by a load” will also regen current in a similar fashion to a drive-motor regenerative braking. 2) When Pump Enable = On and Regen Lower Enable = Off, the pump motor can turn only in the forward direction. In this case, a hydraulic valve is typically used for the Lower function. 3) When Pump Enable = Off, the Regen Lower Enable parameter has no effect on the control system.

* This parameter appears twice in the parameter menu structure. Changing the value of this parameter affects the same parameter in Speed Mode Express.

2 – TORQUE MODE — SPEED LIMITER MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Max Speed Max_Speed_TrqM 0x3021 0x00	500 – 10000 rpm 500 – 10000	Defines the maximum allowed motor rpm for torque control mode (independent of throttle position). In torque control mode, full throttle requests 100% of the available torque. Partially-applied throttle is scaled proportionately; e.g., 40% applied throttle corresponds to a request for 40% of the available torque. NOTE: The maximum motor rpm is subject to the constraints.
Kp Kp_TrqM 0x3023 0x00	0 – 100% 0 – 8192	Determines how aggressively the speed controller attempts to limit the speed of the motor to Max Speed. Larger values provide tighter control. If Kp is set too high, the application may experience oscillations as the controller tries to control speed. Setting Kp too low may result in a top speed much higher than Max Speed.
Ki Ki_TrqM 0x3025 0x00	5 – 100% 50 – 1000	The integral term (Ki) forces zero steady state error, so the motor speed will be limited to Max Speed. Larger values provide faster control. If the gain is set too high, application may experience oscillations as the controller tries to limit speed. If it is set too low, it may take a long time for the motor to approach Max Speed from overspeed.
Kd Kd_TrqM 0x3836 0x00	0 – 100% 0 – 8192	Provides damping as the vehicle approaches top speed, thereby reducing overshoot. If Kd is set too high, the vehicle may take too long to reach top speed. If Kd is set too low, the vehicle may overshoot top speed, especially when traveling downhill.

Quick Link:

Motor Speed Constraints [p.24](#)

2 - TORQUE MODE — RESPONSE MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Accel Rate <i>Accel_Rate_TrqM</i> 0x3026 0x00	0.1 – 30.0 s 100 – 30000	Sets the rate (in seconds) at which the motor torque increases to full when full throttle is applied. Larger values represent slower response.
Accel Release Rate <i>Accel_Release_Rate_TrqM</i> 0x3028 0x00	0.1 – 2.0 s 100 – 2000	Determines how quickly deceleration will be initiated when the throttle is released while the vehicle is still accelerating. If the release rate is fast (i.e., set to a low value), the transition is initiated abruptly. The transition is smoother if the release rate is set to a higher value (slower transition); however, setting the rate too high can cause the vehicle to feel uncontrollable when the throttle is released, as it will continue to drive for a short time.
Brake Rate <i>Brake_Rate_TrqM</i> 0x3029 0x00	0.1 – 5.0 s 100 – 5000	Adjusts the rate (in seconds) at which braking torque builds as the vehicle transitions from drive to braking when direction is reversed, the brake pedal is applied, or neutral braking begins. Lower values represent faster times and therefore faster braking; gentler braking is achieved by setting the braking rate to a higher value.
Brake Release Rate <i>Brake_Release_Rate_TrqM</i> 0x3831 0x00	0.1 – 2.0 s 100 – 2000	Adjusts the rate (in seconds) at which braking torque releases as the vehicle transitions from braking to drive.
Neutral Braking <i>Neutral_Braking_TrqM</i> 0x302E 0x00	0 – 100% 0 – 32767	Neutral braking occurs progressively when the throttle is reduced toward the neutral position or when no direction is selected. The neutral braking parameter is adjustable from 0 to 100% of the regen current limit.
Neutral Taper Speed <i>Neutral_Taper_Speed_TrqM</i> 0x302F 0x00	200 – 6000 rpm 200 – 6000	Determines the motor speed below which neutral braking current is adjusted when throttle is reduced; see Figure 9. The neutral braking current is linearly reduced by the product of the parameters, Neutral Braking x Regen Current Limit, at the Neutral Taper Speed to the Creep Torque current at zero rpm motor speed. NOTE: Setting the taper speed too low may cause oscillations in the motor.
Forward Full Restraint Speed <i>Forward_Full_Restraint_Speed_TrqM</i> 0x386D 0x00	100 – 32000 rpm 100 – 32000	Sets the speed point at which the full regen current will be applied to restrain the vehicle from rolling forward. Although this speed is never actually reached, it does set the slope of the restraint strength and can be thought of as a gain; see Figure 9. Setting this parameter too low can cause oscillations.
Back Full Restraint Speed <i>Back_Full_Restraint_Speed_TrqM_TrqM</i> 0x386E 0x00	100 – 32000 rpm 100 – 32000	Sets the speed point at which the full regen current will be applied to restrain the vehicle from rolling in reverse (backward). Although this speed is never actually reached, it does set the slope of the restraint strength and can be thought of as a gain; see Figure 9. Setting this parameter too low can cause oscillations.

Quick Link:

Current Limits menu [p.44](#)

2 - TORQUE MODE — FINE TUNING MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Creep Torque <i>Creep_Torque_TrqM</i> 0x3071 0x00	0 – 100% 0 – 32767	Determines the amount of torque applied to the vehicle at a stop with no throttle input, to emulate the feel of an automatic transmission automobile; see Fig. 9. WARNING! When interlock is engaged, creep torque allows vehicle propulsion if a direction is selected even though no throttle is applied. Care should be taken when setting up this parameter. If pedal braking is enabled, creep torque is progressively disabled as brake is applied so as to prevent the motor from driving into the brakes and thus wasting energy. Creep Torque and Neutral Taper Speed interact to create the slope of the torque response as the vehicle approaches zero speed; see Figure 9. If the vehicle oscillates as it coasts down toward zero speed, try lowering Creep Torque or increasing Neutral Taper Speed.
Brake Full Creep Cancel <i>Brake_Full_Creep_Cancel_TrqM</i> 0x3837 0x00	25 – 100% 8192 – 32767	Determines the amount of brake pedal input that will fully cancel the creep torque. Amount of cancellation is proportional to the brake input.
Creep Build Rate <i>Creep_Build_Rate_TrqM</i> 0x3832 0x00	0.1 – 5.0 s 100 – 5000	Determines how fast the programmed creep torque builds when a direction is selected.
Creep Release Rate <i>Creep_Release_Rate_TrqM</i> 0x3833 0x00	0.1 – 5.0 s 100 – 5000	Determines how fast the programmed creep torque releases when the brake is cancelling the creep torque or when the direction switches are cleared (neutral).
Gear Soften <i>Gear_Soften_TrqM</i> 0x302A 0x00	0 – 100% 0 – 5000	Adjusts the throttle take-up from linear (0% setting) to an S curve. Larger values create softer throttle take-up, in forward and reverse. Softening is progressively reduced at higher speeds; see Figure 10.
Brake Taper Speed <i>Brake_Taper_Speed_TrqM</i> 0x300F 0x00	200 – 6000 rpm 200 – 6000	Determines the motor speed below which the maximum braking current is linearly reduced from 100% to 0% at zero speed; see Figure 11. Setting the taper speed too low for the braking current will cause oscillations in the motor as it attempts to brake the vehicle to a stop on very steep slopes. Taper speed is applicable only in response to brake pedal input; it does not affect direction reversal braking or neutral braking. If the vehicle is in restraint when the brake is pressed, the applied braking torque is affected by both Brake Taper Speed and Forward (or Back) Full Restraint Speed. If the vehicle oscillates in this mode, it may be necessary to increase one or more of these parameters.
Reversal Soften <i>Reversal_Soften</i> 0x3074 0x00	0 – 100% 0 – 3000	Larger values create a softer reversal from regen braking to drive when near zero speed. This helps soften the transition when the regen and drive current limits are set to different values. Note: This parameter is not mode-specific and appears in two places. Changing the value of this torque-mode parameter affects the parameter listed in: Program » Speed Mode » Response » Fine Tuning » Reversal Soften.
Max Speed Decel <i>Max_Speed_Decel_TrqM</i> 0x3835 0x00	0.1 – 30.0 s 100 – 30000	In some applications, the Max Speed value is changed frequently, through VCL or over the CANbus. The Max Speed Decel parameter controls the rate at which the maximum speed setpoint is allowed to change when the value of Max Speed is lowered. The rate set by this parameter is the time to ramp from Typical Max Speed rpm to 0 rpm. For example, suppose the Max Speed is changed from 3000 rpm to 1000 rpm. If Typical Max Speed is 5000 rpm, and the rate is 5.0 seconds, it will take $5.0 \times (3000 - 1000) \div 5000 = 2.0$ seconds to ramp from 3000 rpm to 1000 rpm.

Quick Link:Brake Pedal Enable [p.49](#)

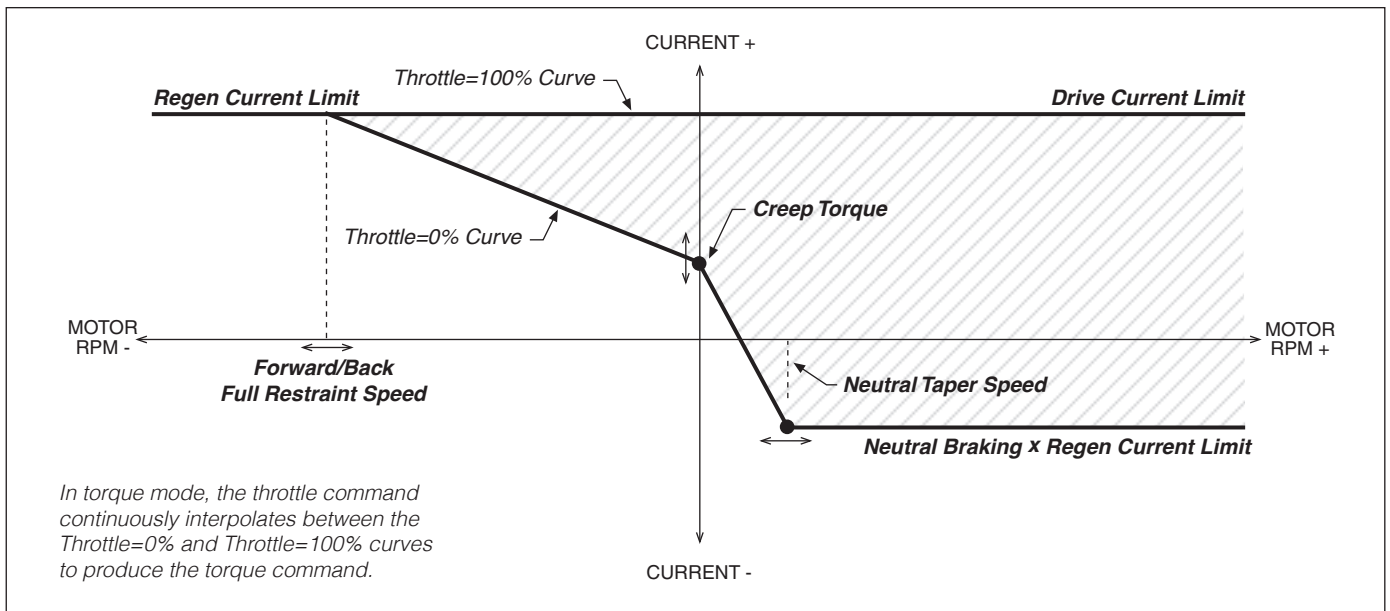


Figure 10
Effect of Gear Soften parameter
(torque control mode).

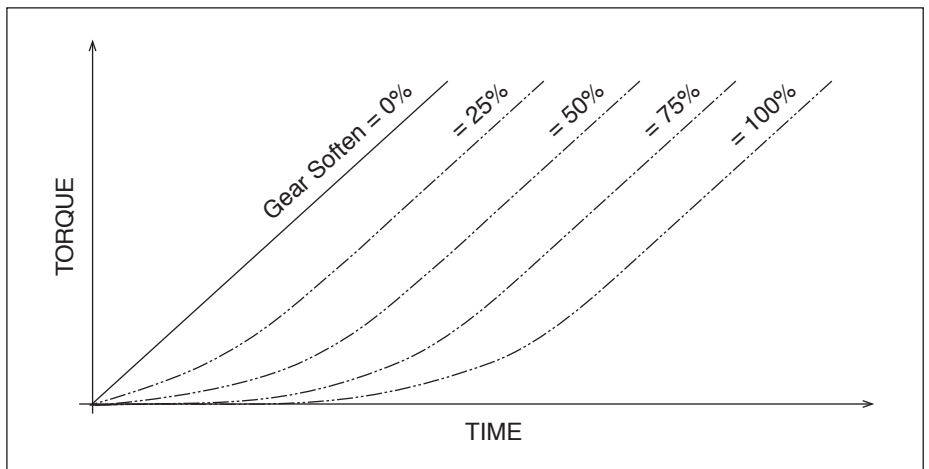
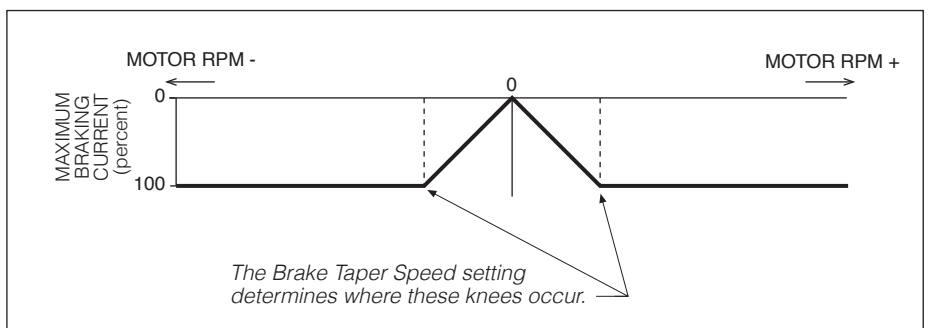


Figure 11
Effect of Brake Taper Speed
parameter (torque control mode).



CURRENT LIMITS MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Drive Current Limit <i>Drive_Current_Limit</i> 0x305B 0x00	5 – 100% 1638 – 32767	Sets the maximum RMS current the controller will supply to the motor during drive operation, as a percentage of the controller's full rated current.* Reducing this value will reduce the maximum drive torque. NOTE: Changing this parameter requires the re-setting of the ACIM FW Base Speed parameter.
Regen Current Limit <i>Regen_Current_Limit</i> 0x305C 0x00	5 – 100% 1638 – 32767	Sets the maximum RMS regen current, as a percentage of the controller's full rated current.* The regen current limit applies during neutral braking, direction reversal braking, and speed limiting when traveling downhill.
Brake Current Limit <i>Brake_Current_Limit</i> 0x305D 0x00	5 – 100% 1638 – 32767	Sets the maximum RMS regen current during braking when a brake command is given, as a percentage of the controller's full rated current.* Typically the brake current limit is set equal to the regen current limit. The brake current limit overrides the regen current limit when the brake input is active.
EMR Current Limit <i>EMR_Current_Limit</i> 0x3037 0x00	5 – 100% 1638 – 32767	Sets the maximum RMS current allowed for braking and drive when in emergency reverse. The emergency reverse current limit is a percentage of the controller's full rated current.*
Interlock Brake Current Limit <i>Interlock_Brake_Current_Limit</i> 0x309D 0x00	5 – 100% 1638 – 32767	Sets the maximum RMS regen current during interlock braking, as a percentage of the controller's full rated current.*

* The full rated current depends on the controller model;
 see specifications in Table D-1 for the rated current of the controller model number.

POWER LIMITING MAP MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
PL Nominal Speed <i>PL_Nominal_Speed</i> 0x305E 0x00	100 – 8000 rpm 100 – 8000	Sets the base speed that will be used in the drive limiting map and regen limiting map. NOTE: Changing this parameter requires the re-setting of the ACIM FW Base Speed parameter.
Delta Speed <i>PL_Delta_Speed</i> 0x305F 0x00	50 – 1000 rpm 50 – 1000	Sets the width of the delta increment that will be used in the drive limiting map and regen limiting map.

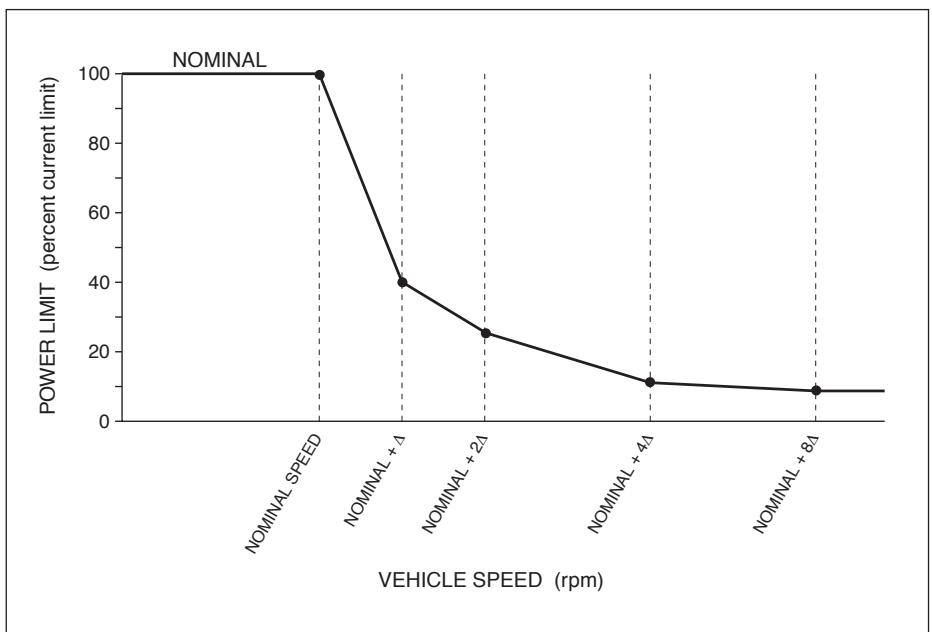
Quick Link:
[Base Speed p.58](#)

DRIVE LIMITING MAP MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Nominal <i>PL_Drive_Nominal</i> 0x3060 0x00	0 – 100% 0 – 32767	These parameters define the percentage of drive current limit that will be applied at the speeds defined by the nominal speed and delta speed parameters. The resulting map allows the controller to reduce the drive current as a function of speed. Reducing the power requirements at certain speeds restricts performance. This can be useful for reducing motor heating. It can also be used to keep consistent vehicle power with changing battery state-of-charge. Figure 12 illustrates these parameters' typical usage.
Plus Delta <i>PL_Drive_Nominal_Plus_Delta</i> 0x3061 0x00	0 – 100% 0 – 32767	
Plus 2xDelta <i>PL_Drive_Nominal_Plus_2xDelta</i> 0x3062 0x00	0 – 100% 0 – 32767	
Plus 4xDelta <i>PL_Drive_Nominal_Plus_4xDelta</i> 0x3063 0x00	0 – 100% 0 – 32767	
Plus 8xDelta <i>PL_Drive_Nominal_Plus_8xDelta</i> 0x3064 0x00	0 – 100% 0 – 32767	

Figure 12

Drive current limiting map
(typical example).



REGEN LIMITING MAP MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Nominal <i>PL_Regen_Nominal</i> 0x3065 0x00	0 – 100% 0 – 32767	These parameters define the percentage of regen current limit or braking current limit that will be applied at the speeds defined by the nominal speed and delta speed parameters. The curve can be shaped to limit the available torque at various speeds. One possible use is to compensate for the torque-speed characteristic of the motor. Figure 13 illustrates two typical usages of these parameters.
Plus Delta <i>PL_Regen_Nominal_Plus_Delta</i> 0x3066 0x00	0 – 100% 0 – 32767	
Plus 2xDelta <i>PL_Regen_Nominal_Plus_2xDelta</i> 0x3067 0x00	0 – 100% 0 – 32767	
Plus 4xDelta <i>PL_Regen_Nominal_Plus_4xDelta</i> 0x3068 0x00	0 – 100% 0 – 32767	
Plus 8xDelta <i>PL_Regen_Nominal_Plus_8xDelta</i> 0x3069 0x00	0 – 100% 0 – 32767	

Figure 13

Regen current limiting map
(two examples).

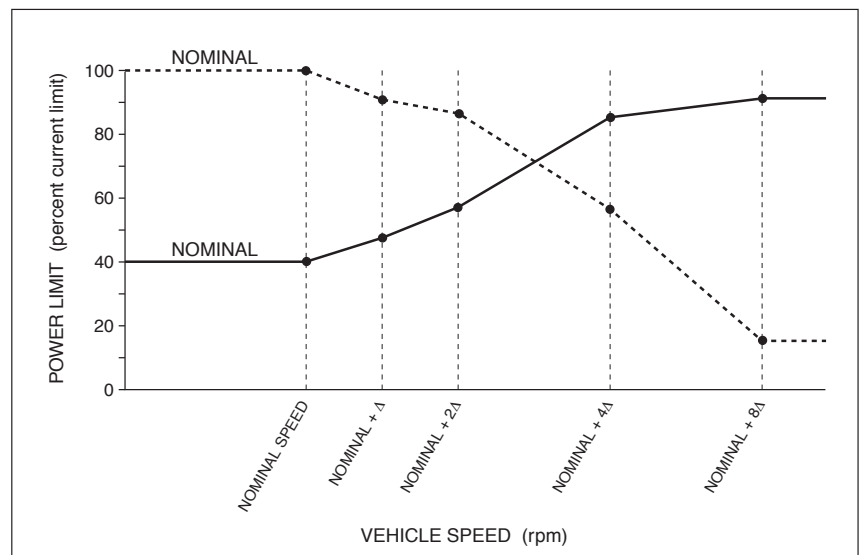


Figure 14

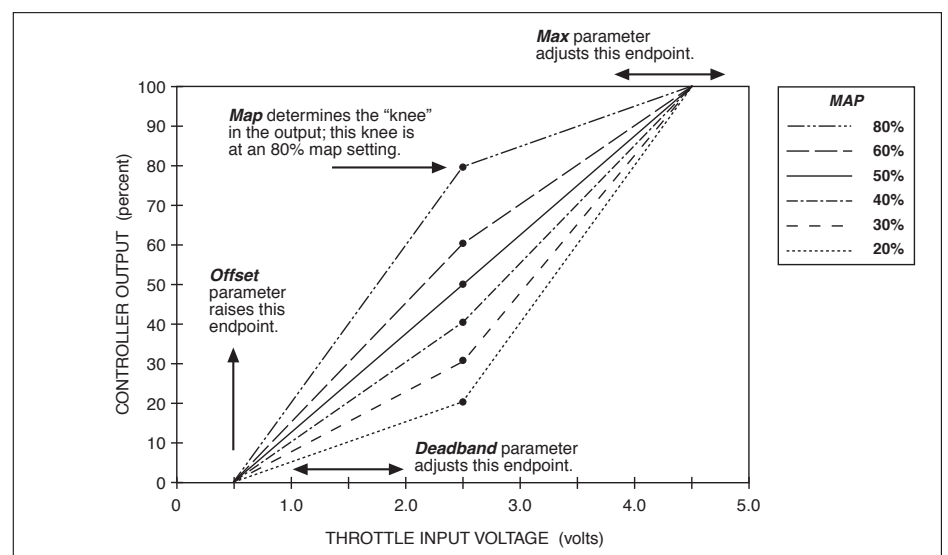
Effect of throttle adjustment parameters. Together these four generic parameters determine the controller's response to throttle demand (in forward or reverse) and to brake demand.

In the examples shown in this figure,

Deadband = 0.5V

Max = 4.5V

Offset = 0.



THROTTLE MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Throttle Type <i>Throttle_Type</i> 0x3000 0x00	1 – 5 <i>1 – 5</i> [PCF]	These E and SE controllers accept a variety of throttle inputs. The throttle type parameter can be programmed as follows: 1. 2-wire rheostat, 5 kΩ–0 input. 2. <u>Single-ended</u> 3-wire 1 kΩ–10 kΩ potentiometer, or 0–5 V voltage source. 3. 2-wire rheostat, 0–5 kΩ input. 4. <u>Wigwag</u> 3-wire 1 kΩ–10 kΩ potentiometer, or 0–5 V voltage source. 5. VCL input (<i>VCL_Throttle</i>). NOTE: Do not change this parameter while the controller is powering the motor. Any time this parameter is changed a Parameter Change Fault (fault code 49) is set and must be cleared by cycling power; this protects the controller and the operator.
Forward Deadband <i>Forward_Deadband</i> 0x3001 0x00	0.00 – 5.00V <i>0 – 32767</i>	Defines the wiper voltage at the throttle deadband threshold. Increasing the throttle deadband setting will increase the neutral range. This parameter is especially useful with throttle assemblies that do not reliably return to a well-defined neutral point, because it allows the deadband to be defined wide enough to ensure that the controller goes into neutral when the throttle mechanism is released.
Forward Map <i>Forward_Map</i> 0x3002 0x00	0 – 100% <i>0 – 32767</i>	Modifies the vehicle's response to the throttle input. Setting the throttle map at 50% provides a linear output response to throttle position. Values below 50% reduce the controller output at low throttle settings, providing enhanced slow speed maneuverability. Values above 50% give the vehicle a faster, more responsive feel at low throttle settings. The map value is the percentage of controller output at half throttle [(deadband + max)/2].
Forward Max <i>Forward_Max</i> 0x3003 0x00	0.00 – 5.00V <i>0 – 32767</i>	Defines the wiper voltage required to produce 100% controller output. Decreasing the throttle max setting reduces the wiper voltage and therefore the full stroke necessary to produce full controller output. This parameter allows reduced-range throttle assemblies to be accommodated.
Forward Offset <i>Forward_Offset</i> 0x3004 0x00	0 – 100% <i>0 – 32767</i>	Defines the initial controller output generated when the throttle is first rotated out of the neutral deadband. For most vehicles, a setting of 0 is appropriate. For heavy vehicles, however, increasing the offset may improve controllability by reducing the amount of throttle required to start the vehicle moving.
Reverse Deadband <i>Reverse_Deadband</i> 0x3005 0x00	0.00 – 5.00V <i>0 – 32767</i>	The four Throttle Reverse parameters are the same as their Throttle Forward counterparts, and apply when the throttle direction is reversed.
Reverse Map <i>Reverse_Map</i> 0x3006 0x00	0 – 100% <i>0 – 32767</i>	
Reverse Max <i>Reverse_Max</i> 0x3007 0x00	0.00 – 5.00V <i>0 – 32767</i>	
Reverse Offset <i>Reverse_Offset</i> 0x3008 0x00	0 – 100% <i>0 – 32767</i>	
Throttle Filter <i>Throttle_Filter</i> 0x3030 0x00	0.5 – 125.0 Hz 131 – 32767	Sets the low pass filter cutoff frequency for the throttle pot wiper input. Higher values will make the throttle more responsive to quick changes. Lower values will make the throttle less responsive to electrical noise.

NOTE: All four throttle adjustment parameters — Deadband, Map, Max, Offset — condition the raw throttle voltage into a single % throttle command, as shown in Figure 14 on the previous page.

THROTTLE MENU, cont'd

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
HPD SRO Type <i>HPD_SRO_Type</i> 0x30CF 0x00	0 – 3 <i>0 – 3</i> [PCF]	Determines the type of HPD (High Pedal Disable) / SRO (Static Return to Off) protection. One type of checks is available for material-handling vehicles, and two types for golf-style vehicles. If any of the HPD/SRO checks finds an input sequencing problem, an HPD/Sequencing Fault (flash code 47) is set. 0. HPD/SRO feature is disabled. 1. HPD/SRO enabled for material-handling vehicles. HPD: If throttle input is received before interlock input. SRO: If direction input is received before interlock input. The HPD/SRO check is made when the interlock input changes from Off to On. If the throttle input >25% or a direction input is On, an HPD/Sequencing Fault is set. The HPD/Sequencing Fault is cleared by returning the throttle input to <25% or the direction inputs to Off. 2. Golf-style HPD that allows direction reversal while driving. HPD: Throttle input received before interlock, or throttle input received before direction input while vehicle is stationary. SRO: None. The HPD check is made when the interlock input or direction inputs are Off and the vehicle is stationary. If the throttle input >25%, an HPD/Sequencing Fault is set. No SRO check is made with this type, so the order of the interlock and direction inputs does not matter. The HPD/Sequencing Fault is cleared by returning the throttle input to <25% or the direction inputs to Off. 3. Golf-style HPD that prevents direction reversal while driving. HPD: If throttle input is received before interlock or direction input. SRO: None. HPD check is made when the interlock input or direction inputs are Off. If the throttle input >25%, an HPD/Sequencing Fault is set. The check is done regardless of vehicle speed, so reversing direction with throttle input >25% will result in a fault. No SRO check is made with this type, so the order of the interlock and direction inputs does not matter. The HPD/Sequencing Fault is cleared by returning the throttle input to <25% or the direction inputs to Off.
Sequencing Delay <i>Sequencing_Delay</i> 0x3009 0x00	0.0 – 5.0 s <i>0 – 1250</i>	Typically the sequencing delay feature allows the interlock switch to be cycled within a set time (the defined sequencing delay), thus preventing inadvertent activation of HPD/SRO. This feature is especially useful in applications where the interlock switch may bounce or be momentarily cycled during operation.
VCL Throttle Enable <i>VCL_Throttle_Enable</i> <i>VCL_Throttle_Enable_Bit0 [Bit 0]</i> 0x3099 0x00	On / Off <i>On / Off</i>	When programmed On, the throttle processing with fault detection will operate normally; however, the throttle command will require VCL to define the connection between the OS_Throttle and VCL_Throttle variables. This allows VCL flexibility and customization of throttle processing, while still allowing Throttle_Type 1–4 with throttle fault detection. Figure 15 illustrates these throttle chains.

Quick Link:Figure 15 [p.114](#)

BRAKE MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Brake Pedal Enable <i>Brake_Pedal_Enable</i> <i>OptionBits1 [Bit 3]</i> 0x306A 0x00	On / Off On / Off	Determines whether the brake input and algorithm are enabled, making the brake throttle part of the motor control command.
Brake Type <i>Brake_Type</i> 0x300A 0x00	1 – 5 1 – 5 [PCF]	These E and SE controllers accept a variety of brake inputs. The brake type parameter can be programmed as follows: 1. 2-wire rheostat, 5 kΩ–0 input. 2. Single-ended 3-wire 1 kΩ–10 kΩ potentiometer, 0–5 V voltage source. 3. 2-wire rheostat, 0–5 kΩ input. 4. (Not applicable). 5. VCL input (<i>VCL_Brake</i>). NOTE: Do not change this parameter while the controller is powering the motor. Any time this parameter is changed a Parameter Change Fault (fault code 49) is set and must be cleared by cycling power; this protects the controller and the operator.
Brake Deadband <i>Brake_Deadband</i> 0x300B 0x00	0.00 – 5.00 V 0 – 32767	The four Brake throttle adjustment parameters are the same as their Drive throttle counterparts; see descriptions and Figure 14.
Brake Map <i>Brake_Map</i> 0x300C 0x00	0 – 100% 0 – 32767	
Brake Max <i>Brake_Max</i> 0x300D 0x00	0.00 – 5.00 V 0 – 32767	
Brake Offset <i>Brake_Offset</i> 0x300E 0x00	0 – 100% 0 – 32767	
Brake Filter <i>Brake_Filter</i> 0x3031 0x00	0.5 – 125.0 Hz 131 – 32767	Sets the low pass filter cutoff frequency for the pot2 wiper input. Higher values will make the brake more responsive to quick changes. Lower values will make the brake less responsive to electrical noise.
VCL Brake Enable <i>VCL_Brake_Enable</i> <i>VCL_Brake_Enable_Bit0 [Bit 0]</i> 0x301E 0x00	On / Off On / Off	When programmed On, the brake processing with fault detection will operate normally; however, the brake command will require VCL to define the connection between the OS_Brake and VCL_Brake variables. This allows VCL flexibility and customization of throttle processing, while still allowing Brake_Type 1–3 with brake fault detection. Figure 15 illustrates these brake chains.

Quick Link:

Figure 15 [p.114](#)

EM BRAKE CONTROL MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Brake Type <i>EM_Brake_Type</i> 0x3097 0x00	0 – 2 0 – 2 [PCF]	This brake type parameter determines how the EM brake responds to the interlock input, throttle, and vehicle motor speed. 0. EM brake function disabled. The EM brake driver (PWM2) is released to general I/O use with VCL. 1. EM brake controlled by interlock. The controller will command the EM brake to release whenever the interlock is closed (Interlock = On). If interlock braking is enabled and the interlock opens when the vehicle is moving at motor speed greater than Zero_Speed_Threshold, the controller will brake the vehicle to a stop (with interlock braking) and then command the EM brake to set. If the vehicle motor speed is less than this threshold, the EM brake will engage after the Sequencing_Delay has expired. If interlock braking is disabled, the EM brake will engage after the Sequencing_Delay has expired. 2. EM brake controlled by interlock and neutral. The controller will command the EM brake to set whenever the throttle command is zero and motor speed is less than Zero_Speed_Threshold. Position Hold will be enabled automatically.
Pull In Voltage <i>EM_Brake_Pull_In_Voltage</i> 0x3072 0x00	0 – 100% 0 – 32767	The EM brake pull-in voltage allows a high initial voltage when the EM brake 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 controls whether the pull-in and holding voltages are battery voltage compensated.
Holding Voltage <i>EM_Brake_Holding_Voltage</i> 0x3098 0x00	0 – 100% 0 – 32767	The EM brake holding voltage allows a reduced average voltage to be applied to the brake coil once the brake has been released. This parameter must be set high enough to hold the brake released under all shock and vibration conditions the vehicle will be subjected to. Note: The Battery Voltage Compensated parameter controls whether the pull-in and holding voltages are battery voltage compensated.
Battery Voltage Compensated <i>EM_Brake_Battery_Voltage_Compensated</i> <i>EM_Brake_Battery_Voltage_Compensated_Bit0</i> [Bit 0] 0x302D 0x00	On / Off On / Off	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 compensated relative to the set Nominal Voltage. 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.
Set EM Brake On Fault <i>EM_Brake_Set_Upon_Fault</i> <i>EM_Brake_Set_Upon_Fault_Bit0</i> [Bit 0] 0x309B 0x00	On / Off On / Off	When programmed On, the controller's operating system will set the electromagnetic brake when a fault occurs that has a fault action of ShutdownEMBrake. See the Troubleshooting Chart, Chapter 10, for a list of all the faults that have a fault action of ShutdownEMBrake.
Zero Speed Threshold <i>Zero_Speed_Threshold</i> 0x306F 0x00	5 – 100 rpm 5 – 100	Determines the speed below which the EM brake will be commanded to set. Setting this speed too high may cause a jerky stop when the EM brake sets and stops the motor. This parameter appears twice in the menu structure. Changing the value of this parameter affects this (same) parameter in the Position Hold menu.
Zero Speed Threshold Time <i>Zero_Speed_Threshold_Time</i> 0x30DA 0x00	0 – 480 ms 0 – 60	Determines how long motor speed must be below Zero_Speed_Threshold to declare zero speed. This parameter appears twice in the menu structure. Changing the value of this parameter affects this (same) parameter in the Position Hold menu.

Quick Links:

[Battery Menu](#) [p.68](#)[Position Hold menu](#) [p.39](#)

EM BRAKE CONTROL MENU, cont'd

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Position Hold Settling Time <i>Position_Hold_Setting_Time</i> 0x3024 0x00	0 – 5000 ms 0 – 156	Determines how long the position hold function is allowed to operate before the EM brake is set. This time should be set long enough for the position hold to settle. This parameter appears twice in the menu structure. Changing the value of this parameter affects this (same) parameter in the Position Hold menu. NOTE *
Brake Set Time <i>EM_Brake_Set_Time</i> 0x30DB 0x00	40 – 2000 ms 5 – 250	Estimated time for the EM brake to physically set after the holding voltage is released. This determines how long the controller waits, after removing voltage from the brake, before released the torque. This should be set longer than the actual brake setting time to ensure the vehicle doesn't move before the brake fully engages.
Torque Release Time <i>EM_Brake_Torque_Release_Time</i> 0x30DC 0x00	40 – 2000 ms 5 – 250	Time to release torque after EM Brake has set and motor has stopped (EMBrakeEngagedAndStopped state).
Brake Release Time <i>EM_Brake_Release_Time</i> 0x3073 0x00	40 – 2000 ms 5 – 250	Estimated time for the EM brake to physically release after the pull-in voltage is applied. This is used to ensure the position hold torque buildup is complete before the brake releases. When set too low, the vehicle may experience rollback on EM brake release.
Torque Preload Time <i>EM_Brake_Torque_Preload_Time</i> 0x3090 0x00	0 – 800 ms 0 – 100	Estimated worst-case time to build up the torque required to hold the vehicle stationary on a hill prior to EM brake release. This is used in conjunction with Release Delay to determine when to release the brake and allow the speed request to slew away from zero. NOTE *
Torque Preload Enable <i>EM_Brake_Torque_Preload_Enable</i> <i>EM_Brake_Torque_Preload_Enable_Bit0 [Bit 0]</i> 0x389D 0x00	On / Off On / Off	When enabled, this function eliminates rollback when the throttle is re-engaged on a ramp by forcing the vehicle to first enter position-hold before setting the EM brake, and then “remembering” the amount of torque that was necessary to hold it on the ramp. When throttle is re-engaged, this value is loaded in the motor before the EM brake is released. The torque value is cleared automatically when KSI power is cycled. Off = When a valid throttle input is received, the speed controller will start with no torque preload as soon as the Release Delay expires. This will allow some rollback when the EM brake releases. This rollback can be reduced by raising the Exit Rollback Reduction parameter in the Position Hold Menu. On = When a valid throttle input is received, the speed controller will start with a pre-set torque as measured by position-hold when the vehicle came to a stop. NOTE *
Save Torque Preload <i>Save_Torque_Preload</i> <i>Save_Torque_Preload_Bit0 [Bit 0]</i> 0x38C9 0x00	On/Off On/Off	This parameter controls whether the EM Brake torque preload is saved over a keyswitch cycle. If Torque_Preload_Cancel_Delay is nonzero and this parameter = On, the timer starts again upon startup (KSI cycle) such that the countdown is from the full value of the Torque_Preload_Cancel_Delay parameter.
Torque Preload Cancel Delay <i>EM_Brake_Torque_Preload_Cancel_Delay</i> 0x3091 0x00	0 – 120 s 0 – 15000	The timer starts after the EM brake is set. If the timer expires before the throttle is re-engaged, the torque preload memory will be cleared. Setting this parameter to zero disables the timer, i.e., the preload is never cancelled. The purpose of this delay is to prevent the vehicle from lunging forward if it is unloaded on a hill such that the torque measured by position-hold is no longer valid. NOTE: This parameter is applicable only when Torque Preload Enable = On (see parameter, previous page).
EM Brake Fault Motor Revs <i>EM_Brake_Fault_Motor_Revs</i> 0x3894 0x00	1.0 – 20.0 10 – 200	Defines the allowable number of motor revolutions after the EM brake is set before an EM Brake Failed to Set fault is issued (fault code 92).

NOTE: *This parameter is applicable only when Speed Mode or Speed Mode Express is selected and either Position Hold Enable = On or EM Brake Type = 2.

Quick Link:

Position Hold menu [p.39](#)

EM BRAKE CONTROL MENU: EM BRAKE TEST

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Mode <i>EM_Brake_Test_Mode</i> 0x30E1 0x00	0 – 1 0 – 1	Controls the test mode for the EM brake. 0 = VCL 1 = Startup
Direction <i>EM_Brake_Test_Direction</i> 0x30E2 0x00	0 – 2 0 – 2	Controls which direction torque is applied during the EM brake test. 0 = FWD 1 = REV 2 = Both
Current <i>EM_Brake_Test_Current</i> 0x30E3 0x00	10.0 – 100.0 % 3277 – 32767	The current as a percentage of base current applied during the EM brake test.
Ramp Time <i>EM_Brake_Test_Current_Ramp_Time</i> 0x30E4 0x00	10 – 1000 ms 10 – 1000	The time taken to ramp the current to the maximum EM_Brake_Test_Current value. The same time is used to ramp it down again.
Test Threshold <i>EM_Brake_Test_Threshold</i> 0x30E5 0x00	0.1 – 2.0 revs 10 – 200	The number of motor revolutions the motor can move during the EM brake test before a fault is declared.
Hold Time <i>EM_Brake_Test_Hold_Time</i> 0x30E6 0x00	0.1 – 1.0 seconds 12 – 125	The amount of time the torque is held between ramp up and ramp down times.
Max Speed Limit <i>Max_Speed_EM_Brake_Fault</i> 0x30E9 0x00	100 – 10000 RPM 100 – 10000	This parameter determines the max speed of the controller when an EM Brake Failed to Set fault is present.
Hill Compensation Enable <i>EM_Brake_Test_Hill_Compensation_Enable</i> 0x30E7 0x00	0 – 1 0 – 1	Enables or disables modifying the torque applied during the EM brake test while the vehicle is on a hill. If turned on, The Torque Preload Memory is added to the request to avoid overstressing the brake in 1 direction and not sufficiently testing it in the other.

DRIVERS: MAIN CONTACTOR MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Main Enable <i>Main_Enable</i> <i>OptionBits1 [Bit 0]</i> 0x306A 0x00	On / Off On / Off	When programmed On, the controller's native software controls the main contactor when the interlock is enabled; when programmed Off, the contactor is controlled by VCL. NOTE: With Main Enable programmed Off, 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.
Main Interlock Type <i>Main_Interlock_Type</i> 0x386F 0x00	0 – 1 0 – 1	When set to 0, the main contactor and interlock each work as determined by their respective parameters. When set to 1, the main contactor will pull in with KSI (like a Type 2 interlock) but the interlock input (pin 9, Switch 3) enables/disables drive and engages interlock braking (like a Type 0 or Type 1 interlock).
Pull In Voltage <i>Main_Pull_In_Voltage</i> 0x303C 0x00	0 – 100% 0 – 32767	The main contactor pull-in voltage 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 Voltage <i>Main_Holding_Voltage</i> 0x303D 0x00	0 – 100% 0 – 32767	The main contactor holding voltage parameter allows a reduced average voltage to be applied to the contactor coil once it has closed. This parameter must be set high enough to hold the contactor closed under all shock and vibration conditions the vehicle will be subjected to. NOTE: The Battery Voltage Compensated parameter (below) controls whether the pull-in and holding voltages are battery voltage compensated.

DRIVERS: MAIN CONTACTOR MENU, cont'd

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Battery Voltage Compensated <i>Main_Driver_Battery_Voltage_Compensated</i> <i>Main_Driver_Battery_Voltage_Compensated_Bit0</i> [Bit 0] 0x302C 0x00	On / Off On / Off	This parameter determines whether the main 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, page 68). 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 x Holding Voltage) to the coil. If Battery Voltage Compensated = Off, the output will be 30V (Actual Voltage x Holding Voltage) to the coil.
Interlock Type <i>Interlock_Type</i> 0x303E 0x00	0 – 2 0 – 2	Three interlock options are available: 0 = interlock turns on with switch 3. 1 = interlock controlled by VCL functions. 2 = interlock turns on with KSL.
Open Delay <i>Open_Delay</i> 0x303F 0x00	0.0 – 40.0 s 0 – 10000	Applicable only when Interlock Type = 0 or 1. The delay can be set to allow the 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.
Weld Check Enable <i>Weld_Check_Enable</i> <i>OptionBits1</i> [Bit 2] 0x306A 0x00	On / Off On / Off	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. This test is not performed if this parameter is Off. The main contactor <u>driver</u>, however, is always protected from short circuits.
Weld Check On Open Enable <i>Weld_Check_On_Open_Enable_Bit0</i> [Bit 0] 0x30E0 0x00	Off / On Off / On	When programmed On, the controller performs a test to make sure the main contactor is open (not welded shut) after it is commanded to open. This test is not performed if this parameter is Off.
Main DNC Check Enable <i>Main_DNC_Check_Enable</i> <i>OptionBits4</i> [Bit 7] 0x306D 0x00	On / Off On / Off	When programmed On, the controller performs a test immediately after the main contactor is commanded to close, to make sure the contactor has in fact closed. The test is not performed if this parameter is programmed Off. The main contactor <u>driver</u>, however, is always protected from short circuits.
Main DNC Runtime Threshold <i>Main_DNC_Runtime_Threshold</i> 0x3032 0x00	0.0 – 200.0 V 0 – 12800	Sets the threshold used for the ongoing check that ensures the main contactor remains closed while in operation. The Main DNC Runtime Threshold is the maximum voltage difference between the Keyswitch and Capacitor voltages. When the voltage difference is above this threshold, and the battery current is low, a Main Did Not Close fault will be set. Setting this parameter lower will increase the sensitivity of the fault detection. Setting this parameter too low may cause false fault trips due to normal voltage drops between the keyswitch and capacitor voltages. Setting this parameter = 0V will disable the Main Did Not Close fault check.
Precharge Enable <i>Precharge_Enable</i> <i>OptionBits2</i> [Bit 6] 0x306B 0x00	On / Off On / Off	Turns the precharge feature on and off. Precharge provides a limited current charge of the controller's internal capacitor bank before the main contactor is closed. This decreases the arcing that would otherwise occur when the contactor is closed with the capacitor bank discharged.

Quick Link:
 Battery menu [p.68](#)

DRIVERS: PROPORTIONAL DRIVER MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
PD Enable PD_Enable OptionBits1 [Bit 6] 0x306A 0x00	On / Off On / Off [PCF]	Determines how the PWM of the proportional driver is controlled. When programmed On, it is controlled by the controller's PD current control software. When programmed Off, it is controlled by the VCL function <i>Put_PWM</i> (PWM5, value); see Figure 17, page 118.
Hyd Lower Enable Hyd_Lower_Enable OptionBits1 [Bit 7] 0x306A 0x00	On / Off On / Off	When programmed On, lowering is controlled by throttle position. When programmed Off, lowering is controlled by the VCL variable <i>VCL_PD_Throttle</i> ; see Figure 17, page 118.
PD Max Current PD_Max_Current 0x3040 0x00	0.00 – 2.00 A 0 – 607	*The lift-lowering speed is determined by the aperture of the proportional valve. This parameter sets the maximum allowed current through the valve, which in turn defines its aperture.
PD Min Current PD_Min_Current 0x3041 0x00	0.00 – 2.00 A 0 – 607	*Sets the minimum allowed current through the proportional valve. Most proportional valves need a non-zero closed current in order to start opening immediately when Lower is requested.
PD Dither % PD_Dither_Percent 0x3042 0x00	0 – 100 % 0 – 32767	*Dither provides a constantly changing current in the coil to produce a rapid back-and-forth motion of the valve; this keeps the valve lubricated and allows low-friction, precise movement. The PD Dither % parameter specifies the amount of dither as a percentage of the PD max current, and is applied in a continuous cycle of add%-subtract%.
PD Dither Period PD_Dither_Period 0x3043 0x00	16 – 112 ms 1 – 7	*Sets the period for proportional valve dither.
PD Kp PD_Kp 0x3044 0x00	1 – 100 % 82 – 8192	*Sets the proportional gain of the current feedback controller. Higher gains force the control loop to respond quickly but may cause oscillations.
PD Ki PD_Ki 0x3045 0x00	1 – 100 % 327 – 32767	*Sets the integral gain of the current feedback controller. Integral gain tries to force the error to zero. Higher gains force the control loop to respond quickly but may cause oscillations.

*These parameter descriptions assume the proportional driver is being used to drive a proportional valve, and that the PD current control software is active (*PD_Enable* = On).

Quick Link:

[Fig 17 p.118](#)

DRIVERS: DRIVER 3 MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Contactor Enable Hydraulic_Contactor_Enable OptionBits4 [Bit 1] 0x306D 0x00	On / Off On / Off	When programmed On, VCL functions control Driver 3 as the hydraulic pump contactor. On: The VCL function <i>Start_Pump()</i> will close the pump contactor according to the defined pull-in and holding voltages. The VCL function <i>Stop_Pump()</i> will open the pump contactor. Off: Driver 3 will be available for general VCL usage. <i>Start_Pump()</i> and <i>Stop_Pump()</i> will not have any effect.
Pull In Voltage Driver3_Pull_In_Voltage 0x3864 0x00	0 – 100% 0 – 32767	The pull-in voltage parameter allows a high initial voltage when the contactor driver first turns on, to ensure contactor closure. After 1 second, this peak voltage drops to the contactor holding voltage. NOTE: This voltage will be battery voltage compensated.
Holding Voltage Driver3_Holding_Voltage 0x3865 0x00	0 – 100% 0 – 32767	The holding voltage parameter allows a reduced average voltage to be applied to the contactor coil once it has closed. This parameter must be set high enough to hold the contactor closed under all shock and vibration conditions the vehicle will be subjected to. NOTE: This voltage will be battery voltage compensated.

DRIVERS: FAULT CHECKING MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Driver1 Checks Enable <i>Driver1_Checks_Enable</i> <i>OptionBits2 [Bit 1]</i> 0x306B 0x00	On / Off <i>On / Off</i>	The five Checks Enable parameters are used to enable driver and coil fault detection at the five individual drivers (at Pins J1-6, J1-5, J1-4, J1-3, and J1-2). When a Driver Checks parameter is enabled, the associated driver, driver wiring, and driver load are checked to verify that the driver correctly drives the load both high and low. The checks will occur regardless of the PWM output of the driver. The checks will detect both open and shorted conditions. When a fault is detected, the controller opens the driver and issues a fault code. If nothing is connected to a driver, its Checks Enable parameter should be set Off. NOTE: Short circuit protection is always active at these five drivers, regardless of how Checks Enable is set.
Driver2 Checks Enable <i>Driver2_Checks_Enable</i> <i>OptionBits2 [Bit 2]</i> 0x306B 0x00	On / Off <i>On / Off</i>	
Driver3 Checks Enable <i>Driver3_Checks_Enable</i> <i>OptionBits2 [Bit 3]</i> 0x306B 0x00	On / Off <i>On / Off</i>	
Driver4 Checks Enable <i>Driver4_Checks_Enable</i> <i>OptionBits2 [Bit 4]</i> 0x306B 0x00	On / Off <i>On / Off</i>	
PD Checks Enable <i>PD_Checks_Enable</i> <i>OptionBits2 [Bit 5]</i> 0x306B 0x00	On / Off <i>On / Off</i>	
External Supply Max <i>External_Supply_Max</i> 0x3046 0x00	0 – 200 mA <i>0 – 800</i>	Sets the upper threshold of the combined current of the 5V and 12V external supplies. At or above this threshold a fault will be created that can be read by VCL.
External Supply Min <i>External_Supply_Min</i> 0x3047 0x00	0 – 200 mA <i>0 – 800</i>	Sets the lower threshold of the combined current of the 5V and 12V external supplies. At or below this threshold a fault will be created that can be read by VCL.

DRIVERS: PWM FREQUENCY PARAMETER

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
PWM Frequency <i>PWM_Frequency</i> 0x3938 0x00	100 – 1000 Hz <i>100 – 1000</i>	This single parameter defines the frequency of Drivers 1 through 4.

MOTOR MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Typical Max Speed <i>Typical_Max_Speed</i> 0x304D 0x00	500 – 10000 rpm <i>500 – 10000</i>	Set this parameter to the typical maximum motor speed of the vehicle. This value does not need to be set precisely; an estimate will do. All of the vehicle response rates are normalized to Typical Max Speed. For example, suppose Typical_Max_Speed is fixed at 6000 rpm, and Full_Accel_Rate_LS_SpdM = 3.0 seconds: If Max_Speed_SpdM = 6000 rpm, it will take 3.0 seconds to accelerate from zero to top speed (6000 rpm). If Max_Speed_SpdM = 3000 rpm, it will take 1.5 seconds to accelerate from zero to top speed (3000 rpm). If Max_Speed_SpdM = 1000 rpm, it will take 0.5 seconds to accelerate to accelerate from zero to top speed (1000 rpm). Note: This parameter appears in multiple places. Changing the value of this parameter affects all parameters listed below: <i>Program » Motor » 1 – PMAC (permanent Magnet Motor) » Commissioning Tests » Typical Max Speed.</i>
Motor Technology <i>Motor_Technology</i> 0x3814 0x00	0 – 1 <i>0 – 1</i> [PCF]	Set this parameter to the type of motor in the vehicle: 0 = ACIM (induction motor). 1 = PMAC (permanent magnet synchronous motor).* Note: A Parameter Mismatch Fault will result unless the Feedback Type matches the Motor Technology setting: If Motor Technology = 0 (ACIM), then the Feedback Type must = 1 (quadrature encoder). If Motor Technology = 1 (PMAC), then the Feedback Type must = 2 (sin/cos device).

*For PMAC applications, step through all the PMAC related parameters and considerations in these motor menus and NOTICE. See Chapters 7 and 8B for further PMAC related procedures.

PMAC Support and Considerations**NOTICE**

PMAC applications are only compatible with controllers that are manufactured with a cross-connect diode. These controllers are models 123XE-xx7x and 123XSE-xx7x. All the 1239E models are PMAC compatible when operated as described below. The cross-connect diode allows the logic section of the controller to power up if the capacitor voltage is above the brownout voltage regardless of whether the KSI pin is powered. Since a field weakened IPM motor can generate voltages significantly above the ratings of the motor, the controller must be able to control the main power bridge if the motor is generating significant voltages in order to protect itself from these damaging voltages. If a controller without a cross connect diode is configured for PMAC operation, a non-clearable Parameter Mismatch Fault will be declared.

Supported 1232E, 1232SE, 1234SE, 1236SE and 1238SE models

The following newly introduced 123X-xx7x models are manufactured with a cross-connect diode and calibrated for operation with OS34+:

- 1232E-2171, 1232E-2371, 1232E-5171, 1232E-5271, 1232E-6271, 1232E-7271.
- 1232SE-2471, 1232SE-5371, 1232SE-6371.
- 1234SE-4571, 1234SE-5471, 1234SE-6371.
- 1236SE-4671, 1236SE-5671, 1236SE-6571.
- 1238SE-4971, 1238SE-5971, 1238SE-6971, 1238SE-7971.

The 1234E, 1236E and 1238E controllers are not available with cross-connect diodes.

1239E PMAC Support (See manual 53095)

All existing 1239E models and the 1269E support PMAC operation with OS 37.0, but the OEM must insure that KSI power is active any time the motor is moving. Removal of KSI power when the motor is spinning past the base speed may result in immediate controller damage.

0 – INDUCTION MOTOR (ACIM): CHARACTERIZATION TESTS

See Chapter 8A for the complete description of the ACIM motor characterization procedure.

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Test Enable <i>IM_AutoChar_Test_Enable</i> 0x3800 0x00	0 – 1 0 – 1	Enables ACIM (induction motor) characterization with quadrature encoder to begin once Test Throttle is asserted.
Test Throtte <i>IM_Test_Throttle</i> 0x3034 0x00	-1 – 1 -1 – 1	Begins ACIM characterization if the Test Enable is asserted. This parameter is also used to change the direction of the motor to the forward direction during ACIM motor characterization. Important: If the motor starts to turn in the positive-rpm vehicle (forward) direction, set Test Throttle = 1 . If it turns in the negative-rpm vehicle (reverse) direction, set Test Throttle = -1 .
Motor Poles <i>IM_MotorData6</i> 0x3805 0x00	2 – 12 1 – 6	Use this parameter to select the number of induction motor <u>poles</u> before beginning the ACIM motor characterization procedure. Induction motor “poles” are always in “pairs” and are part of the motor’s specifications from the manufacturer. Use the number of “poles” ... not “pole pairs” for this parameter value. Note that the VCL variable, <i>IM_MotorData6</i> is in “pole pairs” (1/2 the 1314/1313 display value).
Max Test Speed <i>IM_AutoChar_Max_Test_Speed</i> 0x38A0 0x00	500 – 10000 rpm 500 – 10000	This parameter sets the maximum motor speed allowed during ACIM motor characterization.
Max Test Current <i>IM_AutoChar_Max_Test_Current</i> 0x38A1 0x00	10 – 100% 3277 – 32767	This parameter sets the maximum motor current allowed during ACIM motor characterization.
SlipGain <i>IM_MotorData8</i> 0x3819 0x00	0.00 – 200.00 0 – 20000	Adjust this parameter during ACIM motor characterization procedure to tune the motor to deliver the maximum torque per amp. Always tune this parameter.
Current Reg Tuning Test Enable <i>IM_CR_Tuning_Test_Enable</i> 0x388B 0x00	0 – 1 0 – 1	Performs only the current-regulator tuning portion of the autocharacterization test. To begin, clear any existing faults, then set the Interlock = On and this parameter = 1. Similar to the auto-characterization routine steps in Chapter 8A, then set Test Enable = 1 followed by setting Test Throttle = 1. The controller will run the tuning test and issue a Parameter Change Fault when finished. If a new (or existing Type 0) motor is auto-characterized following the steps in Chapter 8A, do not use this parameter, as it is automatically included in the full auto-characterization routine. Consult the Curtis distributor or support engineer for further assistance based on the motor and its application.

Quick Link:

Chapter 8A (ACIM motor characterization) [p.93](#)

MOTOR: FIELD WEAKENING CONTROL MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
FW Base Speed <i>FW_Base_Speed</i> 0x3857 0x00	200 – 12000 rpm <i>200 – 12000</i>	Sets the speed at which modulation depth has reached 100% (all available voltage is used) and where field weakening begins. This parameter is scaled by an internal motor characterization parameter, so the setting of FW Base Speed should always be done using the tuning test, which will take into account the scaling factor. Entering a speed observed from a torque vs. speed plot should not be done, as this number fails to take into account the internal scaling factor. This parameter needs to be set during initial setup (Chapter 7) and reset each time the Motor Type or the low speed current limit is changed. For example, if the Drive_Current_Limit or PL_Drive_Nominal parameters are lowered, reset this FW Base Speed parameter. To determine the correct value, perform this “base speed” tuning test. Run this test with batteries that have a reasonable charge. In either Speed Mode or Torque Mode, set the accel rates to be fast—so the motor will be accelerating at full current during the test. Next, set this FW Base Speed parameter to 6000 rpm (i.e., at a high-enough speed that its value will not interfere with the test result). From a stop, apply full throttle and accelerate to a high-speed beyond the expected base speed. After the motor settles at this speed for a moment, release the throttle to stop the test. Note the value displayed in <i>Monitor » Controller » Motor Tuning » Base Speed Captured</i>, and enter the value for this FW Base Speed parameter setting. For traction motors, the test restarts each time the vehicle comes to a stop and the throttle is released, so be sure to note the Base Speed Captured value before driving away. For pump motors, follow the instructions in Chapter 8A, Part 2b.
Field Weakening Drive <i>Field_Weakening_Drive</i> 0x3033 0x00	0 – 100% <i>0 – 1024</i>	Sets the amount of field weakening allowed while driving the motor. Field Weakening Drive will affect efficiency and torque only at speeds above the programmed FW Base Speed. Low settings will result in better efficiency but less torque; higher settings will result in more torque and less efficiency. A setting of zero will disable field weakening, thus resulting in the highest efficiency and lowest torque above base speed. To keep motor and controller heating to a minimum (high efficiency), Field Weakening Drive should be set just high enough to meet the high speed performance specifications. (Typical high speed performance specifications are maximum drive speed with full load, or maximum lift speed with full load.) The drive current limiting map can also be used to restrict performance at all speeds. The maximum setting of the Field Weakening Drive parameter depends on the type of motor characterization that was used. If the ACIM motor was dyno characterized (i.e., sent to the Curtis factory for characterization on the motor dyno), Field Weakening Drive can be set anywhere in the range of 0% (lowest torque, highest efficiency) to 100% (highest torque, lowest efficiency). If the ACIM motor was auto characterized (i.e., characterized using the ACIM motor characterization procedure in Chapter 8A), the range is restricted by the results of the Field Weakening Test. For these motors, Field Weakening Drive can be set anywhere in the range of 0% (lowest torque, highest efficiency) to Max Field Weakening Drive (highest torque, lowest efficiency) result. For auto characterized motors, setting Field Weakening Drive greater than the maximum setting found in the Field Weakening Test will result in poorer efficiency and less torque as the motor will be operating “over the slip curve.” NOTE: The Field Weakening Drive setting will have no effect at motor speeds below FW Base Speed.

Quick Links:

[Drive Current Limit p.44](#)[PL Drive Nominal p.45](#)[Base Speed Captured p.83](#)[Figure 12 p.45](#)

MOTOR: FIELD WEAKENING CONTROL MENU, cont'd

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Weakening Rate Drive <i>Field_Weakening_Rate_Drive</i> 0x383C 0x00	0 – 100% 0 – 500	Sets the control loop gains for field weakening. Setting the rate too low may create surging in the vehicle as it accelerates at mid to high speeds. Setting the rate too high may create high frequency oscillations (usually audible) when the vehicle accelerates at mid to high speeds.
Min Field Current <i>Min_Field_Current</i> 0x3821 0x00	0 – 800 A 0 – 8000	Sets the field current when no torque is requested from the motor. This current pre-fluxes the motor and can improve initial take-off at the expense of some battery consumption.
Swap Two Phases <i>Swap_Two_Phases_OptionBits3</i> [Bit 3] 0x306C 0x00	On / Off On / Off [PCF]	If, after Swap Encoder Direction has been set correctly, the vehicle drives in the wrong direction (i.e., drives forward when in reverse, and vice versa), try changing the setting of the Swap Two Phases parameter. This parameter has the same effect as physically swapping the cables on any two of the three motor phase connections. Positive motor speed must be in the forward direction in order for the emergency reverse feature to operate properly. NOTE: Do not change this parameter while the controller is powering the motor. Any time this parameter is changed a Parameter Change Fault (fault code 49) is set and must be cleared by cycling power; this protects the controller and the operator. Adjusting this parameter can be hazardous. For instructions, see Chapter 7, Step 12.
Swap Feedback Direction <i>Swap_Feedback_Direction</i> <i>OptionBits3 [Bit 0]</i> 0x306C 0x00	On / Off On / Off [PCF]	Changes the motor encoder's effective direction of rotation. The encoder provides data used to calculate motor position and speed. This parameter must be set such that when the motor is turning forward, the controller reports back a positive motor speed. Positive motor speed must be in the forward direction in order for the emergency reverse feature to operate properly. Note: Do not change this parameter while the controller is powering the motor. Any time this parameter is changed a Parameter Change Fault (fault code 49) is set and must be cleared by cycling power; this protects the controller and the operator. Adjusting this parameter can be hazardous. For instructions, see Chapter 7, Step 10.
Motor Type <i>IM_Motor_Type</i> 0x3809 0x00	0 – 536* 0 – 536 [PCF]	This parameter references a predefined table of motor parameters for many AC induction motors. Motor Type 0 is reserved for using the auto-characterization procedure in Chapter 8A. NOTE: Changing this parameter requires the re-setting of the FW Base Speed parameter. Other parameters may also need to be reset or evaluated by following the Initial Setup and Tuning Guide chapters instructions. Consult the Curtis distributor or support engineer for information how to set this parameter based on the application and motor.

*536 = os37.49

528 = os37.34

504 = os37.0

MOTOR: LOS (Limited Operating Strategy) MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
LOS Upon Encoder Fault <i>LOS_Upon_Encoder_Fault</i> <i>LOS_Upon_Encoder_Fault_Bit0 [Bit 0]</i> 0x308F 0x00	On / Off On / Off	Limited Operating Strategy (LOS) is typically used to drive the vehicle back to a repair center at very low speeds in the event the motor encoder fails. Following an encoder failure ((either Encoder Fault (fault code 36) or Stall Detected (fault code 73)), and after the Interlock is cycled, the vehicle enters LOS mode thus allowing drive without motor encoder feedback. In LOS mode, the ability to achieve maximum torque—even for a very short time—is considered more important than smoothness. When LOS mode is entered, the Encoder LOS fault (fault code 93) becomes active and the encoder fault ((either Encoder Fault (fault code 36) or Stall Detected (fault code 73)) is cleared. When this parameter is programmed On, LOS mode will be entered in the event of an encoder fault followed by an Interlock cycle. When programmed Off, in the event of an encoder fault the encoder fault remains and drive is disabled.
LOS Max Speed <i>Enc_LOS_Max_Speed</i> 0x3083 0x00	100 – 2000 rpm 100 – 2000	This parameter indirectly defines the maximum speed for LOS mode by setting the maximum frequency that corresponds to LOS Max Speed. In LOS mode the throttle commands a frequency that is interpolated linearly between zero (at Throttle Command = 0%) and the programmed LOS Max Speed (at Throttle Command = 100%).
LOS Max Current <i>Enc_LOS_Max_Current</i> 0x3085 0x00	50 – 1000 A 500 – 10000	In LOS mode, a partial or full throttle command will result in the maximum current set by this parameter. This current setting is clamped by the controller's rated current.
LOS Max Mod Depth <i>Enc_LOS_Max_Mod_Depth</i> 0x3084 0x00	15 – 100% 177 – 1182	In LOS mode, the maximum modulation depth acts to limit the current at higher speeds. This parameter should be set such that the modulation depth limit is reached prior to the LOS Max Speed limit, so that the motor current will fall off from LOS Max Current at higher speeds. This may allow the vehicle to drive longer in LOS mode, as it lessens the chance of the motor or controller overheating.
LOS Accel Rate <i>Enc_LOS_Accel_Rate</i> 0x3086 0x00	2.0 – 15.0 s 2000 – 15000	Defines the rate (in seconds) at which the frequency increases when full throttle is applied, while operating in LOS mode. This parameter should be set to a slow rate (high parameter value) so the frequency command has a very slow slew rate to ensure that the max torque point is hit for a reasonable period of time; this decreases the probability of going over the slip curve, and allows ramps or obstacles to be overcome.
LOS Decel Rate <i>Enc_LOS_Decel_Rate</i> 0x3087 0x00	2.0 – 15.0 s 2000 – 15000	Defines the rate (in seconds) at which the frequency decreases when throttle is released, while operating in LOS mode.

MOTOR: 1 – PMAC (PERMANENT MAGNET MOTOR): COMMISSIONING TESTS MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Typical Max Speed <i>Typical_Max_Speed</i> 0x304D 0x00	500 – 10000 RPM <i>500 – 10000</i>	Set this parameter to the typical maximum motor speed of the vehicle. This value does not need to be set precisely; an estimate will do. All of the vehicle response rates are normalized to Typical Max Speed. For example, suppose Typical_Max_Speed is fixed at 6000 rpm, and Full_Accel_Rate_LS_SpdM = 3.0 seconds: If Max_Speed_SpdM = 6000 rpm, it will take 3.0 seconds to accelerate from zero to top speed (6000 rpm). If Max_Speed_SpdM = 3000 rpm, it will take 1.5 seconds to accelerate from zero to top speed (3000 rpm). If Max_Speed_SpdM = 1000 rpm, it will take 0.5 seconds to accelerate to accelerate from zero to top speed (1000 rpm).
Max Test Speed <i>PMAC_Commissioning_Max_Test_Speed</i> 0x38C1 0x00	1000 – 10000 RPM <i>1000 – 10000</i>	This parameter is used to set the maximum motor speed allowed during PMAC motor commissioning.
Motor Type <i>PMAC_Motor_Type</i> 0x3854 0x00	1 – 127* <i>1 – 127</i> [PCF]	This parameter references a predefined table of motor parameters for many PMAC motors. This parameter appears in multiple places. Changing the value of this parameter affects the parameter in: <i>Program » Motor » 1-PMAC (Permanent Magnet Motor) » Motor Type.</i>
Test Enable <i>PMAC_Commissioning_Test_Enable</i> 0x3862 0x00	0 – 1 <i>0 – 1</i>	This parameter is used to start the PMAC motor commissioning procedure. If the motor type is known, enter it first. Manually close the Interlock switch after this has been set, then set the Test Throttle to start the test.
Test Throttle <i>PMAC_Test_Throttle</i> 0x30D1 0x00	-1 – 1 <i>-1 – 1</i>	Begins commissioning if a Test Enable asserted. IMPORTANT: If motor starts to turn in positive (forward) vehicle speed direction, set the Test Throttle to 1 . If the motor turns in the negative (reverse) direction, set the Test Throttle to -1 .
Swap Direction <i>OptionBits3 [Bit 0]</i> 0x306C 0x00	On/Off <i>On/Off</i> [PCF]	Changes the physical direction of rotation of a PMAC motor relative to the throttle command. Change the state of this parameter if the motor turns in the incorrect direction with a positive throttle after the commissioning procedure is complete. Positive motor speed must be in the forward direction in order for the emergency reverse feature to operate properly. Note: Do not change this parameter while the controller is powering the motor. Any time this parameter is changed a Parameter Change Fault (fault code 49) is set and must be cleared by cycling power; this protects the controller and the operator. Adjusting this parameter can be hazardous. If in doubt, consult the revised PMAC commissioning procedure Chapter 8B in this document, reference step 14.
Max Test Current <i>PMAC_Commissioning_Max_Test_Current</i> 0x38C2 0x00	2 – 30% <i>655 – 9831</i>	This parameter is used to set the maximum motor current allowed during PMAC motor commissioning.

*127 = os37.49

114 = os37.34

89 = os37.0

MOTOR: 1 – PMAC (PERMANENT MAGNET MOTOR): FIELD WEAKENING CONTROL MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Field Weakening Rate <i>PMAC_Field_Weakening_Rate</i> 0x3875 0x00	0 – 100 % 0 – 100	Sets the control loop gains for field weakening. Setting the rate too low may create surging in the vehicle as it accelerates at mid to high speeds. Setting the rate too high may create high frequency oscillations (usually audible) when the vehicle accelerates at mid to high speeds.
FW Current Limit (Apk) <i>PMAC_Field_Weakening_Min_Id</i> 0x355C 0x00	–3200 – 0 A –32000 – 0	Sets the minimum negative Id Current allowed in field weakening.

MOTOR: 1 – PMAC (PERMANENT MAGNET MOTOR) MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Motor Type <i>PMAC_Motor_Type</i> 0x3854 0x00	1 – 127* 1 – 127 [PCF]	This parameter references a predefined table of motor parameters for many PMAC motors. This parameter appears in multiple places. Changing the value of this parameter affects the parameter in: <i>Program » Motor » 1-PMAC (Permanent Magnet Motor) » Commissioning Tests » Motor Type.</i>
Restricted Operation <i>PMAC_Restricted_Operation</i> 0x351A 0x00	0 – 1 0 – 1	Limits motor to a safe range and stops regen to allow the controller to never short the motor. This allows the controller to operate with a current rating less than the motor short circuit current.

* 127 = os37.49

114 = os37.34

89 = os37.0

MOTOR: 1 – PMAC (PERMANENT MAGNET MOTOR): COMMISSIONING RESULTS MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Tuning Voltage <i>PMAC_Tuning_Nominal_Pack_Voltage</i> 0x3856 0x00	22.0 – 200.0V 1408 – 12800	This value is set by the commissioning process and indicates the capacitor voltage at the time the commissioning process was run.
Short Circuit Current (RMS) <i>PMAC_Short_Circuit_Current</i> 0x351B 0x00	0 – 1000 A 0 – 10000	This value should be measured on the motor by shorting the three phases together, external spinning up to high speed, and then measuring any of the three phase currents.
Phasing Mode <i>PMAC_Phasing_Mode_Bit0 [Bit 0]</i> 0x3874 0x00	On/Off On/Off	Set by the commissioning process— do not change manually .
Sensor Position Offset <i>PMAC_Encoder_Position_Offset</i> 0x351C 0x00	0 – 4095 0 – 4095	The electrical offset to the position sensor for magnet alignment in PMAC applications (4096 = 1 Mechanical Rev).
Sensor Compensation Factor <i>PMAC_Sensor_Compensation_Factor</i> 0x351D 0x00	0 – 4095 0 – 4095	Compensation of the delay in motor PWM and Sin/Cos sensor filtering.
Kp d Axis Current <i>PMAC_Kp_d_Axis_Current</i> 0x3859 0x00	50 – 32767 50 – 32767	Determines how aggressively the d axis current regulator attempts to match the current command. Larger values provide tighter control but may cause oscillation. This parameter is set automatically by the commissioning process and should not be adjusted manually.
Ki d Axis Current <i>PMAC_Ki_d_Axis_Current</i> 0x385A 0x00	0 – 5000 0 – 5000	The integral term (Ki) forces zero steady state error, so the d axis current regulator will run at exactly the commanded current. Larger values provide tighter control but may cause oscillation. This parameter is set automatically by the commissioning process and should not be adjusted manually.
Kp q Axis Current <i>PMAC_Kp_q_Axis_Current</i> 0x351E 0x00	50 – 32767 50 – 32767	Determines how aggressively the q axis current regulator attempts to match the current command. Larger values provide tighter control but may cause oscillation. This parameter is set automatically by the commissioning process and should not be adjusted manually.
Ki q Axis Current <i>PMAC_Ki_q_Axis_Current</i> 0x351F 0x00	0 – 5000 0 – 5000	The integral term (Ki) forces zero steady state error, so the q axis current regulator will run at exactly the commanded current. Larger values provide tighter control but may cause oscillation. This parameter is set automatically by the commissioning process and should not be adjusted manually.
Sin Min <i>Feedback_Sin_Min</i> 0x3863 0x00	0 – 1023 0 – 1023	Minimum output by the Sin/Cos sensor on the Sin channel. Input detected at Position Feedback A (pin 31). The value is set during the PMAC motor commissioning procedure.
Sin Max <i>Feedback_Sin_Max</i> 0x385C 0x00	0 – 1023 0 – 1023	Maximum output by the Sin/Cos sensor on the Sin channel. Input detected at Position Feedback A (pin 31). The value is set during the PMAC motor commissioning procedure.
Cos Min <i>Feedback_Cos_Min</i> 0x385F 0x00	0 – 1023 0 – 1023	Minimum output by the Sin/Cos sensor on the Cos channel. Input detected at Position Feedback B (pin 32). The value is set during the PMAC motor commissioning procedure.
Cos Max <i>Feedback_Cos_Max</i> 0x382F 0x00	0 – 1023 0 – 1023	Maximum output by the Sin/Cos sensor on the Cos channel. Input detected at Position Feedback B (pin 32). The value is set during the PMAC motor commissioning procedure.
Rsys <i>PMAC_Rsys</i> 0x3530 0x00	0.0 – 100.0 mOhm 0 – 1000	This is the measured stator resistance of the motor including the wiring between the motor and controller. The value is set during the PMAC motor commissioning procedure.
Commissioning Needed <i>PMAC_Commissioning_Needed</i> 0x3515 0x00	0 – 1 0 – 1	This parameter is set on defaults refresh, including a new OS download. Running a successful autocharacterization clears it, or it can be cleared manually if the user is certain the rest of the parameters are setup correctly.

MOTOR: FEEDBACK OPTIONS MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Feedback Type <i>Feedback_Type</i> 0x382E 0x00	1 – 2 1 – 2 [PCF]	Set this parameter to the type of position feedback device in the vehicle: 1 = quadrature encoder. 2 = Sin/Cos sensor. Note: A Parameter Mismatch Fault will result unless the Feedback Type matches the Motor Technology setting: If Motor Technology = 0 (ACIM), then the Feedback Type must = 1 (quadrature encoder). If Motor Technology = 1 (PMAC), then the Feedback Type must = 2 (Sin/Cos sensor).

MOTOR: FEEDBACK OPTIONS – 1 – QUADRATURE ENCODER MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Encoder Steps <i>Encoder_Steps</i> 0x304B 0x00	32 – 256 32 – 256 [PCF]	Sets the number of encoder pulses per revolution. This must be set to match the encoder; see motor nameplate or its technical data for this information. NOTE: Do not change this parameter while the controller is powering the motor. Any time this parameter is changed a Parameter Change Fault (fault code 49) is set and must be cleared by cycling power; this protects the controller and the operator. Adjusting this parameter can be hazardous; setting it improperly may cause vehicle malfunction, including uncommanded drive. For instructions, see Chapter 7, Step 8.

**WARNING**

MOTOR: FEEDBACK OPTIONS – 1 - QUADRATURE ENCODER: SPEED FILTERS MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Speed Filter <i>Speed_Filter</i> 0x3869 0x00	5.0 – 250.0 660 – 32767	Units are in rad/sec. Divide by 2*pi to get equivalent value in Hz.
Speed Filter Zero Speed <i>Speed_Filter_Zero_Speed</i> 0x3885 0x00	0.5 – 250.0 64 – 32767	Units are in rad/sec. Divide by 2*pi to get equivalent value in Hz.

MOTOR: FEEDBACK OPTIONS – 2 – SIN/COS MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Enable Multiturn Sensor <i>Feedback_Multiturn</i> 0x3531 0x00	0 – 1 0 – 1 [PCF]	Enable Multiturn Sensor = 0 (Off): One sensor cycle per mechanical revolution. (Not using a multiturn encoder). Enable Multiturn Sensor = 1 (On): One sensor cycle per electrical cycle. (Using a multiturn encoder).
Sin Cos Fault Threshold <i>Sin_Cos_Fault_Threshold</i> 0x3878 0x00	5.0 – 20.0 % 51 – 205	Defines the % allowable change in the SIN or COS feedback. A SinCos Fault is generated if this threshold is exceeded 10 times sequentially.
Sin Cos fault Threshold High <i>Sin_Cos_Fault_Threshold_High</i> 0x38CD 0x00	20.0 – 100.0 % 205 – 1024	Defines the % allowable change in the SIN or COS feedback (high value used for first 30 seconds after interlock. This will allow the controller to adjust for slight variances if the sensor was changed.
Sin Cos Fault Time <i>Sin_Cos_Fault_Time</i> 0x38CE 0x00	10 – 1000 ms 10 – 1000	A sin cos fault will be generated if the sensor is outside the tolerance for 10ms during this amount of time.
Sin Cos Startup Time <i>Sin_Cos_Startup_Time</i> 0x3532 0x00	0 – 2000 ms 0 – 2000	Time at startup for the sin/cos sensor to start producing valid data. No sin/cos faults will be declared during this time but the power section will not enable.
Sin Min <i>Feedback_Sin_Min</i> 0x3863 0x00	0 – 1023 0 – 1023	Minimum output by the Sin/Cos sensor on the Sin channel. Input detected at Position Feedback A (pin 31). The value is set during the PMAC motor commissioning procedure.
Sin Max <i>Feedback_Sin_Max</i> 0x385C 0x00	0 – 1023 0 – 1023	Maximum output by the Sin/Cos sensor on the Sin channel. Input detected at Position Feedback A (pin 31). The value is set during the PMAC motor commissioning procedure.
Cos Min <i>Feedback_Cos_Min</i> 0x385F 0x00	0 – 1023 0 – 1023	Minimum output by the Sin/Cos sensor on the Cos channel. Input detected at Position Feedback B (pin 32). The value is set during the PMAC motor commissioning procedure.
Cos Max <i>Feedback_Cos_Max</i> 0x382F 0x00	0 – 1023 0 – 1023	Maximum output by the Sin/Cos sensor on the Cos channel. Input detected at Position Feedback B (pin 32). The value is set during the PMAC motor commissioning procedure.

MOTOR: FEEDBACK OPTIONS – 2 – SIN/COS: SPEED FILTERS MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Speed Filter <i>Speed_Filter_Sin_Cos</i> 0x3534 0x00	5.0 – 250.0 660 – 32767	Units are in rad/sec. Divide by 2*pi to get equivalent value in Hz.
Speed Filter Zero Speed <i>Speed_Filter_Zero_Speed_Sin_Cos</i> 0x3535 0x00	0.5 – 250.0 64 – 32767	Units are in rad/sec. Divide by 2*pi to get equivalent value in Hz.

MOTOR: FEEDBACK OPTIONS – FEEDBACK FAULT SETUP

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Fault Detection Enable <i> Encoder_Fault_Detection_Enable</i> <i> Encoder_Fault_Detection_Enable_Bit0 [Bit 0]</i> 0x389C 0x00	On/Off On/Off	When programmed On, encoder fault checking is enabled. Three fault conditions are checked: Encoder Fault (fault code 36, Stall Detected (fault code 73), and Encoder Pulse Error (fault code 88).
Fault Max Speed <i>Encoder_Fault_Max_Speed</i> No CAN Object	0 – 2000 RPM 0 – 2000	If the controller transitions from above this speed to below minimum speed in 8ms an encoder fault will be declared.
Fault Min Speed <i>Encoder_Fault_Min_Speed</i> 0x308B 0x00	0 – 2000 RPM 0 – 2000	Speed determined to be 0 speed when using a sin/cos feedback sensor. Has no effect using a quadrature encoder, that value is calculated based on the speed for a single pulse.
Pulse Fault Detect Time <i>Encoder_Pulse_Fault_Detect_Time</i> 0x393E 0x00	0.0 – 3.0 S 0 – 94	Defines the minimum time it takes for the controller, while the vehicle is in motion, to detect that the encoder and the Encoder Steps parameter do not match. When the Encoder Steps setup is incorrect, the motor controller cannot properly calculate AC motor field orientation. The loss of field orientation can cause the motor to spin up toward full speed once any throttle is applied. This parameter sets a timer that starts once the throttle has been moved and then released, the drive current is significant, and the motor is still accelerating. Note that the motor can spin at high rpm for several seconds before conditions allow the controller to properly detect a fault. An Encoder Pulse Error (fault code 88) is declared when this fault is detected. Setting the parameter to zero will disable this fault detection.
Fault Stall Speed <i>Enc_Fault_Stall_Speed</i> No CAN Object	0 – 100 RPM 0 – 100	Sets the threshold below which the stall timer may count. If Motor Speed is less than Fault Stall Speed for Fault Stall Time, a Stall Detected fault (fault code 73) may be issued.
Fault Stall Time <i>Enc_Fault_Stall_Time</i> 0x3089 0x00	0 – 10 s 0 – 5000	Sets a timer when no motor encoder movement is detected. In Speed Modes: If no motor encoder movement is detected for the programmed Fault Stall Time, a Stall Detected fault (fault code 73) is issued. In Torque Mode: If no motor encoder movement is detected for the programmed Fault Stall Time, with maximum throttle applied, a Stall Detected fault (fault code 73) is issued.

MOTOR: TEMPERATURE CONTROL MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Sensor Enable <i>MotorTemp_Sensor_Enable</i> <i>OptionBits3 [Bit 1]</i> 0x306C 0x00	On / Off On / Off	When programmed On, the motor temperature cutback and the motor temperature compensation features are enabled. This parameter can be used only if a temperature sensor has been properly configured. The motor temperature cutback feature will linearly cut back the current from 100% to 0% between the Temperature Hot and Temperature Max temperatures. The motor temperature compensation feature will adapt the motor control algorithms to varying motor temperatures, for improved efficiency and more consistent performance.
Sensor Type <i>MotorTemp_Sensor_Type</i> 0x307B 0x00	0 – 9 0 – 9	Nine sensor types are predefined in the software: - Type 1 KTY83–122. - Type 2 2× Type 1, in series. - Type 3 KTY84–130 or KTY84–150. - Type 4 2× Type 3, in series. - Type 5 PT1000. - Type 6 SDN16_FKFN_X. - Type 7 KTY84_150. - Type 8 Omega 44004. - Type 9 KTY81_110. Custom sensor types can be set up using VCL, if the predefined types are not appropriate for the application. Please contact the Curtis distributor or support engineer. NOTE: The industry standard KTY temperature sensors are silicon temperature sensors with a polarity band; the polarity band of a KTY sensor must be the end connected to I/O Ground (pin 7).
Sensor Offset <i>MotorTemp_Sensor_Offset</i> 0x305A 0x00	–20 – 20°C –200 – 200	Often the sensor is placed in the motor at a location with a known offset to the critical temperature; the offset can be corrected with this parameter. The parameter can also be used to correct a known offset in the sensor itself.
Braking Thermal Cutback Enable <i>MotorBrakingThermalCutback_Enable</i> <i>MotorBrakingThermalCutback_Enable_Bit0 [Bit0]</i> 0x386C 0x00	On / Off On / Off	When programmed On, drive current and regen braking current will be cut back based on motor temperature. All forms of regen braking current (Regen_Current_Limit, Brake_Current_Limit, EMR_Current_Limit, Interlock_Brake_Current_Limit) includes emergency reverse braking, interlock braking, brake input braking, direction reversal braking, neutral braking, and speed limit braking. When programmed Off, only drive current (and not regen braking current) will be cut back based on motor temperature. If the vehicle has mechanical brakes, setting this parameter to On may help reduce motor heating.
Temperature Hot <i>MotorTemp_Hot</i> 0x304E 0x00	0 – 250°C 0 – 2500	Defines the temperature at which current cutback begins.
Temperature Max <i>MotorTemp_Max</i> 0x304F 0x00	0 – 250°C 0 – 2500	Defines the temperature at which current is cut back to zero.
MotorTemp LOS Max Speed <i>MotorTemp_LOS_Max_Speed</i> 0x308A 0x00	100 – 3000 rpm 100 – 3000	When a Motor Temp Sensor Fault (fault code 29) is set, a LOS (Limited Operating Strategy) mode is engaged. The maximum speed is reduced to the programmed Max Speed in the operating mode (Max_Speed_SpdMx, Max_Speed_SpdM, Max_Speed_TrqM) or to the MotorTemp_LOS_Max_Speed, whichever is lower.

NOTICE

BATTERY MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Nominal Voltage <i>Nominal_Voltage</i> 0x3048 0x00	24 – 96V 1536 – 6144	Must be set to the vehicle's nominal battery pack voltage. This parameter is used in determining the overvoltage and undervoltage protection thresholds for the electronic system. See the Voltage Limits section for Over, Under, and Brownout voltage definitions and constraints. NOTE: In all cases the OS default Nominal Voltage is the highest value in the controller's range (See Table D-1). Always check and adjust this parameter whenever using a lower nominal voltage. Scaling is 64 counts per volt. For example, controller 1238E-65XX has a 48–80 voltage range. Therefore, when using a 48 volt battery, the <i>Nominal_Voltage</i> parameter will be 3072 ($64 \times 48 = 3072$) in the application's VCL program, or as the raw (decimal) value in TACT.
Kp UV <i>Batt_Kp_UV</i> 0x380E 0x00	0.0 – 100.0% / V 0 – 1024	When the battery voltage goes below the undervoltage threshold, a closed loop PI (Proportional/Integral) controller is enabled in an attempt to keep the battery voltage from drooping. It accomplishes this by cutting back the drive current, thereby reducing the load on the battery.
Ki UV <i>Batt_Ki_UV</i> 0x380F 0x00	0 – 100% 0 – 16384	The Kp term is the proportional gain and is set in units of % cutback per volt; for example, a setting of 25 would provide full current cutback with 4 V of droop. The Ki term is the integral gain. Integral gain will accumulate the voltage droop and attempt to bring the battery droop back to 0 V. Higher gains will react more strongly and quickly. Typically, Kp UV and Ki UV are used together to provide the best response. If the linear response of the previous AC controllers is preferred, set Ki UV = 0.
User Overvoltage <i>User_Overvoltage</i> 0x3049 0x00	105 – 200% 269 – 512	The value of this parameter is a percentage of the Nominal Voltage setting. The User Overvoltage parameter can be used to adjust the overvoltage threshold, which is the voltage at which the controller will cut back regen braking to prevent damage to the electrical system. Typically this parameter is changed only when the controller is being used in an application at the low end of the controller's range: such as a 48–80V controller being used in a system with a 48 V battery pack. In this case, the overvoltage threshold can be raised by setting the User Overvoltage to a higher value. The overvoltage threshold can never be raised above the controller's power base maximum voltage rating.
User Undervoltage <i>User_Undervoltage</i> 0x304A 0x00	0 – 95% 0 – 242	The value of this parameter is a percentage of the Nominal Voltage setting. The User Undervoltage parameter can be used to adjust the undervoltage threshold, which is the voltage at which the controller will cut back drive current to prevent damage to the electrical system. Typically this parameter is changed only when the controller is being used in an application at the high end of the controller's range: such as a 24–36V controller being used in a system with a 36V battery pack. In this case, the undervoltage threshold can be lowered by setting the User Undervoltage to a lower value. The undervoltage threshold can never be lowered below the controller's power base minimum voltage rating.
Reset Volts Per Cell <i>BDI_Reset_Volts_Per_Cell</i> 0x3170 0x00	0.900 – 3.000V 900 – 3000	The reset voltage level is checked only once, when the main contactor first closes. Note that the BDI Reset Percent parameter also influences the algorithm that determines whether BDI Percentage is reset to 100%. Reset Volts Per Cell should always be set higher than Full Volts Per Cell. <u>Reset Voltage Level</u> = Reset Volts Per Cell × number of cells in the battery pack.*

* To determine the number of cells in the battery pack, divide the Nominal Voltage setting by the battery chemistry's nominal volts-per-cell.
 Lead-acid: 2.0 V/cell, nominal.

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

Quick Link:
[Voltage Limits p.25](#)

BATTERY MENU, cont'd

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Full Volts Per Cell <i>BDI_Full_Volts_Per_Cell</i> 0x3171 0x00	0.900 – 3.000V 900 – 3000	The full voltage level sets the B+ (capacitor voltage) that is considered to be 100% state-of-charge; when a loaded battery drops below this voltage, it begins to lose charge. Capacitor Voltage is viewable in the programmer's Monitor » Battery menu. $\text{Full Voltage Level} = \text{Full Volts Per Cell} \times \text{number of cells in the battery pack.}^*$
Empty Volts Per Cell <i>BDI_Empty_Volts_Per_Cell</i> 0x3172 0x00	0.900 – 3.000V 900 – 3000	The empty voltage level sets the Capacitor Voltage that is considered to be 0% state-of-charge. $\text{Empty Voltage Level} = \text{Empty Volts Per Cell} \times \text{number of cells in the battery pack.}^*$
Discharge Time <i>BDI_Discharge_Time</i> 0x3173 0x00	0 – 600 minutes 0 – 600	Sets the minimum time for the BDI algorithm to count down the BDI Percentage from 100% to 0%. The BDI algorithm integrates the time the filtered capacitor voltage is below the state of charge voltage level. When that cumulative time exceeds the Discharge Time / 100, the BDI Percentage is decremented by one percentage point and a new state of charge voltage level is calculated. $\text{State of Charge Level} = [(\text{Full Voltage Level} - \text{Empty Voltage Level}) \times \text{BDI Percentage} / 100] + \text{Empty Voltage Level.}$
BDI Reset Percent <i>BDI_Reset_Percent</i> 0x3174 0x00	0 – 100% 0 – 100	When a battery has a high BDI percentage, its float voltage at KSI On can sometimes cause false resets. The BDI Reset Percent parameter addresses this problem by allowing the user to define a BDI Percentage value above which the BDI Percentage variable will not reset. When KSI is first powered on, the BDI Percentage variable will reset to 100% only if ((Keyswitch Voltage > Reset Voltage Level) and (BDI Percentage < BDI Reset Percent)).

* To determine the number of cells in the battery pack, divide the Nominal Voltage setting by the battery chemistry's nominal volts-per-cell.
 Lead-acid: 2.0 V/cell, nominal.

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

BATTERY MENU: BATTERY CURRENT LIMITING

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Battery Current Limiter Enable <i>Battery_Current_Limiter_Enable</i> 0x31E1 0x00	0 – 1 0 – 1	Setting this parameter to 1 enables drive and regen battery current limits. This is particularly useful when interfacing to Lithium Batteries with defined current limits.
Drive Battery Current Limit <i>Drive_Battery_Current_Limit</i> 0x31E2 0x00	0 – 1000 A 0 – 10000	Sets a limit on the DC current that the controller may draw out of the battery.
Regen Battery Current Limit <i>Regen_Battery_Current_Limit</i> 0x31E3 0x00	0 – 1000 A 0 – 10000	Sets a limit on the DC current that the controller may put into the battery from regenerative braking.
Kp <i>Battery_Current_Kp</i> 0x31E4 0x00	0 – 100.0 % 0 – 1000	Determines how aggressively the battery current limiter attempts to match the Drive Battery Current Limit or the Regen Battery Current Limit. This parameter is active when the motor is spinning at low motor speeds. Larger values provide tighter control but may cause oscillation.
Kp HS <i>Battery_Current_Kp_HS</i> 0x31E5 0x00	0 – 100.0 % 0 – 1000	Sets the proportional gain (Kp) for high motor speeds; see description of Kp.

BATTERY MENU: BATTERY CURRENT LIMITING, cont'd

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Ki <i>Battery_Current_Ki</i> 0x31E6 0x00	0 – 100.0 % 0 – 1000	The integral gain (Ki) forces zero steady state error, so the battery current regulator will limit to exactly the current limit set by the Drive Battery Current Limit or Regen Battery Current Limit. This parameter is active when the motor is spinning at low speeds. Larger values provide tighter control but may cause oscillation.
Ki HS <i>Battery_Current_Ki_HS</i> 0x31E7 0x00	0 – 100.0 % 0 – 1000	Sets the integral gain (Ki) for high motor speeds; see description of Ki.
Kd <i>Battery_Current_Kd</i> 0x31E8 0x00	0 – 100.0 % 0 – 1000	The derivative gain (Kd) provides damping as the battery current approaches the programmed limit, thereby reducing overshoot. This parameter is active when the motor is spinning at lower speeds. If Kd is set too high, the current may cut back before the limit and may take too long to reach the limit. If Kd is set too low, the current may overshoot the limit.
Kd HS <i>Battery_Current_Kd_HS</i> 0x31E9 0x00	0 – 400.0 % 0 – 4000	Sets the derivative gain (Kd) for high motor speeds; see description of Kd.
Limit Ramp Rate <i>Battery_Current_Limit_Ramp_Rate</i> 0x31EA 0x00	0 – 30.0 Seconds 0 – 30000	This parameter limits how fast the actual Battery Current Limit can change when the Drive or Regen battery current limit parameters are changed during controller operation. A setting of 30.0 seconds means that the limit will take 30.0 seconds to ramp from the full current limit of the controller to zero or vice versa. A setting of zero means that the ramp rate will not be limited.

BATTERY MENU: BATTERY POWER LIMITING

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Drive Power Limit <i>Battery_Drive_Power_Limit</i> 0x31EB 0x00	0.00 – 300.00 kW 0 – 30000	Sets a limit on the DC power that the controller will draw out of the battery during drive. Battery Current Limiting must be enabled for Battery Power Limiting to function. The controller will use more restrictive of this or battery current limit.
Regen Power Limit <i>Battery_Regen_Power_Limit</i> 0x31EC 0x00	0.00 – 300.00 kW 0 – 30000	Sets a limit on the DC power that the controller will put into the battery during regen. Battery Current Limiting must be enabled for Battery Power Limiting to function. The controller will use more restrictive of this or battery current limit.
Voltage Filter <i>Battery_Power_Limit_Voltage_Filter</i> 0x31ED 0x00	0.02 – 99.5 Hz 66 – 32767	Sets the low pass filter cutoff frequency for the battery voltage measurement used in the battery power calculation. Lower values will filter the voltage signal more.

DUAL DRIVE MENU: See the Dual Drive Supplement, document # 53097

VEHICLE MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Metric Units <i>Metric_Units</i> OptionBits3 [Bit 5] 0x306C 0x00	On / Off On / Off	When this parameter is programmed On, the distance variables (Vehicle Odometer, Braking Distance Captured, Distance Since Stop, Distance Fine, and the Capture Distance variables) will accumulate and display in metric units (km, meters, or decimeters). When programmed Off, the distance variables will accumulate and display in English units (miles, feet, or inches). Distance variables are displayed in the Monitor » Vehicle menu.
Speed to RPM <i>Speed_to_RPM</i> 0x304C 0x00	10.0 – 3000.0 100 – 30000	This parameter affects the vehicle speed displayed in the Monitor » Motor menu, and also modifies the VCL variable Vehicle_Speed; it does not affect actual vehicle performance. The value entered for Speed to RPM is a conversion factor that scales motor speed to vehicle speed. KPH to RPM: $(G/d)*5305$, where G = gear ratio, d = tire diameter [mm]. MPH to RPM: $(G/d)*336.1$, where G = gear ratio, d = tire diameter [in].
Capture Speed 1 <i>Capture_Speed_1</i> 0x3079 0x00	0 – 10000 rpm 0 – 10000	The controller captures the time it takes the motor to go from 0 rpm to the programmed Capture Speed. The result is stored as “Time to Speed 1” in the Monitor » Vehicle menu. This timer starts every time the motor accelerates from zero speed.
Capture Speed 2 <i>Capture_Speed_2</i> 0x3886 0x00	0 – 10000 rpm 0 – 10000	This parameter allows a second capture speed to be defined, and works identically to Capture Speed 1. The result is stored as “Time to Speed 2” in the Monitor » Vehicle menu.
Capture Distance 1 <i>Capture_Distance_1</i> 0x307A 0x00	1 – 1320 1 – 1320	The controller captures the time it takes the vehicle to travel from 0 rpm to the programmed Capture Distance. The result is stored as “Time to Dist 1” in the Monitor » Vehicle menu. This timer starts every time the vehicle accelerates from zero speed. Note: For accurate distance measuring, the Speed to RPM parameter must be set correctly. With the Metric Units parameter programmed Off, distance is in units of feet. With Metric Units programmed On, distance is in units of meters.
Capture Distance 2 <i>Capture_Distance_2</i> 0x3883 0x00	1 – 1320 1 – 1320	This parameter allows a second capture distance to be defined, and works identically to Capture Distance 1. The result is stored as “Time to Dist 2” in the Monitor » Vehicle menu.
Capture Distance 3 <i>Capture_Distance_3</i> 0x3884 0x00	1 – 1320 1 – 1320	This parameter allows a third capture distance to be defined, and works identically to Capture Distance 1. The result is stored as “Time to Dist 3” in the Monitor » Vehicle menu.

Quick Links:

Monitor » Vehicle Menu [p.84](#)

Vehicle Speed

Vehicle Odometer

Time to Speed 1

Time to Speed 2

Time to Distance 1

Time to Distance 2

Time to Distance 3

EMERGENCY REVERSE MENU [SPEED MODE & SPEED MODE EXPRESS only]

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
EMR Enable <i>EMR_Enable</i> <i>OptionBits1 [Bit 1]</i> 0x306A 0x00	On / Off On / Off	Determines whether the emergency reverse function is active. On = emergency reverse is enabled. Off = emergency reverse is disabled.
EMR Type <i>EMR_Type</i> 0x3036 0x00	0 – 3 0 – 3	Determines where the input comes from for emergency reverse. 0 = emergency reverse activated by switch 1 (pin 24). 1 = emergency reverse is activated by VCL functions \Enable_Emer_Rev() and Disable_Emer_Rev(). 2 = emergency reverse activated by switch 1 (pin 24), but only if the vehicle is moving forward (forks trailing) when emergency reverse is activated. 3 = emergency reverse is activated by VCL functions Enable_Emer_Rev() and Disable_Emer_Rev(), but only if the vehicle is moving forward (forks trailing) when emergency reverse is activated.
EMR Dir Interlock <i>EMR_Dir_Interlock</i> <i>EMR_Dir_Interlock_Bit0 [Bit 0]</i> 0x389F 0x00	On / Off On / Off	Determines whether the interlock switch must be turned off after emergency reverse 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.
EMR Time Limit <i>EMR_Time_Limit</i> 0x303B 0x00	0.0 – 30.0 s 0 – 3750	Defines how long emergency reverse is allowed to be active after the vehicle is moving in the reverse direction. This timer will restart if the vehicle ever goes forward while emergency reverse is still active. The allowable range is 0–30 seconds, where 30 seconds is a special case of no time out. When emergency reverse times out, the Emer Rev Timeout fault is set. Cycling the emergency reverse input will clear the Emer Rev Timeout fault. To stop the vehicle after an EMR event (not move in reverse direction), set this parameter to 0.
EMR Speed <i>EMR_Speed</i> 0x3038 0x00	50 – 6000 rpm 50 – 6000	Defines the maximum reverse speed of the motor (in motor rpm), when emergency reverse is active. The motor speed in reverse during an emergency reverse event will be the lesser of EMR_Speed and either Max_Speed_SpdMx (for Speed Mode Express) or Max_Speed_SpdM (for Speed Mode). Emergency Reverse is not valid for Torque Mode. See Fig.16, Control Mode Processing.
EMR Accel Rate <i>EMR_Accel_Rate</i> 0x3039 0x00	0.1 – 3.0 s 100 – 3000	Sets the rate (in seconds) at which the vehicle accelerates in the opposite direction after it has been brought to a stop. If the vehicle is already traveling in the reverse direction below the EMR Speed, the EMR Accel Rate will bring the vehicle to the EMR Speed.
EMR Decel Rate <i>EMR_Decel_Rate</i> 0x303A 0x00	0.1 – 3.0 s 100 – 3000	Sets the rate (in seconds) at which the vehicle brakes to a stop when emergency reverse is activated and the vehicle is moving forward. If the vehicle is already traveling in the reverse direction above the EMR Speed, the EMR Decel Rate will bring the vehicle down to the EMR Speed.

Quick Link:Fig 16 [p.117](#)

INTERLOCK BRAKING MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Enable <i>Interlock_Brake_Enable</i> <i>OptionBits3 [Bit 7]</i> 0x306C 0x00	On / Off On / Off	Determines whether the interlock braking function is active. On = The controller will attempt to bring the vehicle to a stop using regen braking when the interlock signal is removed. Off = The controller will disable the bridge after Sequencing Delay expires and allow the vehicle to roll freely when the interlock signal is removed. This option is typically used only when there is a user controlled mechanical or hydraulic brake system.
Decel Rate HS <i>Interlock_Brake_Decel_Rate_HS</i> 0x309F 0x00	0.1 – 30.0 s 100 – 30000	Sets the rate (in seconds) that is used to slow down the vehicle when the interlock is released at high vehicle speeds (speeds > HS setting). Larger values represent slower response.
Decel Rate LS <i>Interlock_Brake_Decel_Rate_LS</i> 0x309E 0x00	0.1 – 30.0 s 100 – 30000	Sets the rate (in seconds) that is used to slow down the vehicle when the interlock is released at low vehicle speeds (speeds < LS setting). Larger values represent slower response.
Interlock Brake Timeout <i>Interlock_Brake_Timeout</i> 0x309C 0x00	0.0 – 8.0 s 0 – 1000	Controls the maximum allowable duration of an interlock braking event. The timer starts as soon as the interlock signal is removed. If the time expires before the vehicle has slowed below the Zero_Speed_Threshold, the EM brake will engage automatically. This timeout allows parallel usage of regen braking and the EM brake to reduce stopping distance. If Interlock Brake Timeout expires and the motor is still moving, regen braking will continue to retard vehicle motion in conjunction with the EM brake. NOTE: This parameter is only applicable when EM_Brake_Type = 1 or 2.
Settling Time <i>Interlock_Brake_Setting_Time</i> No CAN Object Index	0 – 8.0 Seconds 0 – 8000	Time after Zero Speed Threshold is reached that interlock braking is active and slowing the vehicle. Once this timer expires, the bridge will be disabled, and torque will be released. WARNING: This parameter should be set longer than Position Hold Settling Time if Torque Preload is enabled in EM Brake Control to make sure that the EM Brake sets before the torque is released.

Quick Links:

Speed Mode — Fine Tuning Menu [p.37](#)

HS (High Speed)

LS (Low Speed)

EM_Brake_Type [p.50](#)

CAN INTERFACE MENU

Quick Link:
CAN Status [p.85](#)

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
CANopen Interlock CANopen_Interlock_Enable OptionBits3 [Bit 2] 0x306C 0x00	On / Off On / Off	When programmed On, CAN NMT State must = 5 (operational state) in order for the interlock to be set; see Monitor » CAN Status menu.
CAN Node ID 1 CAN_Node_ID_1 0x3140 0x00	1 – 127 1 – 127	The Node ID for the primary microprocessor is determined by the state of switches 5 and 6 at KSI turn on: Node_ID_1 is selected when Sw 6 = Off and Sw 5 = Off. Node_ID_2 is selected when Sw 6 = Off and Sw 5 = On. Node_ID_3 is selected when Sw 6 = On and Sw 5 = Off. Node_ID_4 is selected when Sw 6 = On and Sw 5 = On. If the selected parameter has a Node ID of zero, the Node_ID_1 parameter is used. If the Node_ID_1 parameter is also zero, a default ID of 38 is used. Note, to disable the CAN Node ID switch arbitration (the parameter Node ID 1-4 swapping with switches), set all the Node IDs to the same value.
CAN Node ID 2 CAN_Node_ID_2 0x3141 0x00	1 – 127 1 – 127	See description of Node ID 1.
CAN Node ID 3 CAN_Node_ID_3 0x3146 0x00	1 – 127 1 – 127	See description of Node ID 1.
CAN Node ID 4 CAN_Node_ID_4 0x3147 0x00	1 – 127 1 – 127	See description of Node ID 1.
Supervisor Node ID Supervisor_Node_ID 0x3872 0x00	1 – 127 1 – 127	Sets the Node ID for the supervisor microprocessor. The Node ID is the first 7 bits of the 11-bit identifier (the COB ID).
Baud Rate CAN_Baud_Rate 0x3142 0x00	– 3 – 4 – 3 – 4	Sets the CAN baud rate for the CANopen system for this controller: – 3 = 20 kbit/s, – 2 = 50 kbit/s, – 1 = 100 kbit/s, 0 = 125 kbit/s, 1 = 250 kbit/s, 2 = 500 kbit/s, 3 = 800 kbit/s, 4 = 1000 kbit/s.
Heartbeat Rate CANopen_Heart_Beat_Rate 0x1017 0x00	16 – 200 ms 4 – 50	Sets the rate at which the CAN heartbeat messages are sent from this CANopen controller.
PDO1 Timeout Period CAN_PDO_Timeout_Period_1 0x3149 0x00	0 – 200 ms 0 – 50	Sets the PDO timeout period for the CANopen Ancillary system. After the ancillary 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. <i>This parameter previously applied to both PDO1 and PDO2. In previous OS's Both PDO RX messages needed to stop before a PDO timeout was declared.</i>
PDO2 Timeout Period CAN_PDO_Timeout_Period_2 0x314B 0x00	0 – 200 ms 0 – 50	Sets the PDO timeout period for the CANopen Ancillary system. After the ancillary 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 4 – 50	Sets the minimum rate between CAN emergency messages from this CANopen ancillary controller. This prevents quickly changing fault states from generating so many emergency messages that they flood the CANbus.
Suppress CANopen Init Suppress_CANopen_Init 0x3143 0x00	0 – 1 0 – 1	When Suppress CANopen Init is set = 1, at KSI On the initialization of the CANopen system is suppressed. Typically this is done so that the VCL program can make changes to the CANopen system before enabling it (by setting the variable Suppress_CANopen_Init = 0 and running the Setup_CAN() function).

Beginning with OS 32.0, eight new CAN mailboxes were added for use by VCL. The total is now 32.

See the System Information file (available from within the Curtis WinVCL program, Info tab).

RESET CONTROLLER PARAMETER

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Reset Controller <i>ResetController</i> 0x3B00 0x00	0 – 1 0 – 1	Setting this parameter = 1 will reset the controller. This reset is similar to resetting by cycling the keyswitch. Reset Controller has the advantage of resetting without causing the parameter block to be reloaded into the 1313/1314 programmer.

CLONING CONTROLLERS

(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 1313/1314 programmer can read all the information from a 1236E-5421 controller and write it to other 1236E-5421 controllers; however, it cannot write that same information to 1236E-5521 or 1236SE-5621 controllers.

To perform cloning, plug the programmer (1313 or 1314) into the controller that has the desired settings. While the steps using the 1313 versus the 1314 are slightly different in terms of screen images, the basic concept is the same: Save the parameters from the controller to clone, which is the act of saving a “.CPF” file (Curtis Program File) using the Save/Save As . . . operations. When saving the CPF file, chose a file name with meaning as this CPF file is used to write (clone) its data to subsequent controllers.

For example, using the 1314 PC Programming Station, with the controller to be updated connected, select “open” in the File menu, then navigate to the above mentioned (saved) CPF file, which will be loaded (indicated) below the menus bar. In the Communication menu, select the “Save file to connected system (F6)” icon, and follow the prompts to copy the settings into the controller being cloned (i.e., the “connected System” or controller). The 1313 is slightly different, working off the SD card where the CPF file was saved versus the PC’s hard drive, and then using the “Restore” feature to download the settings into the controller being cloned.

For cloning Dual Drive controllers, the Manager controller parameters will also be loaded into Ancillary controller, then the parameters for the Ancillary-vs-Manager must be adjusted using the 1313/1314, as corresponds to the differences between the manager and ancillary dual-drive controllers. See the separate Dual Drive manual, p/n 53097-DD, for the specific steps to perform for cloning/setup of the Manager and Ancillary Dual-Drive controllers.

Note: Using either programmer (1313/1314), the prompts will offer “Advanced Cloning” . . . Advanced Cloning is advisable because it restores all parameters, including those at a higher (hidden level). Yet, due to possible conflicts in a non-standard application or OS version, select “No” in the 1313 prompt, or do not “check” the box in the 1314 if there are any questions. Please contact the Curtis distributor or support engineer if there are any question on cloning controllers.

5 — MONITOR MENU

Through its 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 also while adjusting programmable parameters and the vehicle's initial setup.

Table 4 Monitor Menu: 1313/1314 Programmer

INPUTS MENU..... p. 77 — Throttle Command — Throttle Multiplier — Mapped Throttle — Throttle Pot — Brake Command — Mapped Brake — Pot2 Raw — PD Throttle — Steer Pot — Steer Angle (deg) — Interlock — Emer Rev — Analog 1 — Analog 2 — Switch 1 — Switch 2 — Switch 3 — Switch 4 — Switch 5 — Switch 6 — Switch 7 — Switch 8 — Driver 1 Input — Driver 2 Input — Driver 3 Input — Driver 4 Input — PD Input — DigOut6 Input — DigOut7 Input — Switch 16	OUTPUTS MENU..... p. 79 — Analog Out — Digital Out 6 — Digital Out 7 — Driver 1 PWM — Driver 2 PWM — Driver 3 PWM — Driver 4 PWM — PD PWM — PD Current — 5 Volts — 12 Volts — Ext Supply Current — Pot Low	— VCL Error — PMAC Commissioning Status — Motor Characterization Error — Parameter Mismatch Error — Supervision Error — CUTBACKS..... p. 83 — Motor Temp Cutback — Controller Temp Cutback — Undervoltage Cutback — Overvoltage Cutback — Battery Current Cutback — MOTOR TUNING p. 83 — Motor RPM — Base Speed Captured — Test Field Current — FOLLOWING ERROR -SPEED MODE p. 83 — Following Error — Following Error Timer
	BATTERY MENU..... p. 80 — BDI — Capacitor Voltage — Keyswitch Voltage	
	MOTOR MENU..... p. 80 — Motor RPM — Max Speed Controller Limit — Temperature — MotorSpeed A — MotorSpeed B — Sin Input A — Cos Input B — Rotor Position	
	CONTROLLER MENU..... p. 81 — Current (RMS) — Modulation Depth — Frequency — Temperature — Main State — EMBrake State — EMBrake Test State — Regen — KSI Hour Meter — VCL Error Module	VEHICLE MENU..... p. 84 — Vehicle Speed — Vehicle Odometer — Vehicle Acceleration — Time to Speed 1 — Time to Speed 2 — Time Between Speeds — Time to Dist 1 — Time to Dist 2 — Time to Dist 3 — Braking Distance Captured — Distance Since Stop — Distance Fine
		CAN STATUS MENU..... p. 85 — CAN Node ID — CAN NMT State — PDO1 RX Byte Map — PDO1 TX Byte Map — PDO2 RX Byte Map — PDO2 TX Byte Map

Monitor Menu: INPUTS

VARIABLE	DISPLAY RANGE	DESCRIPTION
Throttle Command <i>Throttle_Command</i> 0x3216 0x00	–100.0 – 100.0% –32767 – 32767	Throttle request to slew rate block.
Throttle Multiplier <i>Throttle_Multiplier</i> 0x3213 0x00	–200.0 – 200.0% –256 – 256	Multiplies or divides the throttle signal; useful in VCL throttle processing (see Throttle Processing and Figure 15).
Mapped Throttle <i>Mapped_Throttle</i> 0x3211 0x00	–100.0 – 100.0% –32767 – 32767	Mapped throttle request. (see Throttle Processing and Figure 15).
Throttle Pot <i>Throttle_Pot_Raw</i> 0x3215 0x00	0.00 – 5.50 V 0 – 36044	Voltage at throttle pot wiper (pin 16).
Brake Command <i>Brake_Command</i> 0x321A 0x00	0.0 – 100.0% 0 – 32767	Brake request to slew rate block. (see Brake Processing and Figure 15).
Mapped Brake <i>Mapped_Brake</i> 0x3212 0x00	0.0 – 100.0% 0 – 32767	Mapped brake request. (see Brake Processing and Figure 15).
Pot2 Raw <i>Pot2_Raw</i> 0x3217 0x00	0.00 – 5.50 V 0 – 36044	Voltage at pot2 wiper (pin 17).
PD Throttle <i>PD_Throttle</i> 0x3210 0x00	0.00 – 100.00 % 0 – 32766	Proportional driver current request. (see Interfacing the Proportional Current Driver and Figure 17).
Steer Pot <i>Steer_Pot_Raw</i> 0x38BB 0x00	0.00 – 6.25 V 0 – 32767	Voltage at steer pot wiper (pin 17) on Dual Drive traction ancillary controller. (reference the Dual Drive supplement manual: 53097_E&SEDD_os37).
Steer Angle (deg) <i>Steer_Angle</i> 0x38BA 0x00	–90 – 90 –90 – 90	Steer angle degrees calculated in Dual Drive traction manager.
Interlock <i> Interlock_State</i> <i> System_Flags1 [Bit 0]</i> 0x322B 0x00	On / Off On / Off	Interlock input on or off. The source of the interlock input is determined by the Interlock Type parameter. from Switch 3 (pin 9) if Interlock Type = 0. from VCL function if Interlock Type = 1. from KSI (pin 1) if Interlock Type = 2.
Emer Rev <i> EMR_State</i> <i> System_Flags1 [Bit 1]</i> 0x322B 0x00	On / Off On / Off	Emergency reverse input on or off. The source of the emergency reverse input is determined by the EMR Type parameter: from Switch 1 (pin 24) if EMR Type = 0, or 2. from VCL function if EMR Type = 1.
Analog 1 <i>Analog1_Input</i> 0x3204 0x00	0.00 – 10.00 V 0 – 1023	Voltage at analog 1 (pin 24).
Analog 2 <i>Analog2_Input</i> 0x3205 0x00	0.00 – 10.00 V 0 – 1023	Voltage at analog 2 (pin 8).
Switch 1 <i> Sw_1</i> <i> Switches [Bit 0]</i> 0x3226 0x00	On / Off On / Off	Switch 1 on or off (pin 24).
Switch 2 <i> Sw_2</i> <i> Switches [Bit 1]</i> 0x3226 0x00	On / Off On / Off	Switch 2 on or off (pin 8).

Quick Links:

Throttle Processing [p.113](#) Figure 15 [p.114](#)
Brake Processing [p.115](#) Prop Driver [p.118](#)

Figure 17 [p.118](#)

Monitor Menu: INPUTS, cont'd

VARIABLE	DISPLAY RANGE	DESCRIPTION
Switch 3 Sw_3 Switches [Bit 2] 0x3226 0x00	On / Off <i>On / Off</i>	Switch 3 on or off (pin 9).
Switch 4 Sw_4 Switches [Bit 3] 0x3226 0x00	On / Off <i>On / Off</i>	Switch 4 on or off (pin 10).
Switch 5 Sw_5 Switches [Bit 4] 0x3226 0x00	On / Off <i>On / Off</i>	Switch 5 on or off (pin 11).
Switch 6 Sw_6 Switches [Bit 5] 0x3226 0x00	On / Off <i>On / Off</i>	Switch 6 on or off (pin 12).
Switch 7 Sw_7 Switches [Bit 6] 0x3226 0x00	On / Off <i>On / Off</i>	Switch 7 on or off (pin 22).
Switch 8 Sw_8 Switches [Bit 7] 0x3226 0x00	On / Off <i>On / Off</i>	Switch 8 on or off (pin 33).
Driver 1 Input Sw_9 Switches [Bit 8] 0x3226 0x00	On / Off <i>On / Off</i>	Driver 1 input on or off (pin 6).
Driver 2 Input Sw_10 Switches [Bit 9] 0x3226 0x00	On / Off <i>On / Off</i>	Driver 2 input on or off (pin 5).
Driver 3 Input Sw_11 Switches [Bit 10] 0x3226 0x00	On / Off <i>On / Off</i>	Driver 3 input on or off (pin 4).
Driver 4 Input Sw_12 Switches [Bit 11] 0x3226 0x00	On / Off <i>On / Off</i>	Driver 4 input on or off (pin 3).
PD Input Sw_13 Switches [Bit 12] 0x3226 0x00	On / Off <i>On / Off</i>	Proportional driver on or off (pin 2).
DigOut6 Input Sw_14 Switches [Bit 13] 0x3226 0x00	On / Off <i>On / Off</i>	Digital Out 6 input on or off (pin 19).
DigOut7 Input Sw_15 Switches [Bit 14] 0x3226 0x00	On / Off <i>On / Off</i>	Digital Out 7 input on or off (pin 20).
Switch 16 Sw_16 Switches [Bit 15] 0x3226 0x00	On / Off <i>On / Off</i>	Switch 16 on or off (pin 14).

Monitor Menu: OUTPUTS

VARIABLE	DISPLAY RANGE	DESCRIPTION
Analog Out <i>Analog_Output</i> 0x321D 0x00	0.00 – 10.00V <i>0 – 32767</i>	Voltage at Analog output (pin 30).
Digital Out 6 <i>Dig6_Output</i> <i>System_Flags1 [Bit 3]</i> 0x322B 0x00	On / Off <i>On / Off</i>	Digital Out 6 output on or off (pin 19).
Digital Out 7 <i>Dig7_Output</i> <i>System_Flags1 [Bit 4]</i> 0x322B 0x00	On / Off <i>On / Off</i>	Digital Out 7 output on or off (pin 20).
Driver 1 PWM <i>PWM1_Output</i> 0x321F 0x00	0 – 100% <i>0 – 32767</i>	Driver 1 PWM output (pin 6).
Driver 2 PWM <i>PWM2_Output</i> 0x3220 0x00	0 – 100% <i>0 – 32767</i>	Driver 2 PWM output (pin 5).
Driver 3 PWM <i>PWM3_Output</i> 0x3221 0x00	0 – 100% <i>0 – 32767</i>	Driver 3 PWM output (pin 4).
Driver 4 PWM <i>PWM4_Output</i> 0x3222 0x00	0 – 100% <i>0 – 32767</i>	Driver 4 PWM output (pin 3).
PD PWM <i>PD_Output</i> 0x321E 0x00	0 – 100% <i>0 – 32767</i>	Proportional driver PWM output (pin 2).
PD Current <i>PD_Current</i> 0x321C 0x00	0.00 – 2.00A <i>0 – 607</i>	Current at proportional driver (pin 2).
5 Volts <i>Five_Volts_Output</i> 0x3202 0x00	0.00 – 6.25V <i>0 – 1023</i>	Voltage at +5V output (pin 26).
12 Volts <i>Twelve_Volts_Output</i> 0x3522 0x00	0.00 – 12.00V <i>0 – 768</i>	Voltage at +12V output (pin 25).
Ext Supply Current <i>Ext_Supply_Current</i> 0x3229 0x00	0 – 200 mA <i>0 – 800</i>	Combined current of the external +12V and +5V voltage supplies (pins 25 and 26).
Pot Low <i>Pot_Low_Output</i> 0x3203 0x00	0.00 – 6.25V <i>0 – 1023</i>	Voltage at pot low (pin 18).

Monitor Menu: BATTERY

VARIABLE	DISPLAY RANGE	DESCRIPTION
BDI <i>BDI_Percentage</i> 0x3308 0x00	0 – 100% <i>0 – 100</i>	Lead-acid battery discharge indicator (BDI). Effectively the battery's remaining state-of-charge. NOTE: For non lead-acid batteries, including Lithium-Ion battery packs, use the pack's or cell manufacturer's approved Battery Management System (BMS) for determining the BDI.
Capacitor Voltage <i>Capacitor_Voltage</i> 0x324C 0x00	0.0 – 200.0V <i>0 – 12800</i>	Voltage of controller's internal capacitor bank at B+ terminal.
Keyswitch Voltage <i>Keyswitch_Voltage</i> 0x324D 0x00	0.0 – 105.0V <i>0 – 10500</i>	Voltage at KSI (pin 1).

Monitor Menu: MOTOR

VARIABLE	DISPLAY RANGE	DESCRIPTION
Motor RPM <i>Motor_RPM_Display</i> 0x3559 0x00	–12000 – 12000 rpm <i>–12000 – 12000</i>	Motor speed in revolutions per minute. Filtered for display on the 1313/1314. No CAN Index is assigned to this monitor variable. Possible usage for a tachometer, due to the filtered (smoothing) nature of this VCL variable over the Motor_RPM variable (see Motor RPM in monitor Controller » Motor Tuning » Motor_RPM).
Max Speed Controller Limit <i>Max_Speed_Controller_Limit</i> 0x3559 0x00	0 – 12000 rpm <i>0 – 12000</i>	Displays the maximum allowed speed, which clamps the upper limit of the Max Speed parameters. It is the least of the three motor-speed constraints described in Chapter 3.
Temperature <i>Motor_Temperature</i> 0x320B 0x00	–100 – 300°C <i>–1000 – 3000</i>	Temperature sensor readout.
MotorSpeed A <i>MotorspeedA</i> 0x35D1 0x00	–12000 – 12000 rpm <i>–12000 – 12000</i>	Motor encoder phase A speed in revolutions per minute. This can be used to verify that phase A of the encoder is operating correctly. MotorSpeed A should equal MotorSpeed B in a properly operating motor encoder. MotorSpeed A does not indicate direction.
MotorSpeed B <i>MotorspeedB</i> 0x35D2 0x00	–12000 – 12000 rpm <i>–12000 – 12000</i>	Motor encoder phase B speed in revolutions per minute. This can be used to verify that phase B of the encoder is operating correctly. MotorSpeed B should equal MotorSpeed A in a properly operating motor encoder. MotorSpeed B does not indicate direction.
Sin Input A <i>encoder_sin_calibrated</i> 0x38C5 0x00	–1023 – 1023 <i>–1023 – 1023</i>	Input at Position Feedback A (pin 31). This parameter has scaling changes and a variable name change.
Cos Input A <i>encoder_cos_calibrated</i> 0x38C6 0x00	–1023 – 1023 <i>–1023 – 1023</i>	Input at Position Feedback B (pin 32). This parameter has scaling changes and a variable name change.
Rotor Position <i>rotor_position_raw</i> 0x38C7 0x00	–32768 – 32767 <i>–32768 – 32767</i>	The displayed value will never exceed 4095. 0 – 4095 maps to mechanical 0°– 360°.

Quick Link:

[Motor Speed Constraints p.24](#)

Monitor Menu: CONTROLLER

VARIABLE	DISPLAY RANGE	DESCRIPTION
Current (RMS) <i>Current_RMS</i> 0x3209 0x00	0.0 – 1000.0 A <i>0 – 10000</i>	RMS current of the controller, taking all three phases into account.
Modulation Depth <i>Modulation_Depth</i> 0x3208 0x00	0.0 – 100.0% <i>0 – 1182</i>	Percentage of available voltage being used.
Frequency <i>Frequency</i> 0x3206 0x00	–1000.0 – 1000.0 Hz <i>–3000 – 3000</i>	Controller electrical frequency. This parameter has scaling changes to support higher frequencies.
Temperature <i>Controller_Temperature</i> 0x322A 0x00	–100 – 300°C <i>–1000 – 3000</i>	Controller internal temperature.
Main State <i>Main_State</i> 0x3223 0x00	0 – 10 <i>0 – 10</i>	Main contactor state: 0 = Open. 1 = Precharge. 2 = Weld check. 3 = Closing delay. 4 = Missing check. 5 = Closed (when Main Enable = On). 6 = Delay. 7 = Arc check. 8 = Open delay. 9 = Fault. 10 = Closed (when Main Enable = Off).
EMBrake State <i>EMBrakeState</i> 0x3892 0x00	0 – 4 <i>0 – 4</i>	EM brake state: 0 = Engaged. 1 = Releasing. 2 = Released. 3 = Engaging. 4 = Engaged and vehicle stopped.
EMBrake Test State <i>EM_Brake_Test_State</i> 0x30E8 0x00	0 – 4 <i>0 – 4</i>	The current state of the EM brake test. Can be stop (0), starting (1), running (2), done (3), or failed (4).
Regen <i>Regen_State</i> <i>System_Flags1 [Bit 2]</i> 0x322B 0x00	On / Off <i>On / Off</i>	On when regen braking is taking place; Off when it is not.
KSI Hour Meter <i>KSI_Hour_Meter</i> 0x3160 0x00	0 – 429496729.5 s <i>0 – 4294967295</i>	This is the total hours the controller has been powered (keyswitch hours). The timer is controlled by the OS software and cannot be reset.
VCL Error Module <i>Last_VCL_Error_Module</i> 0x3473 0x00	0 – 65535 <i>0 – 65535</i>	A VCL Runtime Error (fault code 68) will store additional information about the cause of a VCL runtime error in the VCL Error Module and VCL Error variables. The resulting non-zero values can be compared to the runtime VCL module ID and error code definitions listed in the controller's OS SysInfo file, which should help pinpoint the VCL error that caused the runtime error.
VCL Error <i>Last_VCL_Error</i> 0x3472 0x00	0 – 65535 <i>0 – 65535</i>	

Monitor Menu: CONTROLLER, cont'd

VARIABLE	DISPLAY RANGE	DESCRIPTION
PMAC Commissioning Status <i>PMAC_Commissioning_State_Display</i> 0x3879 0x00	0 – 8 0 – 8	0 - Motor Needs Commissioning. 1 - Commissioning Not Started. 2 - Align Rotor to Field. 3 - Current Regulator Tuning. 4 - Sin/Cos Peak Voltage Detection. 5 - Determine Phasing. 6 - Determine Sin/Cos Compensation. 7 - Commissioning Complete. 8 - Commissioning Error.
Motor Characterization Error <i>Motor_Characterization_Error</i> 0x3899 0x00	0 – 24 0 – 24	A Motor Characterization Fault (fault code 87) will store additional information in the Motor Characterization Error variable: 0 = Sequencing error. Normally caused by turning off Motor Characterization Test Enable before running the test. 1 = Encoder signal seen but step size not auto-detected; set up Encoder Steps manually. 2 = Motor temp sensor fault. 3 = Motor temp hot cutback fault. 4 = Controller overtemp cutback fault. 5 = Reserved. 6 = Undervoltage cutback fault. 7 = Severe overvoltage fault. 8 = Encoder signal not seen, or one or both channels missing. 9 = Motor parameters out of characterization range. 20 = Sin/Cos sensor not found. 21 = Phasing not detected. 22 = Sin/Cos sensor characterization failure. 23 = Started characterization procedure while motor rotating. 24 = Sin/Cos signal too noisy for characterization.
Notes: Errors 1 and 8 apply to ACIM motors only. Errors 10, 20, 21, and 23 apply to PMAC motors only. Errors indicate the motor characterization data is invalid, except in the case of Error 1.		
Parameter Mismatch Error <i>Parameter_Mismatch_Error</i> 0x388F 0x00	0 – 10 0 – 10	A Parameter Mismatch Fault (fault code 99) will store additional information in the Parameter Mismatch Error variable: 0 = No Parameter Mismatch Error Present. 1 = Dual Motor Enable = On, and Control Mode Select is not set to either 0-Speed Mode Express or 1-Speed Mode. 2 = Feedback Type is incompatible with Motor Technology setting. 3 = Save Torque Preload = On and Torque Preload Enable = Off. 4 = Speed Filter Mapping Incorrect. 5 = This Control Mode does not support EM Brake Control. 6 = This Control Mode does not support Interlock Braking. 7 = Unused. 8 = Controller not compatible with motor, short circuit current too high. 9 = Single voltage models require cross connect. 10 = Cannot run restricted operation without back emf data. 11 = Mode not yet intended for production use. 12 = This Motor Type not compatible with Direct Torque Mode. 13 = This Motor Type not compatible with Released OS. 14 = Last Step of PMAC Commissioning Skipped. 15 = PMAC Motor Type 0 disabled. 16 = This Throttle Type not compatible with Direct Torque Mode.

Monitor Menu: CONTROLLER, cont'd

VARIABLE	DISPLAY RANGE	DESCRIPTION
Supervision Error <i>Supervision_Error</i> 0x3897 0x00	0 – 4 0 – 4	A Supervision Fault (fault code 77) will store additional information in the Supervision Error variable: 0 = None. 1 = Watchdog timeout comms. 2 = Power supply fault detected. 3 = Queue fault detected. 4 = ALU fault detected. The Supervisor Fault (flash code 77) has several causes. The most common cause is when one of the digital inputs is stuck in the range between the guaranteed high threshold and the guaranteed low threshold for more than 100 milliseconds.

Monitor Menu: CUTBACKS

VARIABLE	DISPLAY RANGE	DESCRIPTION
Motor Temp Cutback <i>MotorTempCutback</i> 0x3581 0x00	0 – 100% 0 – 4096	Displays the current available as a result of the motor temperature cutback function. A value of 100% indicates no cutback in current.
Controller Temp Cutback <i>ControllerTempCutback</i> 0x35F3 0x00	0 – 100% 0 – 4096	Displays the current available as a result of the controller temperature cutback function. A value of 100% indicates no cutback in current.
Undervoltage Cutback <i>UndervoltageCutback</i> 0x3605 0x00	0 – 100% 0 – 4096	Displays the current available as a result of the undervoltage cutback function. A value of 100% indicates no cutback in current.
Overvoltage Cutback <i>OvervoltageCutback</i> 0x3604 0x00	0 – 100% 0 – 4096	Displays the current available as a result of the overvoltage cutback function. A value of 100% indicates no cutback in current.
Battery Current Cutback <i>BatteryCurrentCutback</i> No CAN Object	0 – 100% 0 – 4096	Displays the current available as a result of the battery current cutback function. A value of 100% indicates no cutback in the battery current.

Monitor Menu: MOTOR TUNING

VARIABLE	DISPLAY RANGE	DESCRIPTION
Motor RPM <i>Motor_RPM</i> 0x3207 0x00	–32768 – 32767 rpm –32768 – 32767	Motor speed in revolutions per minute. This is the raw value (compare to the filtered value under the Motor menu). Use Motor_RPM for VCL and CAN applications.
Base Speed Captured <i>Base_Speed_Captured</i> 0x35BE 0x00	0 – 65535 rpm 0 – 65535	Displays the value of the motor base speed captured in the most recent acceleration. This value is used to set the FW Base Speed parameter, using the FW Base Speed set procedure or using the ACIM motor characterization procedure (Chapter 8A).
Test Field Current <i>Test_Field_Current</i> 0x3873 0x00	0 – 800 A 0 – 800	Field test current reading to be used during the ACIM motor characterization procedure (Chapter 8A).

Monitor Menu: FOLLOWING ERROR – SPEED MODE

VARIABLE	DISPLAY RANGE	DESCRIPTION
Following Error <i>Following_Error</i> 0x3555 0x00	–1200 – 1200 rpm –1200 – 1200	The signal error between the commanded motor speed trajectory and the actual motor RPM.
Following Error Timer <i>Following_Error_Timer</i> 0x38CA 0x00	0 – 5.000 S 0 – 2.500	The time that the following error exceeded the parameterized limit. If this time exceeds the parameterized time limit, the system will set a flowing error fault.

Monitor Menu: VEHICLE

VARIABLE	DISPLAY RANGE	DESCRIPTION
Vehicle Speed <i>Vehicle_Speed</i> 0x320A 0x00	-3276.8 – 3276.7 <i>-32768 – 32767</i>	Vehicle speed, in units of MPH or KPH, depending on the setting of the Metric Units parameter (see Program » Vehicle menu). For accurate speed estimates, the Speed to RPM parameter must be set correctly.
Vehicle Odometer <i>Vehicle_Odometer</i> 0x3200 0x00	0.0 – 10000000.0 <i>0 – 100000000</i>	Vehicle distance traveled, in units of miles or km, depending on the setting of the Metric Units parameter in the Vehicle parameters menu. For accurate distance measurements, the Speed to RPM parameter must be set correctly.
Vehicle Acceleration <i>Vehicle_Acceleration</i> 0x35C1 0x00	0.000 – 10.000 g <i>0 – 10000</i>	Vehicle acceleration. This is a calculated value. The Speed to RPM parameter must be set correctly for an accurate measurement.
Time to Speed 1 <i>Time_to_Capture_Speed_1</i> 0x35BF 0x00	0.00 – 128.00 s <i>0 – 32000</i>	Time taken for the vehicle to go from zero rpm to the programmed Capture Speed 1 (see Program » Vehicle menu) during its most recent such acceleration.
Time to Speed 2 <i>Time_to_Capture_Speed_2</i> 0x35BB 0x00	0.00 – 128.00 s <i>0 – 32000</i>	Time taken for the vehicle to go from zero rpm to the programmed Capture Speed 2 (see Program » Vehicle menu) during its most recent such acceleration.
Time Between Speeds <i>Time_Between_Capture_Speeds</i> 0x35C4 0x00	0.00 – 128.00 s <i>0 – 32000</i>	Time taken for the vehicle to go from programmed Capture Speed 1 to programmed Capture Speed 2 (see Program » Vehicle menu) during its most recent such acceleration.
Time to Dist 1 <i>Time_to_Capture_Distance_1</i> 0x3612 0x00	0.00 – 128.00 s <i>0 – 32000</i>	Time taken for the vehicle to travel from zero rpm to the programmed Capture Distance 1 (see Program » Vehicle menu) during its most recent such trip. For accurate distance measurements, the Speed to RPM parameter must be set correctly.
Time to Dist 2 <i>Time_to_Capture_Distance_2</i> 0x3613 0x00	0.00 – 128.00 s <i>0 – 32000</i>	Time taken for the vehicle to travel from zero rpm to the programmed Capture Distance 2 (see Program » Vehicle menu) during its most recent such trip. For accurate distance measurements, the Speed to RPM parameter must be set correctly.
Time to Dist 3 <i>Time_to_Capture_Distance_3</i> 0x3614 0x00	0.00 – 128.00 s <i>0 – 32000</i>	Time taken for the vehicle to travel from zero rpm to the programmed Capture Distance 3 (see Program » Vehicle menu) during its most recent such trip. For accurate distance measurements, the Speed to RPM parameter must be set correctly.
Braking Distance Captured <i>Braking_Distance_Captured</i> 0x3201 0x00	0.0 – 1000000.0 <i>0 – 40000000</i>	Distance traveled by the vehicle starting with vehicle braking (initiated by throttle reversal, brake pot, VCL_Brake, or interlock braking) and ending when Motor_RPM = 0. Units are meters or feet, depending on the setting of the Metric Units parameter. For accurate distance measurements, the Speed to RPM parameter must be set correctly.
Distance Since Stop <i>Distance_Since_Stop</i> 0x324E 0x00	0.0 – 1000000.0 <i>0 – 40000000</i>	Distance traveled by the vehicle starting from a stop. In effect, the vehicle is used as a tape measure. (In other words, if the vehicle travels 300 feet forward and then 300 feet in reverse, the distance would be 600.) The distance is continuously updated and will stop (and restart) when Motor_RPM = 0. For accurate distance measurements, the Speed to RPM parameter must be set correctly. Units are meters or feet, depending on the setting of the Metric Units parameter.
Distance Fine <i>Distance_Fine_Long</i> 0x3616 0x00	-214748364.8 – 214748364.7 <i>-2147483648 – 2147483647</i>	Position measurement. Net distance in both the forward and reverse directions. (In other words, if the vehicle travels 20 inches forward and then 20 inches in reverse, the distance would be zero.) The distance is continuously updated and will roll over when the variable goes over the limits. Resets to zero on key cycle. Units are decimeters or inches, depending on the setting of the Metric Units parameter. For accurate distance measurements, the Speed to RPM parameter must be set correctly.

NOTE: All vehicle calculations assume no tire slippage.

Quick Links:

[Vehicle Menu parameters](#) [p.71](#)[Metric Units](#)[Speed to RPM](#)[Capture Speed 1, 2](#)[Capture Distance 1, 2, 3](#)

Monitor Menu: CAN STATUS

VARIABLE	DISPLAY RANGE	DESCRIPTION
CAN Node ID <i>CAN_Node_ID</i> 0x3145 0x00	0 – 127 0 – 127	Displays the controller's Node ID.
CAN NMT State <i>CAN_NMT_State</i> 0x3328 0x00	0 – 127 0 – 127	Controller CAN NMT state: 0=initialization, 4=stopped, 5=operational, 127=pre-operational.
PD01 RX Byte Map*	0 – 2 ³²	Mapping objects for RPD01 RX (receive) eight bytes.
PD01 TX Byte Map*	0 – 2 ³²	Mapping objects for TPD01 TX (transmit) eight bytes.
PD02 RX Byte Map*	0 – 2 ³²	Mapping objects for RPD02 RX (receive) eight bytes.
PD02 TX Byte Map*	0 – 2 ³²	Mapping objects for TPD02 TX (transmit) eight bytes.

* Each of these byte maps is a submenu containing 8 variables, one for each byte. Each variable is 32 bits.

RPD01 RX Byte Map

<i>CAN_PD01_RX_MAP_1</i>	0x1600 0x01
<i>CAN_PD01_RX_MAP_2</i>	0x1600 0x02
<i>CAN_PD01_RX_MAP_3</i>	0x1600 0x03
<i>CAN_PD01_RX_MAP_4</i>	0x1600 0x04
<i>CAN_PD01_RX_MAP_5</i>	0x1600 0x05
<i>CAN_PD01_RX_MAP_6</i>	0x1600 0x06
<i>CAN_PD01_RX_MAP_7</i>	0x1600 0x07
<i>CAN_PD01_RX_MAP_8</i>	0x1600 0x08

TPD01 TX Byte Map

<i>CAN_PD01_TX_MAP_1</i>	0x1A00 0x01
<i>CAN_PD01_TX_MAP_2</i>	0x1A00 0x02
<i>CAN_PD01_TX_MAP_3</i>	0x1A00 0x03
<i>CAN_PD01_TX_MAP_4</i>	0x1A00 0x04
<i>CAN_PD01_TX_MAP_5</i>	0x1A00 0x05
<i>CAN_PD01_TX_MAP_6</i>	0x1A00 0x06
<i>CAN_PD01_TX_MAP_7</i>	0x1A00 0x07
<i>CAN_PD01_TX_MAP_8</i>	0x1A00 0x08

RPD02 RX Byte Map

<i>CAN_PD02_RX_MAP_1</i>	0x1601 0x01
<i>CAN_PD02_RX_MAP_2</i>	0x1601 0x02
<i>CAN_PD02_RX_MAP_3</i>	0x1601 0x03
<i>CAN_PD02_RX_MAP_4</i>	0x1601 0x04
<i>CAN_PD02_RX_MAP_5</i>	0x1601 0x05
<i>CAN_PD02_RX_MAP_6</i>	0x1601 0x06
<i>CAN_PD02_RX_MAP_7</i>	0x1601 0x07
<i>CAN_PD02_RX_MAP_8</i>	0x1601 0x08

TPD02 TX Byte Map

<i>CAN_PD02_TX_MAP_1</i>	0x1A01 0x01
<i>CAN_PD02_TX_MAP_2</i>	0x1A01 0x02
<i>CAN_PD02_TX_MAP_3</i>	0x1A01 0x03
<i>CAN_PD02_TX_MAP_4</i>	0x1A01 0x04
<i>CAN_PD02_TX_MAP_5</i>	0x1A01 0x05
<i>CAN_PD02_TX_MAP_6</i>	0x1A01 0x06
<i>CAN_PD02_TX_MAP_7</i>	0x1A01 0x07
<i>CAN_PD02_TX_MAP_8</i>	0x1A01 0x08

NOTE: RX (PDO request from the manager)
TX (PDO response from the ancillary)

6 – CONTROLLER INFORMATION MENU

This menu provides ID and version numbers for the connected controller's hardware and software.

CONTROLLER INFORMATION MENU

VARIABLE	DISPLAY RANGE	DESCRIPTION
Serial Number <i>Serial_Number</i> 0x1018 0x04	0 – 4294967295 <i>0 – 4294967295</i>	Serial number. For example, if the serial number printed on the controller is 15045L.11493, the Serial Number variable will have the value of 11493.
Model Number <i>Model_Number</i> 0x3464 0x00	0 – 4294967295 <i>0 – 4294967295</i>	Model number. For example, for a 1236E controller with the model number 1236E-4521, the Model Number variable will have a value of 12364521.
Mfg Date Code <i>Manuf_Date</i> 0x3466 0x00	0 – 32767 <i>0 – 32767</i>	Controller date of manufacture, with the first two digits indicating the year and the last three indicating the day. For example, if the serial number printed on the controller is 15045L.11493, the Mfg Date Code variable will have the value of 15045 (45th day of 2015).
Hardware Version <i>Hardware_Ver</i> 0x3467 0x00	0 – 32.767 <i>0 – 32767</i>	The hardware version number uniquely describes the combination of power base assembly and the logic, cap, and IMS board assemblies used in the controller.
OS Version <i>OS_Ver</i> 0x346A 0x00	0 – 65535 <i>0 – 65535</i>	Version number of the operating system software that is loaded into the controller. This variable specifies the <u>major</u> version number of the controller's operating system.
Build Number <i>TBuild_Number</i> 0x346B 0x00	0 – 65535 <i>0 – 65535</i>	Build number of the operating system software that is loaded into the controller. This variable specifies the <u>minor</u> version number of the controller's operating system.
SM Version <i>SM_Ver</i> 0x332B 0x00	0 – 327.67 <i>0 – 32767</i>	Version number of the Start Manager software that is loaded into the controller.
Param Blk Version <i>Param_Blk_Ver</i> 0x3468 0x00	0 – 327.67 <i>0 – 32767</i>	Version number of the parameter block that is loaded into the controller.
VCL App Version <i>VCL_App_Ver</i> 0x3463 0x00	0 – 327.67 <i>0 – 32767</i>	Version number of the VCL application software that is loaded into the controller. This value is set in the VCL program by assigning a value to the VCL_App_Ver variable.
Vehicle Serial Number <i>Vehicle_Serial_Number</i> 0x393D 0x00	0 – 4294967295 <i>0 – 4294967295</i>	Provided for OEM use and definition.

7 — INITIAL SETUP

The 1232E/SE, 1234E/SE, 1236E/SE, and 1238E/SE controllers can be used in a variety of vehicles, which differ widely in characteristics. Before driving the vehicle, it is imperative that these initial setup procedures be carefully followed to ensure that the controller is set up to be compatible with the application.

BEFORE BEGINNING

Correct values for the motor parameters (Motor Type, FW Base Speed, and Field Weakening) must be determined individually for each motor.

For ACIM motors, use one of these three options to obtain ACIM motor data:

- Contact the Curtis distributor or support engineer with the manufacturer's part number for the motor. Curtis has a database of many motors for which the motor data has already been determined. Motors in the Curtis database are assigned a number, which is the number used for the ACIM Motor Type parameter.
- Send the ACIM motor to Curtis for testing on the factory motor dyno. The motor's data will be entered into the Curtis database and issue updated software (OS) with the appropriate parameter values set to match the motor. Contact the Curtis distributor or support engineer before shipping the motor.
- Use the ACIM Motor Characterization Procedure (Chapter 8A), which has the controller "learn" the motor parameter data. The procedure should take about half an hour and can be done on the vehicle. Go ahead and complete setup steps 1 through 10 of the Initial Setup before conducting the ACIM characterization procedure.

For PMAC motors, use the PMAC Motor Commissioning Procedure (Chapter 8B). This PMAC motor commissioning has the controller "learn" the motor parameter data. The procedure should take about 10 minutes and can be completed on the vehicle (with the drive wheels off the ground- free to rotate). First complete these Initial Setup steps 1 through 9 before conducting the PMAC commissioning. For PMAC motor data,

- Contact Curtis for the motor number that matches the application's motor.
- Send the PMAC motor to Curtis for testing on the Curtis motor dynamometer (dyno) located in California. Once characterized, the motor's data goes into the Curtis database. Contact the Curtis distributor or support engineer before shipping the motor. Shipping and characterizing a motor required extra time. This process can take weeks to months, plus shipping.

ACIM:

If Curtis has provided the values for Motor Type, FW Base Speed, and Field Weakening and they are "programmed" into the controller (see ACIM Field Weakening Control menu and the 1313/1314 Parameters), complete these initial setup procedures prior to fine-tuning the motor. If using the automated Motor Characterization Procedure, that will come later—after completing steps 1 through 9. Following the motor characterization, then finish this Chapter's Initial Steps prior to completing the fine-tuning in Chapter 9.

PMAC:

For PMAC motor applications, start these initial setup procedures now. The PMAC motor commissioning will be performed after completing steps 1 through 9. Then finish this Chapter's Initial Steps prior to completing the fine-tuning in Chapter 9.

⚠ WARNING

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

Turn on the controller and plug in the 1313 handheld or 1314 PC programmer.

Step 1: Motor Temperature Sensor

Set the Sensor Type parameter to the predefined type (1–9) that corresponds to the motor's temperature sensor. Typically, the motor temperature sensor will be a thermistor that is connected from Analog 2 (pin 8) to I/O Ground (pin 7) as shown in Figure 3.

Quick Links:

Monitor/Inputs Menu [p.77](#)
 Monitor/Motor Menu [p.80](#)
 Temperature Control
 Menu [p.67](#)

To check whether the Sensor Type and Sensor Offset parameter settings and the motor-thermistor connections yield the correct motor temperature, ensure the motor's Sensor Enable parameter is set to On, then read the Temperature value displayed in the programmer's Monitor » Motor menu.

If the 1313/1314 programmer does not display the correct motor temperature, double-check the motor temperature control parameters Sensor Enable, Sensor Type, and Sensor Offset. Ensure that the thermistor's wiring orientation is as shown in Figure 3.

To setup a custom sensor type, VCL and an OS parameter-block change for a Sensor Type "Type 0" is required by the Curtis distributor or support engineer.

To proceed if the correct motor temperature is not displayed, if there is no motor temperature sensor, or while awaiting a custom sensor os/VCL, the initial setup procedure can continue only if the Sensor Enable parameter is set to Off.

If the 1313/1314 displays the correct motor temperature, complete this step by setting the remaining motor temperature control parameters: Sensor Offset (if not already completed, above), Braking Thermal Cutback Enable, Temperature Hot, Temperature Max, and MotorTemp LOS Max Speed.

Step 2: Current Limits

The Drive, Regen, Brake, EMR, and Interlock Current Limit parameters are a percentage of the controller's full rated current. The controller's full rated current is printed on the label of the controller. Set the five current limit parameters to the application's desired values.

Quick Links:

Drivers/Main Contactor menu
 Main Interlock Type [p.52](#)
 Interlock Type [p.53](#)
 Current Limits Menu [p.44](#)
 Battery Menu [p.68-69](#)
 Main Contactor
 Menu [p. 52-53](#)
 EM Brake Control
 Menu [p.50-51](#)

Step 3: Battery

Set the Nominal Voltage parameter to match the nominal battery pack voltage. For Li-ion batteries, follow the manufacturer's settings and the Li-ion BMS recommendations. The controller's BDI is only applicable for lead-acid batteries.

Step 4: Main Contactor

Set up the parameters in the Drivers » Main Contactor Menu.

Step 5: EM Brake

Set up the parameters in the EM Brake Control Menu.

Step 6: Throttle

Before the throttle can be setup, the interlock must be verified as Off. Read the Interlock value displayed in the Monitor » Inputs menu. If the 1313/1314 programmer indicates the interlock is On, review the Interlock Type parameter setting (Drivers » Main Contactor Menu) and turn the interlock off. Verify that the programmer displays that the interlock is now Off.

Contact the Curtis distributor or support engineer to resolve any issues about the interlock before continuing with these initial setup procedures.

Once the interlock has been verified Off, proceed to the throttle input setup. First set the Throttle Type parameter to match the type of throttle (1–5) and wiring used as described in Chapter 3, Throttle section. Then adjust the Forward Deadband, Forward Max, Reverse Deadband and Reverse Max parameters to match the range of the throttle. The Throttle Pot value displayed on the Monitor » Inputs menu is useful when setting up these parameters.

For the forward and reverse directions, read the displayed Throttle Pot voltage at the point when the throttle moves out of neutral and at the point just before full throttle and enter these values for the deadband and max settings for that direction. Set up the remaining parameters in the Throttle Menu as required by the application.

Verify that the throttle settings are correct by checking the Mapped Throttle value displayed in the Monitor » Inputs menu over the entire range of throttle pot movement. The value displayed for Mapped Throttle should be = 0% through the range of throttle motion that is considered neutral. The displayed Mapped Throttle should be = 100% through the range of motion that is considered maximum forward throttle and it is = –100% through the range considered maximum reverse throttle.

Contact the Curtis distributor or support engineer to resolve any issues about the throttle setup before continuing with the initial setup procedures.

Step 7: Brake

If the application does not use the brake function, set the Brake Pedal Enable parameter = Off and Brake Type = 5. Otherwise, configure the Brake parameters.

Before setting the brake parameters, verify that the interlock is Off. Read the Interlock value displayed in the Monitor » Inputs menu. If the 1313/1314 programmer indicates the interlock is On, review the Interlock Type parameter setting (Drivers » Main Contactor Menu) and turn the interlock off.

The Brake Type parameter must be set to match the type of brake throttle (1, 2, 3, 5) and wiring. Next, adjust the Brake Deadband and Brake Max parameters to match the range of the brake pot. The Brake Pot value (Pot2 Raw) displayed in the Monitor » Inputs menu is useful when setting up these parameters. Read the displayed brake pot voltage at the point when the brake moves out of neutral and at the point just before full brake. Enter in these values for the deadband and max parameter settings. Then set up the other parameters in the Brake Menu as required by the applications.

Verify that the brake settings are correct by checking the Mapped Brake value displayed in the Monitor » Inputs menu over the entire range of brake pot movement. The value displayed for Mapped Brake should be = 0% through the range of brake pot motion that is considered neutral. The displayed Mapped Brake should be = 100% through the range of motion that is considered maximum brake.

Contact the Curtis distributor or support engineer to resolve any issues about the brake setup before continuing with the setup procedure.

Quick Links:[Motor Technology](#) p.56[Motor Feedback](#) p.64**Step 8: Motor Technology & Feedback Type**

Set the Motor Technology parameter to match the motor used in the application (0 = ACIM, 1 = PMAC).

Set the Motor Feedback Type parameter based on the Motor Technology setting:

Feedback Type = 1 (quadrature position sensor {encoder}) for ACIM,

Feedback Type = 2 (Sin/Cos sensor) for PMAC.

Make sure that the Feedback Type matches the feedback device used on the motor, and the right type of device for the motor. If there is not a match, stop the setup procedure until a compatible motor and feedback device are obtained and installed.

The ACIM motors use a quadrature encoder. Program the Encoder Steps parameter to the correct setting. This information is typically available from the motor manufacturer. In the ACIM Characterization Procedure (Chapter 8A), it can determine the encoder steps, but only for encoders with 32, 48, 64, or 80 pulses per revolution (ppr).

⚠ WARNING

Setting the Encoder Steps parameter improperly may cause vehicle malfunction, including uncommanded drive.

If the application uses a PMAC motor (which must use a Sin/Cos sensor), the Sin/Cos min and max voltages will be learned during the PMAC Commissioning Procedure (Chapter 8B).

Step 9: Faults (see Chapter 11)

Cycle KSI (to clear any parameter change faults) and use the programmer to check for faults in the controller. All faults and errors (including those in the History file) must be cleared before continuing with the initial setup. Use Chapter 10 for help in troubleshooting. Contact the Curtis customer support engineer to resolve any fault issues before continuing with the setup procedure.

Step 10: Setting Feedback Direction**ACIM:**

Check the encoder while the vehicle drive wheels remain off the ground, no faults present in the controller, the interlock Off (as verified in the Monitor » Inputs menu), and both the throttle and brake in neutral (Mapped Throttle = 0% and Mapped Brake = 0% in the Monitor » Inputs menu).

Use the Monitor » Motor menu to view the Motor RPM display. Turn the motor by hand and observe the sign of Motor RPM. Positive is forward and negative is reverse. If there is a positive Motor RPM when rotating the motor in the forward direction, and a negative Motor RPM when rotating the motor in the reverse direction, the *Swap Feedback Direction* parameter is correct and should not be changed. If the Motor RPM value is negative when rotating the motor forward, the *Swap Feedback Direction* parameter must be changed. Cycle KSI power and repeat the procedure until the *Swap Feedback Direction* setting is correct. If the vehicle will use the emergency reverse feature, the reverse direction (negative Motor RPM) must be correctly selected so that when the Emergency reverse input is active the motor will rotate in the reverse direction.

Contact the Curtis distributor or support engineer to resolve any issues about encoder direction or emergency reverse before continuing with the setup procedure.

PMAC:

Step 10 is not applicable to PMAC motors. Go to the next step for PMAC motor in-vehicle commissioning (Chapter 8B).

Quick Links:[Monitor/Inputs Menu](#) p.77[Monitor/Motor Menu](#) p.80[Swap Feedback Direction](#) p.59

Step 11: Motor Characterization

For an ACIM motor, use either:

Quick Link:

ACIM characterization [p.93–98](#)

(a) Already have software with the motor data programmed to match the motor. The pre-set motor data software is usually the result of contacting the Curtis distributor or support engineer and getting a match between the motor and the Curtis ACIM motor database, or by shipping a motor to Curtis for dyno characterization.

or (b) Use the in-vehicle ACIM Motor Characterization Procedure (Chapter 8A). This procedure will set the direction, may set the Encoder Steps, learn ACIM motor data, and set the FW Base Speed and Field Weakening Drive. Be sure to complete the SlipGain setting for maximum torque and efficiency.

After ensuring that the controller has the motor data values that match the system, complete Steps 12 through 15 of the initial setup.

WARNING

NOTE: Do not take the vehicle down off the blocks if re-running the Motor Characterization procedure.

Step 12: Checking Direction of Rotation

Do not take the vehicle down off the blocks until the motor is responding properly.

Only after the motor is responding properly is it safe to lower the vehicle and place the drive wheels on the ground.

Once the encoder direction set correctly, this step tests to see whether the motor rotation is correct. Begin by cycling KSI off and on (to clear any parameter change faults), then use the 1313/1314 programmer to check for faults in the controller. All faults must be cleared before continuing with the setup procedure. Apply the interlock input and verify that Interlock = On (as verified in the Monitor » Inputs menu). Then, while keeping the brake in neutral*, select a direction and apply throttle. The motor should begin to turn.

For an ACIM motor, observe the motor rotation and if the motor appears to be "fighting itself" (struggling at full current while jerkily turning at very low speed), change the setting of the *Swap Two Phases* parameter.

Contact the Curtis distributor or support engineer if the proper motor direction cannot be resolved.

Step 13: Emergency reverse

Set up the parameters in the Emergency Reverse Menu. Note: Emergency Reverse is active only if using Speed Mode Express or Speed Mode as the motor tuning mode.

Step 14: Interlock Braking

Set up the parameters in the Interlock Braking Menu.

Step 15: As Applicable

If not already completed, within the Speed Controller menu,

15.1 Set the Response Restraint parameters.

15.2 Set the Position Hold parameters.

Set the parameters in the EM Brake Control menu.

15.3 New since os32.0 is the EM Brake Test menu

CAUTION

Note that the interactions of the Position Hold and the EM Brake Control can be complex. Be sure to test all operational modes, including uphill (ramp) and downhill (ramp) stops and starts (re-starts) going in both directions (forward and reverse). Test both loaded and unloaded vehicles, and whether a load is removed or added to a vehicle when stopped on a hill (ramp). Protect personnel and property when testing on a hill (ramp) from unintended (or expected) vehicle movement or roll-away conditions.

Quick Link:

Fig 16 [p.117](#)

**Note: The Throttle Command is zeroed if the Brake Command is > 0 for Control Modes 0 and 1. See Figure 16.*

8A — AUTOMATED ACIM MOTOR CHARACTERIZATION PROCEDURE

⚠ WARNING Motor will rotate during this procedure. Do not take the vehicle down off the blocks.

This procedure is only for applications using an ACIM motor and quadrature encoder.

The procedure is performed in two parts. Both parts must be completed for proper controller setup. The first part applies to all systems, and the second part is different for traction systems and hydraulic systems. It is assumed that Initial Setup steps 1 through 10 have been completed.

The procedure assumes the motor is approximately room temperature (20–25°C); do not characterize a hot motor without a temperature sensor.

If characterizing a traction system, the vehicle should have its drive wheels clearly off ground, and the vehicle should be safely blocked from accidental movement. The drive wheels should be freely spinning—any dragging brake or excessive friction may invalidate this test, or cause it to fail.

If characterizing a hydraulic system, the motor must be unbolted from the hydraulic pump and allowed to spin freely. The test will not work properly if the motor is left connected to the pump. Note: The motor may spin to high speed in either direction.

Hint: The procedure is easier to conduct if the various motor Parameter (Program) and Monitor variables are added to the 1313's Favorites folder. The PC based 1314 programmer can also be used, although it does not offer the convenience of the Favorites folder.

PART 1: TRACTION AND HYDRAULIC SYSTEMS

1. Verify that Motor Technology = 0-ACIM and Feedback Type = 1-encoder. This configuration, and these settings, are required to run this procedure.

Motor » Control Algorithms » 0-ACIM.

Motor » Feedback Options » Feedback Type » 1-Encoder.

2. Enter the number of motor poles in the Motor Poles parameter. The vast majority of induction motors will have 4 poles.

Motor » Control Algorithms » 0-ACIM » Characterization Tests » Motor Poles.

3. Enter the Max Test Speed for the characterization test. Note that this speed might not be achieved, depending on system characteristics; this is normal.

Set this to the application's corresponding Max Speed value (Max_Speed_SpdMx, Max_Speed_SpdM, or Max_Speed_Trq).

Motor » Control Algorithms » 0-ACIM » Characterization Tests » Max Test Speed.

4. Enter the desired Max Test Current for the characterization test. A typical setting is 70% (70% of the maximum motor or controller rating, whichever is higher). Generally this is only reduced if motor heating during the test is a problem, or resonance in the motor occurs at high currents.

Motor » Control Algorithms » 0-ACIM » Characterization Tests » Max Test Current.

5. Using the 1313/1314, clear the Fault History (Faults » Clear Fault History).

6. Ensure that Interlock is enabled (Monitor » Inputs » Interlock).

7. Set Test Enable = 1.

Motor » Control Algorithms » 0-ACIM » Characterization Tests » Test Enable.

8. If any faults are present, stop and resolve the issue before continuing.

9. Note that no cutbacks should be in effect (thermal, voltage, etc.) when these tests are run. Also, the power limiting maps should not cut back current over the characterization speed

range. Check and verify that the power limiting map parameters are set to 100% over the characterization speed range.

Current Limits » Power Limiting Map.

WARNING Motor will start to rotate after next step.

10. Set Test Throttle = +1.

Motor » Control Algorithms » 0-ACIM » Characterization Tests » Test Throttle.

11. After approximately 30 or more seconds the motor will begin to rotate. It is critical to verify the motor is turning in the FORWARD vehicle direction. If it is not, set Test Throttle = 0, wait for the motor to come to a stop, then set Test Throttle = -1.

Notes: (1) This step runs faster if TACT is not running. (2) If the motor starts to accelerate then slows down again, increase **Max Test Current** (IM_AutoChar_Max_Test_Current). See step 4 above.

12. The automated test may take several minutes. When it is complete, the controller will have a Parameter Change Fault. This is normal. Check whether other faults are present. If there is a Characterization Error fault, read the number at Monitor » Controller » Motor Characterization Error and reference the following table. All errors except “1” indicate the motor characterization data is invalid. For error “1,” the data is valid but Encoder Steps must be set manually. Contact the Curtis distributor or support engineer if the Motor Characterization Errors indicated cannot be resolved.

Motor Characterization Errors	
0	Sequencing error. Normally caused by turning off Motor Characterization Test. Enable before running test.
1	Encoder signal seen but step size could not be auto-detected; set up Encoder Steps manually. (Only encoder steps of 32, 48, 64, and 80 will be identified automatically.) <i>Program » Motor » Encoder Steps.</i>
2	Motor temperature sensor fault.
3	Motor temperature > MotorTemp_Hot (150°C is the typical default). This is active only when the motor temp sensor is present and enabled.
4	Controller overtemperature.
5	Controller undertemperature.
6	Battery undervoltage.
7	Battery severe overvoltage.
8	Encoder signal not seen, or one or both channels missing.
9	Motor parameter out of characterization range; check to be sure motor has no load.

13. Cycle KSI power. The motor control should now be operational, though likely poorly optimized. The Slip Gain test is very important to ensure the optimum torque and motor efficiency. The following steps will complete the optimization process.

PART 2A: TRACTION SYSTEMS ONLY

14. Run the **SlipGain** test.

Adjust the SlipGain parameter to provide maximum torque at stall (0 rpm motor speed). Note: The present SlipGain value was determined by the auto-characterization procedure, and is not the default value.

Motor » Control Algorithms » 0-ACIM » Characterization Tests » SlipGain.

To determine the SlipGain setting, use either of the following two methods, with (a) being the most accurate:

14.1 Configure the vehicle to measure stall torque, by using a load cell / draw bar test. Note: Perform this test quickly to avoid excessive motor heating and provide the most accurate results.

Turn off encoder fault detection (Motor » Encoder Fault Setup » Fault Detection Enable).

While applying and holding full throttle, adjust SlipGain until the peak torque is measured.

To have the encoder fault detection active (recommended), turn it back on (now).

14.2 Alternately, the vehicle can be used as the “draw bar test” by accelerating the motor to a predetermined speed (rpm)—with the quickest time corresponding to the optimum SlipGain. This method uses the controller’s time-to-speed function. The test-acceleration will be at full controller current.

Begin by setting the corresponding Accel Rates to very fast values, based upon the application’s Control Mode.

Speed Mode Express: Accel Rate.

Speed Mode: Full Accel LS, Full Accel HS, Low Accel.

Torque Mode: Accel Rate.

On the 1313 programmer, add these items to the Favorites shortcut, to easily toggle between them:

Monitor » Vehicle » Capture Speed 1.

Monitor » Vehicle » Time to Speed 1.

Parameters (Program) » Motor Control Tuning » 0-ACIM » Characterization Tests » SlipGain.

Next, set the **Capture Speed 1** parameter to an rpm enveloping the motor’s desired maximum torque range. The monitor item, **Time to Speed 1**, will record the time taken for the vehicle (motor) to go from zero rpm to the programed Capture Speed 1 during its most recent acceleration.

To perform the test, from a stop, accelerate the vehicle through the predetermined speed. The timer starts counting when full throttle is applied, and stops counting when the speed is reached. A built-in trigger will allow the test to begin again once the vehicle comes to a stop and the throttle is re-engaged, so be sure to note the value of the test first before accelerating-away in subsequent runs. Each time, adjust the SlipGain value until the Time to Speed 1 is minimized.

This test should be run repeatedly over the same stretch of flat or uphill ground. Loading the vehicle will improve results, the idea being to have the motor produce maximum torque as in the draw-bar method. Work quickly, to avoid excessive motor heating, to obtain the most accurate results.

15. Run the **FW Base Speed** test.

Set the FW Base Speed to 6000 rpm. The test should be run with batteries that have a reasonable charge.

Motor » Control Algorithms » 0-ACIM » Field Weakening Control » FW Base Speed.

Set the Accel Rates to be very fast, so that the vehicle will be accelerating at full current during the test. From a stop, quickly apply full throttle to accelerate to a high speed. After the motor settles at a speed for a moment, release the throttle to stop the test run. Note the value of the Base Speed Captured (Monitor » Controller » Motor Tuning » Base Speed Captured), and enter this value for the FW Base Speed setting.

Note:

- The base speed capture test restarts each time the vehicle comes to a stop and the throttle is released, so be sure to note the value before driving (accelerating) away.
- **This FW Base Speed parameter not only needs to be set during initial setup, yet reset each time the Motor Type or the low speed current limit is changed.** For example, if the *Drive_Current_Limit* or *PL_Drive_Nominal* parameters are lowered (changed), retest and reset this parameter.

16. Run the **Field Weakening** test.

This test will enable the choice between a more efficient (less torque) or more torque (less efficient) operation above the base speed, by adjusting the **Field Weakening Drive** parameter.

Note: the present Field Weakening Drive value (0%) was determined by the above auto-characterization routine, and is not the default value (100%) in the native OS intended for the Curtis dynamometer characterized motors when a motor type (number) is utilized. The differences are:

Auto Characterized: Field Weakening Drive is adjusted UP from 0% to the max Field Weakening Drive determined in this text.

Dyno Characterized: Field Weakening Drive is adjusted DOWN from 100% (can be set from 0–100%)

Increasing this parameter toward 100% will progressively allow more torque (along with more current) at high motor speeds, but increasing it too much will actually reduce torque while still driving a great deal of current. The auto-characterization value of 0% will give the most efficient motor operation, but will not give the highest torque at high speeds.

If more torque is required at high speeds, run this Field Weakening Drive test.

On the 1313 programmer, add these two items to the Favorites shortcut, along with Field Weakening Drive parameter, to easily toggle between them:

Parameters (Program) » Motor » Control Algorithms » 0-ACIM » Field Weakening Control » Field Weakening Drive.

Parameters (Program) » Vehicle » Capture Speed 1, 2.

Monitor » Vehicle » Time Between Speeds.

Set **Capture Speed 1 and 2** to values that are close to the Max Speed setting. The Capture Speed 1 and 2 values **MUST** be set higher than the measured FW Base Speed in step 15 (because Field Weakening only affects performance at speeds above the “base speed”). Ideally the Capture Speed 1, 2 values should be above the second corner point of the motor torque – speed curve. For example, if the Max Speed is set to 4000 rpm, set Capture Speed 1 = 3500 rpm and Capture Speed 2 = 3800 rpm:

Parameters (Program) » Speed Mode Express » Max Speed.

Parameters (Program) » Speed Mode » Speed Controller » Max Speed.

Parameters (Program) » Torque Mode » Speed Limiter » Max Speed.

From a stop, apply full throttle to accelerate to a speed greater than Capture Speed 2. Note the value displayed in Monitor » Vehicle » Time Between Speeds.

Now increase the Field Weakening Drive setting and repeat the acceleration in the same direction, with the same load, and with the same full throttle and again note the Time Between Speeds value.

Keep adjusting the Field Weakening Drive value and repeating the test until finding the Field Weakening Drive value that results in the smallest Time Between Speeds. This Field Weakening Drive value should be used for applications that require the most torque at high speeds.

Some Field Weakening Drive settings may result in a test acceleration run where Capture Speed 2 cannot be obtained. These Field Weakening Drive settings cannot provide enough torque to get the vehicle past Capture Speed 2, and thus can be ignored.

Use the Field Weakening Drive setting that best matches the application. It will be somewhere between 0% (most efficient, least motor heating) and the setting found in this test (highest torque at high speeds).

NOTE: The Field Weakening Drive setting will have no effect at motor speeds below the FW Base Speed.

17. Return to Chapter 6 and complete Initial Setup Steps 12 through 15.

Quick Links:

Max Speed

Express Mode [p.33](#)

Speed Mode [p.34](#)

Torque Mode [p.40](#)

Capture Speed 1,2 [p.71](#)

Field Weakening Drive [p.58](#)

Time Between Speeds [p.84](#)

Quick Links:[SlipGain](#) [p.57](#)[Max Speed](#)[Express Mode](#) [p.33](#)[Speed Mode](#) [p.34](#)[Current \(RMS\)](#) [p.81](#)**PART 2B: HYDRAULIC SYSTEMS ONLY****14. Run the SlipGain test.**

Adjust the **SlipGain** parameter to provide maximum torque at low speeds. Note: The present SlipGain value was determined by the auto-characterization procedure, and is not the default value.

On the 1313 programmer, add the motor current to the items in the Favorites shortcut, to easily toggle between them:

Monitor » Controller » Current (RMS).

Parameters (Program) » Motor Control Tuning » 0-ACIM » Characterization Tests » SlipGain.

Set Max Speed to a fairly low speed (but high enough to be able to lift a heavy load on the forks), typically 800 rpm. The Max Speed setting must be below the motor base speed, which is measured in the next step.

Put a very heavy load on the forks, preferable a max load.

Apply full throttle, and adjust the SlipGain parameter until the motor current is minimized. If the vehicle has a multistage mast, work in the first stage only so that the load is consistent.

This will be the SlipGain parameter value, unless following step 15, the base speed used here (e.g., 800 rpm) was too high.

15. Run the FW Base Speed test.

Note: For this test, because the hydraulic motor (disconnected from the pump) can quickly accelerate past the “Base Speed”, it will be necessary to pre-flux the motor current prior to running this test to get an accurate measure of base speed. Since the controller will be running current with no throttle, ensure this test’s 1313 Favorites items are loaded so the test continues without delay to prevent motor and controller heating. Then, be sure to re-set the controller to cancel the pre-flux following the test. This test is for hydraulic (pump) motor working in Speed Mode only. Run this test with batteries that have a reasonable charge.

Set in the 1313 Favorites

Parameters (program) » Motor » Control Algorithms » 0-ACIM » Field Weakening Control » FW Base Speed.

Parameters (Program) » Motor » Control Algorithms » 0-ACIM » Field Weakening Control » Min Field Current.

Monitor » Controller » Motor Tuning » Test Field Current.

Monitor » Controller » Motor Tuning » Base Speed Captured.

Set the Max Speed to 6000 rpm (i.e., beyond the maximum desired pump speed).

Set the Accel rates to be very fast, so the motor will be accelerating at full current during the test.

Record the value of the Test Field Current (Monitor) and enter this value for the Min Field Current parameter.

⚠ CAUTION

The controller is now running current with no throttle, so continue this test without delay to prevent motor and controller heating.

From a stop, quickly apply full throttle to accelerate to a high speed. After the motor settles at a speed for a moment, release the throttle to stop the test. Note the value of the Base Speed Captured and enter this value for the FW Base Speed parameter setting.

⚠ WARNING

When this test is finished, re-set the parameter Min Field Current = 0 Amps to cancel the pre-flux current. Verify that the pre-flux current is canceled before continuing.

Quick Links:[FW Base Speed](#) [p.58](#)[Express Mode](#) [p.33](#)[Accel Rate](#)[Speed Mode](#) [p.36](#)[Full Accel HS](#)[Full Accel LS](#)[Low Accel Rate](#)[Test Field Current](#) [p.83](#)[Min Field Current](#) [p.59](#)[Base Speed Captured](#) [p.83](#)

16. Run the Field Weakening test.

This test will enable the choice between a more efficient (less torque) or more torque (less efficient) operation at high speeds, by adjusting the hydraulic motor's Field Weakening Drive parameter. While the most efficient operation will occur with field weakening at the 0% auto-characterization value, more torque at high speeds may be required, necessitating adjusting the field weakening. Field Weakening Drive should be set just high enough to meet the high speed performance specifications, typically maximum lift speed with full load.

Note: the present Field Weakening Drive value (0%) was determined by the above auto-characterization routine, and is not the default value (100%) in the native OS intended for the Curtis dynamometer characterized motors when a motor type (number) is utilized. The differences are:

Auto Characterized: Field Weakening Drive is adjusted UP from 0% to the max Field Weakening Drive determined in this test.

Dyno Characterized: Field Weakening Drive is adjusted DOWN from 100% (can be set from 0–100%).

Increasing the Field Weakening Drive parameter toward 100% will progressively allow more torque (along with more current) at high motor speeds, but increasing it too much will actually reduce performance while still driving a great deal of current.

On the 1313 programmer, add these two items to the Favorites shortcut to easily toggle between them:

Monitor » Controller » Motor Tuning » Motor RPM.

Parameter (Program) » Motor » Control Algorithms » 0-ACIM » Field Weakening Control » Field Weakening Drive.

Run this test with no load on the forks. Work with only the first stage of a multistage mast. Work quickly to avoid overheating the controller or motor.

Using the application's Speed (control) Mode, set the Max Speed parameter to a high speed that likely cannot be achieved by the hydraulic system.

Set Field Weakening Drive = 100% (note: this test work works DOWN from 100% to find the actual maximum value).

Apply full throttle, running the motor towards the maximum speed. Record the maximum (achieved) Motor RPM.

Repeat the acceleration runs while continuing to lower the Field Weakening Drive value before on each run until the achievable maximum motor speed fall off significantly (20–30 rpm). This number represents the **maximum** Field Weakening Drive value (%) to achieve the maximum torque/speed envelope of the hydraulic motor (do not use a higher percentage). The Field Weakening Drive percentage can always be lowered to reduce energy consumption and improve efficiency (but reducing torque output) at high speeds.

NOTE: The Field Weakening Drive setting will have no effect at motor speeds below FW Base Speed.

17. Return to Chapter 6 and complete Initial Setup steps 12 through 15.

Quick Links:

Motor RPM [p.83](#)

Field Weakening Drive [p.58](#)

Max Speed

Express Mode [p.33](#)

Speed Mode [p.34](#)

8B — AUTOMATED PMAC MOTOR COMMISSIONING PROCEDURE

PMAC CONSIDERATIONS

NOTICE

PMAC applications are only compatible with controllers that are manufactured with a cross-connect diode. These controllers are models 123XE-xx7x and 123XSE-xx7x. All the 1239E models are PMAC compatible when operated as described below. The cross-connect diode allows the logic section of the controller to power up if the capacitor voltage is above the brownout voltage regardless of whether the KSI pin is powered. Since a field weakened IPM motor can generate voltages significantly above the ratings of the motor, the controller must be able to control the main power bridge if the motor is generating significant voltages in order to protect itself from these damaging voltages. If a controller without a cross connect diode is configured for PMAC operation, a non-clearable Parameter Mismatch Fault will be declared.

Supported 1232E, 1232SE, 1234SE, 1236SE and 1238SE models

The following newly introduced 123X-xx7x models are manufactured with a cross-connect diode and calibrated for operation with OS34+:

- 1232E-2171, 1232E-2371, 1232E-5171, 1232E-5271, 1232E-6271, 1232E-7271.
- 1232SE-2471, 1232SE-5371, 1232SE-6371.
- 1234SE-4571, 1234SE-5471, 1234SE-6371.
- 1236SE-4671, 1236SE-5671, 1236SE-6571.
- 1238SE-4971, 1238SE-5971, 1238SE-6971, 1238SE-7971.

The 1234E, 1236E and 1238E controllers are not available with cross-connect diodes.

1239E PMAC Support (See manual 53095)

All existing 1239E models and the 1269E support PMAC operation with OS 37.0, but the OEM must insure that KSI power is active any time the motor is moving. Removal of KSI power when the motor is spinning past the base speed may result in immediate controller damage.

Use this Chapter when commissioning a PMAC application after first completing the initial setup steps.

WARNING

The PMAC motor will rotate during this procedure. Do not take the vehicle down off the blocks. This procedure is only for applications using a PMAC motor and Sin/Cos sensor.

Perform this procedure in one stage. It takes approximately 5–10 minutes. It can only be used on a sinusoidal wound PMAC motor with a Sin/Cos absolute position sensor (e.g. RLS RM22, SKF BX-BMB-7050, etc.) set up with one sensor revolution per mechanical revolution.

This procedure's assumption is that Initial Setup steps 1 through 9 have been completed and that the motor is approximately room temperature (20–25°C); **do not characterize a hot motor without a temperature sensor.**

If commissioning a traction system, the vehicle should have its drive wheels clearly off ground, and the vehicle should be safely blocked from accidental movement. The drive wheels should be freely spinning—any dragging brake or excessive friction may invalidate this test, or cause it to fail.

If commissioning a hydraulic system, the motor must be unbolted from the hydraulic pump and allowed to spin freely. The test will not work properly if the motor is left connected to the pump. Note: The motor may spin to Max Test Speed in both directions.

If using the 1314, enable Include monitor values for logging:

Options » Settings » Include monitor values.

1. Verify that Motor Technology = 1-PMAC and Feedback Type = 2-Sin/Cos sensor. This configuration, and these settings, are required to run this procedure.

Motor » Motor Technology » 1-PMAC.

Motor » Feedback Options » Feedback Type » 2-Sin/Cos.

2. Enter the maximum speed the motor is intended to turn in the application as Typical Max Speed.

Motor » 1-PMAC » Commissioning Tests » Typical Max Speed.

3. Enter the maximum speed the motor is intended to turn as the Max Test Speed for the commissioning test. Note this speed might not be achieved, depending on system characteristics; this is normal. A typical setting is 5000 rpm, but clearly the Max Test Speed should not be set to a higher speed than safe for the mechanical configuration.

Motor » 1-PMAC » Commissioning Tests » Max Test Speed.

4. Enter the correct PMAC Motor Type before beginning commissioning test. Contact Curtis if there is a question as to the application's motor type.

Motor » 1-PMAC » Commissioning Tests » Motor Type.

Motor » 1-PMAC » Commissioning Tests » Motor Poles.

5. Enter the desired Max Test Current for the commissioning test. A typical setting is 20% (note this is 20% of the maximum controller rating). Generally this is only reduced if motor heating during the test is a problem. Generally, this is only increased if the motor fails to rotate during commissioning.

Motor » 1-PMAC » Commissioning Tests » Max Test Current.

6. Using the 1313/1314, clear Fault History (*Faults » Clear Fault History*).

7. Turn on (enable) Interlock (Monitor » Inputs » Interlock).

8. Enable the PMAC Commissioning by setting **Test Enable = 1**.

Motor » 1-PMAC » Commissioning Tests » Test Enable.

9. If any faults are present, except *PMAC Commissioning Needed*, stop and resolve the issue before continuing.

10. Set **Test Throttle = +1**.

Motor » 1-PMAC » Commissioning Tests » Test Throttle.

WARNING

WARNING – Motor will start to rotate in step 11 and faster in step 12.

11. The controller will start to inject test pulses. After a few minutes (when pulsing stops), the motor will begin to rotate very slowly. It is critical to verify the motor is turning in the FORWARD vehicle direction. If it is not, set Test Throttle = 0, wait for the motor to come to a stop, then set Test Throttle = –1. The motor will start again in the same direction. The controller will properly set the direction at the end of commissioning process. (Note: This is a different operation than that in the ACIM characterization procedure, which does turn the motor in the other direction when Test Throttle is set negative).

12. After about a minute, the motor will spin up to a moderate speed, then to high speed (limited by Max Test Speed) in each direction.
13. When the automated test is complete, the controller will have a Parameter Change Fault. This is normal. Check whether other faults are present. If there is a Characterization Error fault, read the number at Monitor » Controller » Motor Characterization Error and reference the following table. An error during the PMAC commissioning test indicates the motor commissioning data is invalid; contact the Curtis distributor or support engineer for assistance if the indicated error cannot be resolved.

Motor Characterization Errors	
0	Sequencing error. Normally caused by turning off Motor Commissioning Test. Enable before running test.
2	Motor temperature sensor fault.
3	Motor overtemperature > Motor Temperature Hot parameter (active only when temp sensor is present).
4	Controller overtemperature.
5	Controller undertemperature.
6	Battery overvoltage.
7	Battery severe overvoltage.
9	Motor parameter out of range; check to be sure motor has no load.
10	Motor Poles not set (set the correct PMAC Motor Type).
20	Sin/Cos sensor not found.
21	Phasing not detected (check to be sure the motor has no load, and Motor Type is setup first, as this sets Motor Poles automatically).
22	Sin/Cos sensor characterization failure. (as above, check Motor Type is setup first, as this sets Motor Poles automatically).
23	Started motor commissioning while motor rotating.
24	Sin/Cos signal too noisy for characterization.

14. When the test is complete, cycle KSI to clear the Parameter Change fault. Then attempt to drive the motor in the forward direction. If the motor turns in the reverse direction, invert the Swap Direction parameter in the Commissioning Tests menu (there is no swap two phases parameter like the induction motor control – swap direction takes care of both).

Motor » 1-PMAC » Commissioning Tests » Swap Direction.

15. Return to Chapter 7 and complete Initial Setup Steps 12 through 15.

9 – TUNING GUIDE

Many aspects of vehicle performance can be optimized, using the wide variety of adjustable parameters available to these E and SE controllers. Once a vehicle/motor/controller combination has been tuned, the parameter values can be made standard for the system or vehicle model. Any changes in the motor, the vehicle drive system, or the controller will require that the system be tuned again to provide optimum performance.

Selecting the Control Mode

Before starting to tune the vehicle's performance, select which control mode to use. Set the Control Mode Select parameter = 0 (Speed Mode Express for either traction or pump motors), = 1 (Speed Mode, for either traction or pump motors) or = 2 (Torque Mode, for traction motors). Cycle KSI input Off and then On (to clear any parameter change faults) and use the 1313 handheld or 1314 PC programmer to check for faults in the controller. Then proceed to the tuning steps for the control mode selected.

Conduct the steps in the sequence given, because successive steps build upon the ones before. It is important that the effect of these programmable parameters be understood in order to take full advantage of the controller's powerful features. Please refer to the descriptions of the applicable parameters in Chapter 4 if there is any question about what any of them do.

0 – Speed Mode Express Tuning

Speed Mode Express is the same as Speed Mode with the exception that it has fewer parameters and is therefore simpler to use. Most vehicle applications will find success with Speed Mode Express; however, for some applications vehicle performance cannot be satisfactorily fine-tuned in Speed Mode Express. In this case, change the control mode to Speed Mode (i.e., set Control Mode Select = 1).

- 0.1 For traction applications, set Pump Enable to Off. For AC pump applications, set this parameter On.
- 0.2 Adjust Max Speed to the maximum speed the motor should turn in the vehicle application; this speed setting corresponds to an input of full throttle.
- 0.3 Adjust Typical Max Speed to the approximate maximum speed that the motor will spin. This is usually the same value as the setting for Max Speed, but some applications have a Max_Speed_SpdMx that changes (in the VCL software). If the Max_Speed_SpdMx changes, set Typical Max Speed to the highest speed the motor is expected to reach. This value does not need to be set precisely since it will not change motor performance. Typical Max Speed sets a reference point for the "rate" parameters (accel, decel, brake rates), so that applications that have a changing Max_Speed_SpdMx will not experience changes in the rates (because the rates are referenced to the unchanging Typical Max Speed value). Once the Typical Max Speed parameter is set, do not readjust it without adjusting all the rate parameters as well.
- 0.4 Kp and Ki typically do not need to be changed as the default values will work well in most applications. If choosing to adjust Kp (for looser or tighter following of the speed trajectory set by the accel, decel, and brake rates), follow the procedure in step "d" in the Speed Mode tuning section.
- 0.5 Adjust the Accel Rate and Decel Rate as necessary while moving the throttle to different positions (i.e., neutral to full throttle, half throttle to full throttle, full throttle to half throttle, full throttle to neutral, neutral to low throttle, etc.). For AC pump applications, typically the Accel Rate and Decel Rate are set very fast for quick response to hydraulic inputs.

Quick Links:

Control Mode select [p.32](#)
 Speed Mode Express menu [p.33](#)
 Max Speed
 Kp
 Ki
 Accel Rate
 Decel Rate
 Brake Rate
 Typical Max Speed [p.56](#)

⚠ WARNING

- 0.6 Adjust the Brake Rate as necessary while reversing the throttle input (i.e., full throttle forward to low throttle reverse, full throttle forward to full throttle reverse, full throttle reverse to low throttle forward, etc.). If a brake input is present in the application (Brake Pedal Enable = On) continue adjusting Brake Rate by applying different amounts of brake throttle (i.e., full throttle forward, then apply full brake or full throttle forward, then apply low brake, etc.). For AC pump applications the Brake Rate has no effect on performance since the AC pump will not run in reverse nor with a brake input.

AC pump motor applications, which in step “a” had the Pump Enable parameter set to On, may also be controlling hydraulic valves (load hold valve, proportional valve). The Proportional Driver parameters and some VCL will typically be used to set up the control of these valve driver outputs. Consult with the Curtis distributor or support engineer regarding hydraulic valve control.

1 – Speed Mode Tuning

- 1.1 For traction applications, set Pump Enable to Off. For AC pump applications set this parameter On.
- 1.2 Adjust Max Speed to the maximum speed the motor should turn in the vehicle application; this speed setting corresponds to an input of full throttle.
- 1.3 Adjust the Typical Max Speed to the approximate maximum speed that the motor will spin. This is usually the same value as the setting for Max Speed, but some applications have a Max_Speed_SpdM that changes (in the VCL software). If the Max_Speed_SpdM changes, set the Typical Max Speed to the highest speed the motor is expected to reach. This value does not need to be set precisely since it will not change motor performance. Typical Max Speed sets a reference point for the “rate” parameters (accel, decel, brake rates), so that applications that have a changing Max_Speed_SpdM will not experience changes in the rates (because the rates are referenced to the unchanging Typical Max Speed value). Once the Typical Max Speed parameter is set, do not readjust it without adjusting all the rate parameters as well.
- 1.4 Kp and Ki (LS & HS) typically do not need to be changed as the default values work well in most applications. These parameters control how tightly the actual motor speed will track the requested speed trajectory (speed trajectory is set by the Accel, Decel, and Brake rates).

To adjust the Kp (for looser or tighter following of the speed trajectory), follow these guidelines.

- Set the following parameters. Before setting them, make a note of their present (default) settings so they can be returned to the original values at the end of this procedure.
 - In the Speed Mode » Speed Controller menu, set the **Max Speed** to low value (≈1000 rpm), as high speed operation is not needed to observe system response.
 - In the Speed Mode » Response menu, set all the **Accel and Decel** rates to their fastest values (0.1 seconds); this allows better observation of the system response.
 - In the Speed Mode » Restraint menu, set the **Soft Stop Speed** parameter = 0 rpm to disable the soft stop speed function.
 - In Speed Mode » Restraint » Position Hold Menu, set **Position Hold Enable** = Off such that the position hold function will not interfere with the speed control gain setup procedure.
 - *If the vehicle has an EM Brake*, in the EM Brake menu, set the **EM Brake Type** = 1. This setting will release the EM Brake as soon as interlock is asserted.

Quick Links:

Control Mode select [p.32](#)
 Speed Mode menu [p.34](#)
 Speed Controller menu [p.34](#)
 Max Speed
 Kp
 Ki (LS/HS)
 Response menu [p.36](#)
 Accel Rate
 Decel Rate
 Brake Rate
 Restraint menu [p.38](#)
 Soft Stop Speed
 Position Hold
 menu [p.39](#)
 EM Brake Control menu [p.50](#)
 Typical Max Speed [p.56](#)

- Cycle KSI to clear any faults. Then, assert (close) the Interlock.
- Using very quick, pulsing throttle movements, increase the throttle and then release it to 0%. The intent is to give the speed controller torque impulses.
- Increase Kp and repeat the throttle tests. Increase Kp until noticing marginal stability (normally motor bouncing, or continuous oscillation in the gears, is heard). Note: It is possible that very heavy vehicles will not experience marginal stability even at the highest setting of Kp.
- Once the Kp setting for marginal stability is found, reduce the Kp value by about one third (i.e., final Kp = marginal stability Kp * 2/3).
- If using Speed Mode Express, enter this Kp value for the Kp parameter in the Speed Mode Express menu.
- Set the Max Speed, Accel/Decel, Soft Stop Speed, Position Hold, and Brake Type parameters back to their original values.

Quick Links:[Speed Controller menu p.34](#)[Max Speed](#)[Kp](#)[Ki \(LS/HS\)](#)[Response menu p.36](#)[Accel Rate](#)[Decel Rate](#)[Brake Rate](#)[Restraint menu p.38](#)[Soft Stop Speed](#)[Position Hold menu p.39](#)

1.5 In the Speed Mode » Response menu, adjust the five Accel and Decel Rate parameters as necessary while moving the throttle to different positions (i.e., neutral to full throttle, half throttle to full throttle, full throttle to half throttle, full throttle to neutral, neutral to low throttle, etc.). For AC pump applications, these parameters are typically set very fast for quick response to hydraulic inputs.

1.6 In the Speed Mode » Response menu, adjust the remaining three brake rate parameters as necessary while reversing the throttle input (i.e., full throttle forward to low throttle reverse, full throttle forward to full throttle reverse, full throttle reverse to low throttle forward, etc.).

If a brake input is present in the application (Brake Pedal Enable = On) continue adjusting these three brake rates by applying different amounts of brake throttle (i.e., full throttle forward, then apply full brake or full throttle forward, then apply low brake, etc.).

For AC pump applications these parameters have no effect on performance since the AC pump will not run in reverse nor with a brake input.

1.7 The parameters in the Speed Mode » Response » Fine Tuning menu typically do not need to be changed as the default values work well in most applications.

AC pump motor applications, which in step “1.1” had the Pump Enable parameter set to On, may also be controlling hydraulic valves (load hold valve, proportional valve). The Proportional Driver parameters and some VCL will typically be used to set up the control of these valve driver outputs. Consult with the Curtis distributor or support engineer regarding hydraulic valve control.

WARNING

2 – Torque Mode Tuning

- 2.1 Set Max Speed to the application's maximum motor speed (rpm).
- 2.2 Kp, Ki, and Kd typically do not need to be changed as the default values will work well in most applications. These parameters control how tightly the controller limits the speed of the motor to the programmed Max Speed.
- 2.3 Set Typical Max Speed to the expected maximum speed of the motor.
- 2.4 Adjust the parameters shown in Figure 9 to set up the throttle mapping:
 - Drive Current Limit (Current Limits menu).
 - Regen Current Limit (Current Limits menu).
 - Forward/Back Full Restraint Speed (Torque Mode » Response menu).
 - Brake Taper Speed (Torque Mode » Response » Fine Tuning menu).
 - Neutral Braking (Torque Mode » Response menu).
 - Neutral Taper Speed (Torque Mode » Response menu).
 - Creep Torque (Torque Mode » Response » Fine Tuning menu).
- 2.5 In the Torque Mode » Response menu, adjust the four accel, decel, and release rate parameters as necessary while moving the throttle to different positions (i.e., neutral to full throttle, half throttle to full throttle, full throttle to half throttle, full throttle to neutral, neutral to low throttle, etc.).
- 2.6 The other parameters in the Torque Mode » Response » Fine Tuning menu typically may need to be changed for some applications. Read the parameter descriptions and adjust as necessary.

Quick Links:

Torque Mode/Speed

Limiter [p.40](#)

Kp

Ki

Kd

Response menu [p.41](#)

Fine Tuning [p.42](#)

Figure 9 [p.43](#)

Current Limits menu [p.44](#)

Typical Max Speed [p.56](#)

10 — VEHICLE CONTROL LANGUAGE (VCL)

Curtis 1232E/34E/36E/38E and 1232SE/34SE/36SE/38SE controllers have a built-in programmable logic controller with application-specific functions. VCL (Vehicle Control Language) software provides a way to implement unique and complex vehicle control functions.

VCL is a simple programming language that will feel very familiar to anyone who has worked with BASIC, Pascal, or C. Working with VCL requires the installation of the WinVCL program onto a PC. WinVCL will compile VCL programs and flash download the software into the controller through the computer's serial port. The install process for WinVCL will also install three important manuals on the PC: the WinVCL User's Guide, VCL Programmer's Guide, and VCL Common Functions Manual. These manuals, which are in PDF format, include more detailed information about VCL than is included here.

This chapter of the manual summarizes VCL and also describes aspects and functions of VCL that are specific to these E and SE controllers. For a more complete understanding of the functions and capabilities of VCL, see the WinVCL User's Guide, VCL Programmer's Guide, and VCL Common Functions Manual.

Summary of VCL Basics

- VCL is not case-sensitive:
put_pwm(), Put_PWM(), and PUT_PWM() are identical.
- Spaces in variable names are not allowed in VCL; use underscores in place of spaces.
Example: Forward_Offset is the VCL name for the 1313/1314 parameter
Forward Offset.
- Functions are followed by parentheses; for example:
Reset_Controller() is a function.
Reset_Voltage is a variable.
- Logical statements must be inside parentheses; examples:
IF (setpoint >50)
ELSE IF ((setpoint <20) & (temperature >100)).
- Comments are preceded by semicolons.

The VCL functions described in the VCL Common Functions Manual are available to all 1232E/SE, 1234E/SE, 1236E/SE, and 1238E/SE controllers. These controllers have these additional specific functions, listed below. All VCL functions, common and specific, are also available in the controller's os37 SysInfo file.

ENABLE_PRECHARGE()	p. 126
DISABLE_PRECHARGE()	p. 127
SET_DIGOUT()	p. 128
CLEAR_DIGOUT()	p. 128
ENABLE_EMER_REV()	p. 129
DISABLE_EMER_REV()	p. 129
SET_INTERLOCK()	p. 130
CLEAR_INTERLOCK()	p. 130
SETUP_POT()	p. 131
GET_POT()	p. 131
SETUP_POT_FAULTS()	p. 132
START_PUMP()	p. 133
STOP_PUMP()	p. 133
AUTOMATE_FREQUENCY_OUTPUT()	p. 134
OVERRIDE_EM_BRAKE_PWM()	p. 135
RESTORE_EM_BRAKE_PWM()	p. 135
MAP_TWO_POINTS()	p. 136
BATTERY_COMPENSATE()	p. 137
PUT_MESSAGE_TO_STRING()	p. 137
AUTOMATE_PWM_INPUT()	p. 139
SETUP_TIMER_LONG()	p. 139
SETUP_MAILBOX_MASK()	p. 140
SETUP_MAILBOX_EXTENDED_MASK()	p. 140
AUTOMATE_ENCODER()	p. 141

VARIABLE TYPES

VCL provides dedicated space in which to store custom variables. There are four types of variables, based on their type of storage: volatile storage (RAM) and three types of non-volatile storage (EEPROM) are available.

RAM variables are stored only while power is on; they are lost at power-down. They must be initialized on power-up by explicit VCL assignments (i.e., User1 = 12).

NVUser1–15 EEPROM variables are 15 variables automatically saved at power-down and restored at power-on. During power-down or a power-failure (brownout), these variables are saved to NVM1. During operation, these variables are also periodically saved (every 6 minutes) to NVM2, thus two locations are used to preserve the data. At power-up, the data in NVM1 is read first, yet if it is corrupted, the data from NVM2 is restored. See the section on non-volatile memory access in the VCL Common Functions manual for more information, including caution to changing the NVM2 save rate, and rational of the save/restore strategy.

Block EEPROM are 38 blocks of 15 variables (total of 570 variables), which are stored and recalled using the functions NVM_Block_Read and NVM_Block_Write. The 38 blocks are called NVM3–NVM40. The read and write functions must point to the RAM variables that the EEPROM blocks should be written from or read to. For example, NVM_Block_Read(NVM10,0,15,User20) will read the 15 variables stored in EEPROM block NVM10 and restore those variables to the 15 variables starting with RAM variable User20 (so the 15 EEPROM variables would be restored to User20–34). See the section on non-volatile memory access in the VCL Common Functions manual for more information.

Parameters EEPROM variables are a special type of EEPROM variable that is intended to be used to create OEM defined 1313/1314 programmer parameters. These parameters can be defined as 16-bit by using the P_User variables or they can be defined as bit (On/Off) by using the P_User_Bit variables. These variables are typically written to EEPROM through the 1313 handheld or 1314 PC programmer interface (i.e., when a user changes a parameter setting using the 1313). They can be used in the VCL code, but changing a P_User (or P_User_Bit) value with VCL will only change the variable value in RAM and will not change the value in EEPROM. Thus, these variables are intended for creating and defining 1313/1314 programmable parameters only.

VCL can modify the control mode parameters in RAM by using the VCL variable name for the programmable parameter. For example,

```
Brake_Rate_SpdMx = 3000 ;Change Brake Rate to 3.0sec
```

will change the RAM value of the Speed Mode Express's Brake Rate; the new value will be used in determining the braking rate. However, the value of the parameter's stored EE value remains unchanged; when the controller is turned off, the RAM value will be lost. The next time the controller is powered back on, the "old" value of Brake Rate will be restored from EE memory. To save this RAM value in VCL, use the NVM_Write_Parameter function ((i.e., NVM_Write_Parameter(Brake_Rate_SpdMx)).

Parameter values that are changed by using the 1313 handheld or 1314 PC programmer are saved directly to EE memory. The 1313/1314 changes will be retained and restored the next time the controller is powered back on.

The table below summaries the available VCL variables.

Type	Quantity	Range
RAM	516 variables	User1 – User200 User_Long1 – User_Long16 AutoUser1 – AutoUser300
NVUser EEPROM	15 variables	NVUser1 – NVUser15
Block EEPROM	38 blocks (15 variable each)	NVM3 – NVM40
Parameters EEPROM	150 variables and 10 variables of 8 bits each (80 bits)	P_User1 – P_User150 P_User_Bit1 – P_User_Bit10

VCL RUNTIME RATES

VCL is an interpreted language. Each line of VCL code is converted (compiled) into a set of codes and then flash loaded into the controller. The controller interprets these codes one line at a time while the system is powered up. Here are the processing rates of the various functions:

Item	Function/Item Description	Quantity	Service Rate
ABS	Absolute Value	2	4 ms
ADC	Analog to Digital Converter Input (Analog1 and Analog2)	2	1 ms
CAN	CAN Communications	24	4 ms
CPY	Copy	8	4 ms
DLY	Delay	32	1 ms
FLT	Filter	4	1 ms
LIM	Limit	4	4 ms
MAP	Map	8	4 ms
MTD	Multiply then Divide	4	4 ms
NVM	Non-Volatile Memory block (NVM3-40)	38	2 ms
PID	Proportional Integral Derivative	2	4 ms
POT	Potentiometer Input	2	8 ms
PWM	Pulse Width Modulated output	6	4 ms
RMP	Ramp	4	1 ms
SCL	Scaling	8	4 ms
SEL	Selector, 2-position switch	8	4 ms
SEL_4P	Selector, 4-position switch	8	32 ms
SW	Switch Input	1*	4 ms
TMR	Timers (hourmeters)	3	1 ms

* There is only one Switch variable; it has 16 associated bit-variables.

OS 36.0 includes 32 CAN Communications mailboxes running at a 4 ms service rate where OS 31.0 included 24 mailboxes.

I/O CONTROL WITH VCL

Digital Inputs

Quick Link:

Fig 3 [p.12](#)

The controllers each have 16 digital inputs. Nine are switch inputs (Sw_1 through Sw_8, and Sw_16). These switch inputs are shown on the standard wiring diagram (Figure 3). The remaining seven digital inputs are less obvious: one on each driver and digital output (Sw_9 through Sw_15). These can be used as digital inputs or to sense the state of the output or its wiring (e.g., open coil check).

To address a digital input in a VCL program, use the desired input label (Sw_1 through Sw_16). Use On or Off in the code when determining a switch state; using true/false or 1/0 will give erroneous results.

```
if (Sw_1 = ON)
{
;put code here to run when switch 1 is On
}
if (Sw_16 = OFF)
{
;put code here to run when switch 16 is Off
}
```

All switch inputs are automatically debounced by the VCL operating system. This prevents noisy contacts or contact bounce from causing erroneous events in the application's VCL code. The debounce time can be varied from 0 to 32 ms in 4 ms steps, using this function:

```
Setup_Switches(5); 20 milliseconds
```

If this line is not in the VCL code, the debounce time is set at 16 ms.

The VCL function *Automate_PWM_Input()* can measure the duty cycle of a PWM signal applied to the switch (SW1 – SW16) inputs.

Driver and Digital Outputs

Quick Link:

*Digital and PWM
Outputs* [p.15](#)

There are five driver outputs (PWM1 through PWM5) and two digital outputs (DigOut6 and DigOut7). These outputs have variations in current and frequency range. For their specifications, see Digital and PWM Outputs in Chapter 2.

Driver outputs have high current FET output stages and can be pulse width modulated (PWM) to vary the average output to inductive loads such as contactors and relays. This is useful when the battery voltage needs to be brought down for lower voltage coils. The two digital outputs are 1 A drivers that are only On or Off.

Drivers use a special VCL function to set their PWM level. This PWM level can be set up in a signal chain to update automatically or can be set directly in the main loop. PWM can be set from 0–100% using the digital range of 0 to 32767.

```
Put_PWM(PWM2,16384)
```

will output a 50% waveform on Driver 2.

```
Automate_PWM(PWM2,@user1)
```

will continually update the Driver 2 output with the present value of variable User1. This automate statement needs only to be run once, usually in the initialization section of the VCL program. VCL can monitor the present value of a PWM driver: the variable PWMx_Output (where “x” is the PWM channel number) is automatically filled with the present value of the driver output.

The proportional driver (Driver 5 , Pin 2) is different from Drivers 1–4. It can be controlled in two ways: with the proportional driver processing function (see Figure 17) or with the VCL Put_PWM() function. The VCL statement Put_PWM(PWM5, 16383) will result in a 50% PWM output on pin 2 only if the parameter PD Enable is set to Off. For more information about the proportional driver, see the section entitled “INTERFACING THE PROPORTIONAL CURRENT DRIVER” later in this Chapter.

Control of the two digital outputs (Digital Outputs 6 and 7) is done using the VCL functions Set_Digout() and Clear_Digout().

```
Set_DigOut (DigOut6)
```

will set Digital Output 6 On (active). VCL can monitor the present value of a digital output driver: the bit variable Digx_Output (where “x” is the digital output channel number) is automatically filled with the present value of the driver output (On or Off).

It is important to note that all outputs are active Low. With 100% PWM or an output of “On,” the FET or transistor will be pulling hard to ground. A DVM on the output will measure near 0 volts.

Potentiometer Inputs

These controllers have two potentiometer inputs, which are typically used for throttle and brake. Many features (mapping, acceleration rates, etc.) are built in as 1313/1314 programmer parameters. Still, there are times that these potentiometer inputs may be needed for other functions such as steering angle or height sensing, or simply as data inputs. The standard way to input pot information is to set the parameter Throttle Type (or Brake Type) to an appropriate value of 1–4 as described in the Throttle Type pages, and the Throttle and Brake parameter menus. When set to a value of 1–4, the resulting signal chain can operate without the use of any VCL.

However, if an OEM wishes to control the throttle (or brake) signal chain in VCL or use the throttle (or brake) inputs for signals that are not throttle (or brake) signals, then the parameter Throttle Type (or Brake Type) should be set to a value of 5. Setting the parameter Throttle Type (or Brake Type) to a value of 5 changes the routing of the appropriate signal chain (either throttle or brake) and allows the VCL programmer access to the Throttle Pot (or Brake Pot) output variables; see Figure 15.

Using the Setup_Pot() function, executed at the beginning of a VCL program, will define the potentiometer input connection(s) as THREE_WIRE (uses Pot High and Pot Low connections), TWO_WIRE (variable resistor, or rheostat, uses Pot Low but no connection to Pot High), or ONE_WIRE (a voltage input, no connection to either Pot High or Pot Low).

THREE_WIRE potentiometer connections are the same as the 3-wire potentiometer connections shown in Figure 5 for Throttle Type 2.

TWO_WIRE potentiometer connections are the same as the 2-wire potentiometer connections shown in Figure 4 for Throttle Type 1.

ONE_WIRE potentiometer connections are the same as the Voltage Source or Current Source connections shown in Figure 5 for Throttle Type 2.

Note that the Setup_Pot() function will only work (and is only needed) if the corresponding Type is set to 5 (Throttle Type = 5 or Brake Type = 5).

```
Setup_Pot (THROTTLE_POT, THREE_WIRE)
```

will set up the throttle pot input for wiring using all three connections (pins 15, 16, 18).

To set up the brake pot input for use in VCL, use the Brake_Pot constant in place of the Throttle_Pot constant in the Setup_Pot function.

```
Setup_Pot (BRAKE_POT, TWO_WIRE)
```

will set up the brake pot input for wiring using two connections (pins 17, 18).

Quick Links:

Throttle Type

Selection p.20–23

Three-Wire/Type 1 p.20

Two-Wire/Type 2 p.21

One-Wire/Voltage p.21–22

Figure 15 p.114

Figure 17 p.118

Throttle menu p.47

Brake menu p.49

Setup_Pot () p.131

The 0–100% position of the potentiometer is represented by a value from 0–32767 in VCL, and is set up through the VCL Setup_Pot() function. It is important to use the correct setup (ONE_WIRE, TWO_WIRE, or THREE_WIRE) since the input is automatically re-scaled for 0–100% based on the wiring used; for example, the voltage at the Pot Low pin is automatically subtracted and re-scaled on a THREE_WIRE pot.

Another effect of setting the Throttle Type = 5 is that the signal chain for the throttle now gets its input from a different source. The input to the throttle chain is now a VCL variable called VCL_Throttle instead of the throttle pot. Similarly, Brake Type = 5 means the brake signal chain will get its input from a VCL variable called VCL_Brake rather than from the brake pot. The VCL_Throttle and VCL_Brake variables will need to be controlled in the VCL program.

One of the unique features of the potentiometer inputs (as opposed to the analog inputs) is that they have automatic pot fault detection functions running in the motor controller OS. The VCL programmer has access to the pot detection functions with the Setup_Pot_Faults() function. With this function, VCL can re-set the high and low thresholds at which a fault occurs. This function also forces the pot value to a definable level if a fault occurs. Note that the Setup_Pot_Faults() function will work for all throttle Types (1–5). Reference this VCL function for more details, and the **Interfacing the Throttle and Brake Commands** (below), for further information on both the Throttle and Brake processing.

Quick Link:
Setup_Pot_faults () [p.132](#)

Analog Inputs

These controllers have two generic analog inputs (pins 24 and 8). These are shared as switch inputs 1 and 2 (Sw_1, Sw_2). The values of the analog inputs are automatically placed in VCL variables Analog1_Input and Analog2_Input every 1 millisecond. Scaling is 0–10V = 0–1023.

```
User2 = Analog2_Input
```

will fill the User2 RAM variable with the value of the voltage at pin 8.

The filtered values of the analog inputs are also available and are automatically placed in VCL variables Analog1_Filtered and Analog2_Filtered. Scaling is 0–10V = 0–1023. The default filter value is 328 (10 Hz) and can be changed in VCL by changing the Analog1_Filter and Analog2_Filter values. Scaling is 0–999 Hz = 0–32767.

Analog Output

The controllers have one analog output (pin 30). This output is a special driver output. The switching stage is filtered to provide a smooth average voltage, instead of the actual PWM waveform seen on Drivers 1–5. However, AnalogOut uses the same VCL functions Put_PWM() and Automate_PWM() used by these other drivers. The scaling is 0–10 V = 0–32767.

```
Put_PWM(PWM6, 6553)
```

will generate 2.0 volts at the analog output. VCL can monitor this output using the variable Analog_Output.

INTERFACING THE THROTTLE AND BRAKE COMMANDS

VCL can interface and modify the throttle and brake signals at several points, from the potentiometer to the final motor controller command. VCL can be used to create a completely unique command, adjust parameters to provide MultiMode, or modify the throttle command based on steering angle, height, etc.

The throttle and brake signal chains within the controller are sophisticated and flexible. Before applying VCL to modify these chains, it is important to fully understand the ramifications of implementing changes. The motor and brake command signal chains are illustrated in Figure 15 and discussed below.

Throttle Processing

The top section of Figure 15 shows the throttle processing. The throttle signal chain flows left to right starting with the throttle pot (wiper). The voltage on the throttle wiper input (pin 16) is processed by the controller and has the VCL variable name *Throttle_Pot_Raw*. This variable is displayed as **Throttle Pot** (voltage) in the Monitor » Inputs menu of the 1313/1314 programmer. This throttle signal is then modified by the Throttle Type Processing and Throttle Mapping blocks.

The Throttle Type Processing block combines the *Throttle_Type* parameter and the throttle potentiometer input (*Throttle_Pot_Raw*) to create a 16-bit variable containing the magnitude of the raw command. This raw command passes to the Throttle Mapping block, which re-shapes the throttle signal magnitude and direction based on the various Throttle Menu parameters and the direction inputs.

Following the Throttle Mapping block are two switches whose purpose is to give the throttle signal a small value (1 for the forward switch, and -1 for the reverse switch) to indicate that a direction switch is On—but only if the throttle signal output from the Throttle Mapping block = 0.

The signal then passes through a selector switch. If the *Throttle_Type* parameter is set to 5 the Throttle Mapping block output signal is ignored and the command comes from the VCL variable *VCL_Throttle*. The VCL program manipulates the *VCL_Throttle* variable to get a throttle command. When the Throttle Type is set to 1–4, the variable *VCL_Throttle* does nothing, and the Throttle Mapping block output signal passes through.

After the “Throttle Type = 5” switch, the throttle signal is modified by the multiplying and summing nodes. These nodes can be adjusted by VCL through the variables *Throttle_Multiplier* and *Throttle_Offset*. This is the basic input point for creating functions like MultiMode, dual drive algorithms, and height vs. speed control. Note that the throttle multiplier has a built-in “divide by 128.” This allows the VCL to either multiply (*Throttle_Multiplier* > 128) or divide (*Throttle_Multiplier* < 128) the nominal throttle value. Typically the default multiplier is set to 128, thus having no net effect. Both *Throttle_Multiplier* and *Throttle_Offset* can be positive or negative.

The throttle signal next passes through a limiter that is active only if a pump motor is being operated (Pump Enable = On, and optionally if Regen Lower Enable = On, the pump motor can operate in reverse). This limiter acts to constrain the pump's throttle signal to 100% (positive for forward motion, or negative when Regen_Lower_Enable is On), which is appropriate for controlling a hydraulic pump motor in speed mode.

The output of the multiplying, summing and (pump) limiter nodes is the VCL variable called *Mapped_Throttle*, which is displayed as Mapped Throttle (percent) in the Monitor » Inputs menu of the 1313 handheld or 1314 PC programmers. A VCL program can control this throttle signal by changing the three variables *VCL_Throttle* (only if Throttle Type = 5), *Throttle_Multiplier*, and *Throttle_Offset*. The effect of these variables can be observed as in the 1313/1314's Mapped Throttle or the VCL *Mapped_Throttle* variables. Checking the value of Mapped Throttle (*Mapped_Throttle*) is a good way to see if the Throttle Menu parameters are set correctly.

Prior to passing to the final selector switch, the throttle signal passes through a selector which validates for forward pump operation in speed modes. It will set the pump throttle signal = 0% when, in either speed mode, Pump Enable = On and the *Mapped_Throttle* is < 0 (negative). This will prevent a pump operating in reverse when Regen Lower Enable = Off, or if the controller is in Torque Mode.

The throttle signal continues to a selector switch that will set the throttle signal = 0% if any of the following conditions are present: *Interlock_State* = Off, a fault has set throttle request = 0% (Fault action = ShutdownThrottle), or if *Main_State* ≠ 5 or 10, or an invalid throttle. An invalid throttle is when both the Forward and Reverse switches are On, for example.

Quick Links:

Figure 15 [p.114](#)

Throttle_Type parameter [p.47](#)

Throttle menu [p.47](#)

Pump parameter menu

Speed Mode = 0 [p.33](#)

Speed Mode = 1 [p.34–40](#)

Monitor Inputs [p.77](#)

Monitor menu

Inputs » Interlock State [p.77](#)

Controller » Main State [p.81](#)

Troubleshooting

Chart [p.146–153](#)

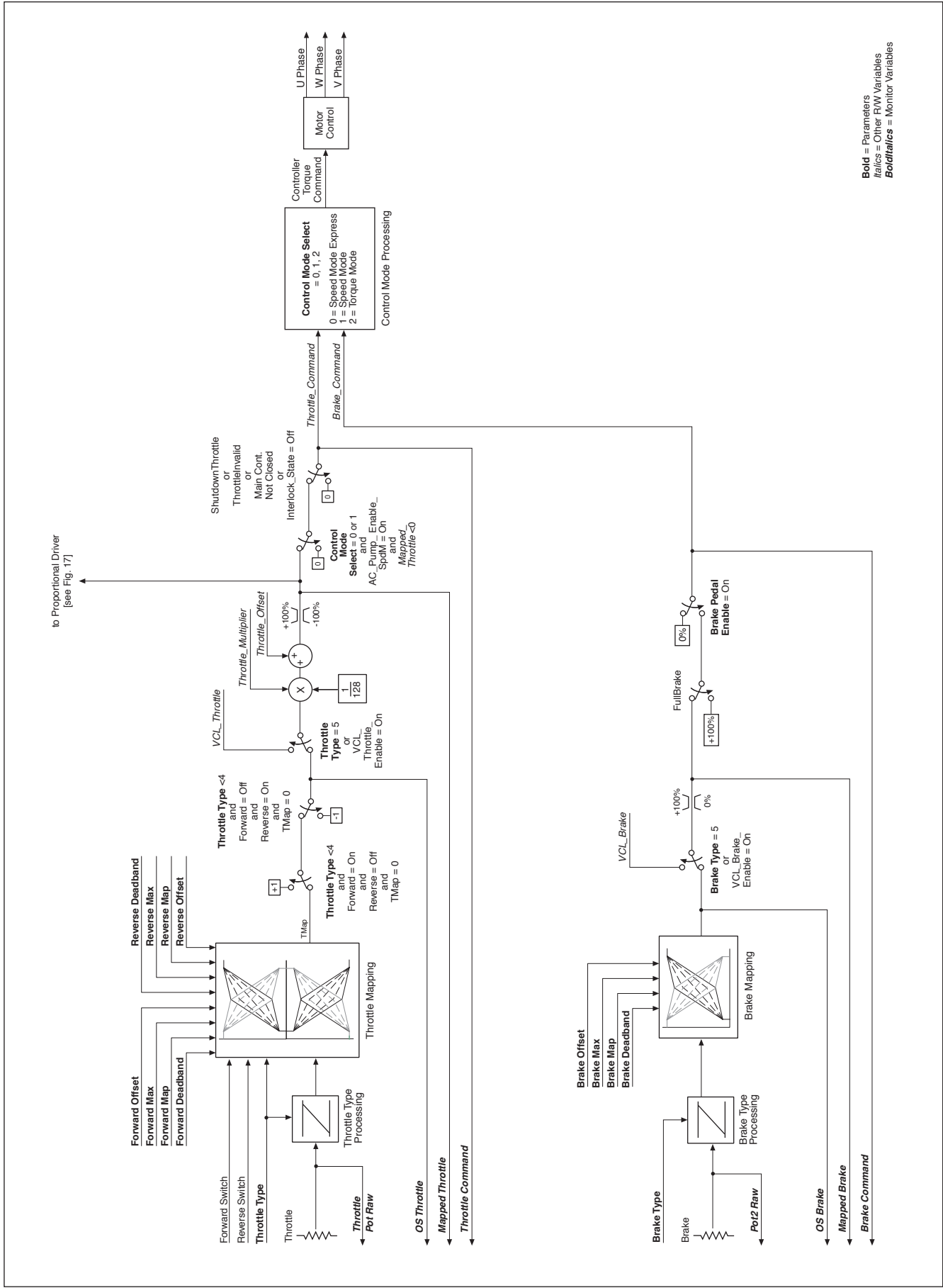


Figure 15
Motor command diagram.

After this final selector switch the throttle signal is the VCL variable called `Throttle_Command`, which is displayed as `Throttle Command` in the 1313/1314 Monitor » Inputs menu. `Throttle_Command` is the final value of the throttle signal chain that is input to the Control Mode Processing block; see Figure 16. Checking the value of `Throttle Command` using the 1313/1314 is a good way to see the final throttle signal.

If `ABS (Throttle_Command) > 1` count, the motor controller will output signals to the motor to make it spin.

For investigating why a motor is not spinning, it is useful to use the 1313/1314 programmer to check the state of the throttle signal from beginning to end: using `Throttle_Pot_Raw`, `Mapped_Throttle`, and `Throttle_Command`. Once these values are known, the Motor Command Diagram (Figure 15) can be used to find how that signal progressed from input to final value.

The following throttle processing variables are accessible by VCL:

VCL Variable CAN Object Index	Access	Description
<code>Throttle_Pot_Raw</code> 0x3215 0x00	Read Only	Voltage measurement at pin 16, scaled for the proper wiring.
<code>OS_Throttle</code> 0x3518 0x00	Read Only	Throttle pot value after mapping, to be used in VCL when VCL Throttle Enable = On and Throttle Type = 1 – 4. Scale: 0 – 100%, 0 – 32,767.
<code>Mapped_Throttle</code> 0x3211 0x00	Read Only	Throttle pot value after mapping.
<code>VCL_Throttle</code> 0x3218 0x00	Read/Write	VCL-accessible throttle command.
<code>Throttle_Multiplier</code> 0x3213 0x00	Read/Write	Multiplies or divides the throttle signal.
<code>Throttle_Offset</code> 0x3214 0x00	Read/Write	Provides a $\pm$ offset to the throttle signal.
<code>Throttle_Command</code> 0x3216 0x00	Read Only	Command resulting from throttle processing.

Brake Processing

Brake processing is optional as it can be turned Off by setting `Brake_Pedal_Enable = Off`. If turned On, brake processing can be done with or without VCL. When the controller is in Speed Mode, any non-zero brake command will then override the throttle signal and the motor controller will brake to a stop as determined by the Brake Current Limit and Brake Taper Speed parameters. When the controller is in Torque Mode, the brake command is multiplied by 2 and subtracted from the throttle command to produce the final command; thus a 50% brake command will cancel a 100% throttle command.

The lower part of Figure 15 shows the brake signal processing section. The brake signal chain flows from left to right starting with the physical brake pot. The voltage on the brake wiper input (pin 17) is processed by the controller and has the VCL variable name `Pot2_Raw` which is displayed as `Pot2 Raw` in the programmer's Monitor » Inputs menu. This brake signal is then modified by the Brake Type Processing and Brake Mapping blocks.

The Brake Type Processing block uses the `Brake_Type` parameter and the brake potentiometer input (`Pot2_Raw`) to create a signed 16-bit variable. This brake signal then passes to the Brake Mapping block, which re-shapes the brake signal according to the various Brake Menu parameters.

The brake signal next passes through a selector switch. If the `Brake_Type` parameter is set to 5 (Brake Type = VCL input), the Brake Mapping block output signal is ignored and the command comes from the VCL variable `VCL_Brake`. The VCL program manipulates the `VCL_Brake` variable to get a brake command. Custom braking functions can be set up in this fashion; e.g., braking based on a switch

Quick Links:

[Brake Menu](#)

[Brake Pedal Enable](#) p.49

[Current Limits Menu](#)

[Brake Current Limit](#) p.44

[Brake Taper Speed](#) p.42

[Monitor](#)

[Inputs » Pot2 Raw](#) p.77

[Figure 15](#) p.114

[Figure 16](#) p.117

position or an internal fault. The brake potentiometer can still be used, but must be set up using the Setup_Pot() function. When the Brake Type is set to 1–4, the variable VCL_Brake does nothing and the Brake Mapping block output signal passes through.

After the “Brake Type = 5” switch, the brake signal passes through a limiter which limits the brake signal to a range of 0–100% (0–32767). After the limiter the brake signal is a VCL variable called Mapped_Brake, which is displayed as Mapped Brake in the Monitor » Inputs menu. Checking the value of Mapped_Brake is a good way to see if the Brake Menu parameters are set correctly. A VCL program can control the brake by changing the variable VCL_Brake (only if Brake Type = 5).

The brake signal then goes through a selector which will change it to 100% if the fault-action FullBrake is active. (see Troubleshooting Chart).

The brake signal then goes through a third selector switch that will set the brake signal = 0% if the Brake Pedal Enable parameter is set Off. If set On, then the brake signal will pass through to the Control Mode Processing block. The brake signal after this third selector switch is the VCL variable called Brake_Command and Brake Command in the Monitor » Inputs menu.

Brake_Command is the final value of the brake signal chain that is input to the Control Mode Processing block; see Figure 16. Checking the value of Brake_Command using the 1313 is a good way to see the final brake signal. Note: If Brake_Command is non-zero in Speed Mode Express or Speed Mode, the Throttle_Command will be set to 0% (see Fig. 16).

The following brake processing variables are accessible by VCL:

VCL Variable <i>CAN Object Index</i>	Access	Description
Pot2_Raw 0x3217 0x00	Read Only	Voltage measurement at pin 17.
OS_Brake 0x3519 0x00	Read Only	Brake pot value after mapping, to be used in VCL when VCL Brake Enable = On and Brake Type = 1–3.
VCL_Brake 0x3219 0x00	Read/Write	VCL-accessible brake command.
Mapped_Brake 0x3212 0x00	Read Only	Brake pot value after mapping.
Brake_Command 0x321A 0x00	Read Only	Command resulting from brake processing.

Control Mode and Motor Control Processing

Figure 16 begins with the Throttle_Command and Brake_Command inputs and routes the signals according to the selected control mode. For the speed modes, a switch will zero the Throttle_Command if the Brake_Command is any value but 0%. The signal chains are then directed to Speed Mode Express, Speed Mode, or Torque Mode based on Control Mode Select. Note that in Torque Mode there is no emergency reverse.

The control mode function uses algorithms to convert the incoming throttle and brake signals, the motor rpm, and associated parameter settings into a Controller Torque Command.

Based upon the control mode selected, the desired Controller Torque Command is calculated, which is passed to the Motor Control block (see Figure 15). The Motor Control block uses its mathematical model of the specific motor & technology to generate the high efficiency three-phase outputs that drive the AC motor via the cables connected to the U, V, and W terminals. For AC Induction motors, 504 motor characterizations are available using the Motor Type parameter, or a motor can be user characterized (Motor Type 0) following the procedures in Chapter 8A.

Quick Links:

Brake Menu

Brake parameters [p.49](#)

Monitor Inputs

Brake Command [p.77](#)

Figure 15 [p.114](#)

Figure 16 [p.117](#)

Quick Links:

Figure 15 [p.114](#)

Figure 16 [p.117](#)

Motor Type [p.59](#)

Automated Characterization

AC Induction motors [p.93](#)

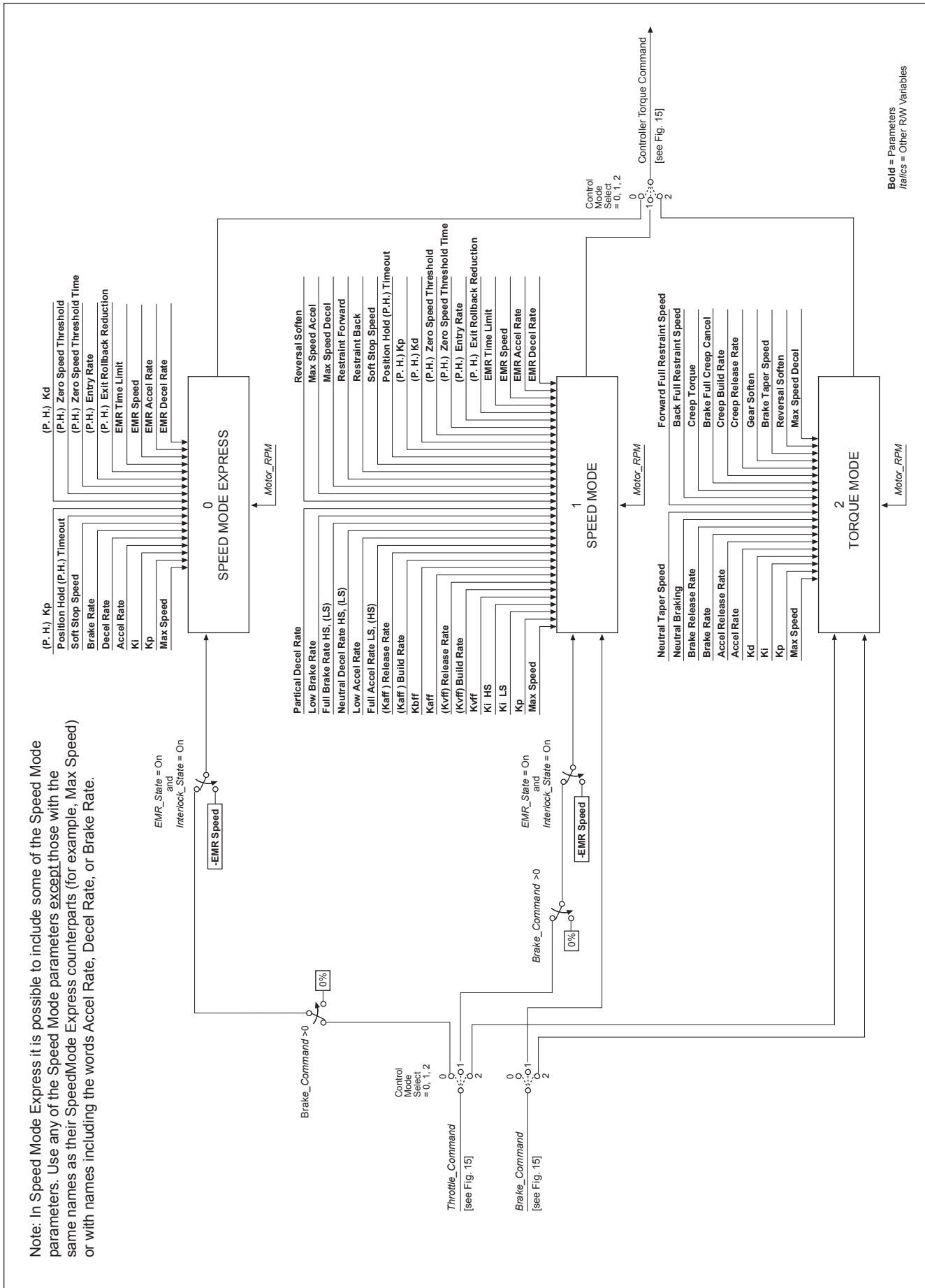


Figure 16
Control Mode processing.

INTERFACING THE PROPORTIONAL CURRENT DRIVER

VCL code can directly interface the proportional current driver (PD), as shown in Figure 17. VCL can change the working parameters of the PD and can provide the command.

Quick Links (for PD text):

Proportional Driver
parameters [p.54](#)

Monitor

PD Input [p.78](#)

PD PWM [p.79](#)

PD Current [p.79](#)

Depending on how the PD system is to be used, certain parameters need to be set; they can be set via the 1313/1314 programmer or via VCL.

1. **PD_Enable** must be set On for current control, otherwise the PD_Output will be controlled by the VCL function Put_PWM(PWM5,xxxx), which is voltage control.
2. **Hyd_Lower_Enable** must be set On to use the throttle input to control lowering in a hydraulic lift/lower system.
3. **Hyd_Lower_Enable** must be set Off to allow using a VCL variable (VCL_PD_Throttle) as the PD command.

Once the PD parameters are set, the PD_Throttle variable will be mapped between PD_Min_Current and PD_Max_Current and sent to the dither function. Note that Mapped_Throttle is inverted; lowering its value (making it more negative) increases the PD_Throttle value.

The Dither function adds and subtracts from the current command to the PD based on PD_Dither_Percent, at a rate set by PD_Dither_Period.

The dithered current command is compared to the present PD_Current and the error is fed into a PI controller. The feedback gains are set by the parameters PD Kp (proportional gain) and PD Ki (integral gain). The final output is the VCL variable PD_Output, which is displayed in the 1313/1314 programmer's Monitor » Outputs menu as PD PWM.

The following PD processing variables are accessible by VCL:

VCL Variable <i>CAN Object Index</i>	Access	Description
Mapped_Throttle 0x3211 0x00	Read Only	Command from throttle section.
VCL_PD_Throttle 0x320E 0x00	Read/Write	VCL-accessible PD command.
PD_Throttle 0x3210 0x00	Read Only	Resultant command to the PD.
PD_Current 0x321C 0x00	Read Only	Average current flowing in the PD.
PD_Output 0x321E 0x00	Read Only	Resultant PWM at PD output.

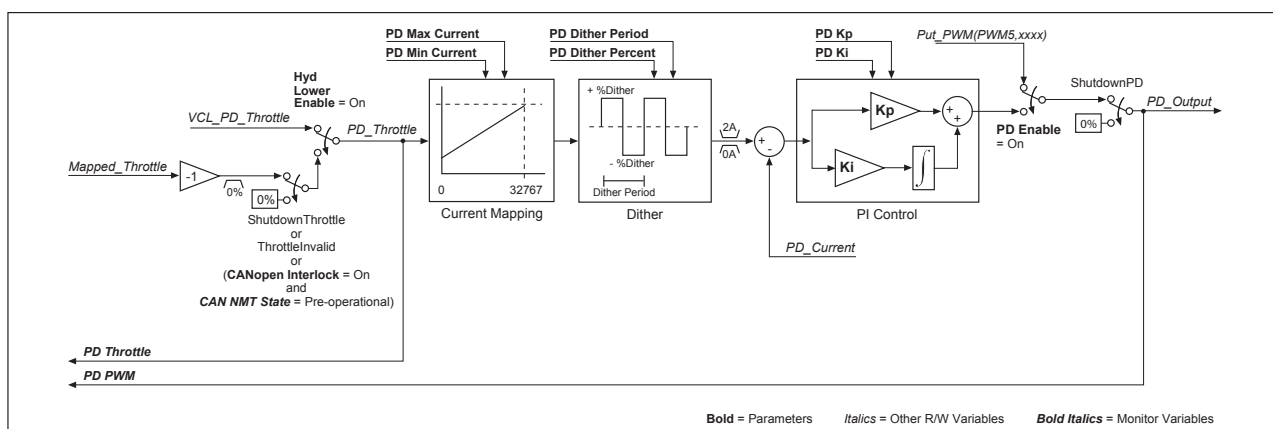


Figure 17
Proportional driver processing.

USING THE FAULT HANDLER IN VCL

The controller's operating system detects various faults and takes appropriate fault actions to protect the controller. These faults have fault codes that are flashed on the controller status LEDs. The fault text is displayed in the System Faults and Fault History menus of the 1313 handheld and 1314 PC programmers. These operating system faults are covered in more detail in Chapter 10's Troubleshooting Chart.

Additionally, the operating system makes the status of the operating system faults available for use in VCL programs in the form of nine variables called Status1, Status2, Status3, Status4, Status5, Status6, Status7, Status8, and Status9. Each of these 16-bit variables contains the status of 8 faults in the lower byte (the upper byte is always set to 0). These Status1–9 variables are read only (RO) and can be used in a VCL program to trigger additional fault actions such as sending fault text messages to a display or blinking a dashboard LED.

Listed below are the bit locations of each of the operating system faults in the Status 1–9 variables, applicable to the default 1232E/SE - 1238E/SE controllers:

Status1

Bit0 = Main Contactor Welded (Code 38)
 Bit1 = Main Contactor Did Not Close (Code 39)
 Bit2 = Pot Low OverCurrent (Code 45)
 Bit3 = Throttle Wiper Low (Code 42)
 Bit4 = Throttle Wiper High (Code 41)
 Bit5 = Pot2 Wiper Low (Code 44)
 Bit6 = Pot2 Wiper High (Code 43)
 Bit7 = EEPROM Failure (Code 46)

Status2

Bit0 = HPD/Sequencing Fault (Code 47)
 Bit1 = Severe B+ Undervoltage (Code 17)
 Bit2 = Severe B+ Overvoltage (Code 18)
 Bit3 = B+ Undervoltage Cutback (Code 23)
 Bit4 = B+ Overvoltage Cutback (Code 24)
 Bit5 = Sin/Cos Sensor Fault (Code 36)
 Bit6 = Controller Overtemp Cutback (Code 22)
 Bit7 = Controller Severe Undertemp (Code 15)

Status3

Bit0 = Controller Severe Overtemp (Code 16)
 Bit1 = Coi11 Driver Open/Short (Code 31)
 Bit2 = Coi12 Driver Open/Short (Code 32)
 Bit3 = Coi13 Driver Open/Short (Code 33)
 Bit4 = Coi14 Driver Open/Short (Code 34)
 Bit5 = PD Open/Short (Code 35)
 Bit6 = Main Open/Short (Code 31)
 Bit7 = EMBrake Open/Short (Code 32)

Status4

Bit0 = Precharge Failed (Code 14)
 Bit1 = Digital Out 6 Overcurrent (Code 26)
 Bit2 = Digital Out 7 Overcurrent (Code 27)
 Bit3 = Controller Overcurrent (Code 12)
 Bit4 = Current Sensor Fault (Code 13)
 Bit5 = Motor Temp Hot Cutback (Code 28)
 Bit6 = Parameter Change Fault (Code 49)
 Bit7 = Motor Open (Code 37)

Status5

Bit0 = External Supply Out of Range (Code 69)
Bit1 = Motor Temp Sensor Fault (Code 29)
Bit2 = VCL Run Time Error (Code 68)
Bit3 = +5V Supply Failure (Code 25)
Bit4 = OS General (Code 71)
Bit5 = PDO Timeout (Code 72)
Bit6 = Encoder Fault (Code 36)
Bit7 = Stall Detected (Code 73)

Status6

Bit0 = Bad_Calibrations_Fault (Code 82)
Bit1 = [Not Used]
Bit2 = Emer Rev HPD (Code 47)
Bit3 = [Not Used]
Bit4 = Motor Type Fault (Code 89)
Bit5 = Supervisor Fault (Code 77)
Bit6 = Motor Characterization Fault (Code 87)
Bit7 = [Not Used]

Status7

Bit0 = [Not Used]
Bit1 = VCL/OS Mismatch (Code 91)
Bit2 = EM Brake Failed to Set (Code 92)
Bit3 = Encoder LOS (Limited Operating Strategy) (Code 93)
Bit4 = Emer Rev Timeout (code 94)
Bit5 = Dual Severe Fault (Code 75)
Bit6 = Fault On Other Traction Controller (Code 74)
Bit7 = Illegal Model Number (Code 98)

Status8

Bit0 = [Not Used]
Bit1 = [Not Used]
Bit2 = [Not Used]
Bit3 = Parameter Mismatch (Code 99)
Bit4 = Severe KSI Undervoltage (Code 17)
Bit5 = [Not Used]
Bit6 = Insulation Resistance Low (Code 76)
Bit7 = Encoder Pulse Count Fault (Code 88)

Status9

Bit0 = Supervisor Incompatible (Code 78)
Bit1 = [Not Used]
Bit2 = [Not Used]
Bit3 = PMAC Commissioning Needed (Code 19)
Bit4 = [Not Used]
Bit5 = [Not Used]
Bit6 = Driver Supply (Code 83)
Bit7 = [Not Used]

CANbus Emergency Messages

The Status1–9 faults form CAN emergency message using little-endian byte ordering, where: Byte1&2 is the Error Category, Byte3 is the error register which is “set” (01) if there is an active fault, and Bytes 4–8 map into the Status 1–9 faults as per the error category.

Error category = 0x1000 (for Status1 – Status5)

Byte4 = Status1

Byte5 = Status2

Byte6 = Status3

Byte7 = Status4

Byte8 = Status5

Error category = 0x1001 (for Status6 – Status 10)

Byte4 = Status6

Byte5 = Status7

Byte6 = Status8

Byte7 = Status9

Byte8 = Status ... n/a (Status 10 is not utilized)

Example

CANbus error 00 10 01 00 00 00 08 00 equates to Status4, Bit3, Controller Overcurrent (Code 12)

The OEM-defined user faults, described next, are also accessible in a CANbus emergency message.

Error category = 0x6200 (for UserFault1 and UserFault2, OEM Defined, implemented in VCL)

Byte4 = UserFault1

Byte5 = UserFault2

Byte6 = Not applicable

Byte7 = Not applicable

Byte8 = Not applicable

Example

CANbus error 00 62 01 02 00 00 00 00 equates to VCLfault1, Bit 1 (Code 52)

OEM-defined User Faults

The operating system provides the capability to create OEM-defined custom faults using VCL. Just as with system faults, the VCL fault codes are flashed on the controller Status LEDs and fault text is displayed on the System Faults and Fault History menus of the 1313/1314 programmer. Optionally, the VCL can assign fault actions to occur automatically when the associated fault is set. Sixteen VCL faults are available, stored in the VCL variables UserFault1 and UserFault2. The UserFault1,2 variables are Read/Write (R/W) and the 16 faults are mapped & stored in the lower byte of each variable as shown below:

UserFault1

```

    Bit0 = VCLfault1 (Code 51)
    Bit1 = VCLfault2 (Code 52)
    Bit2 = VCLfault3 (Code 53)
    Bit3 = VCLfault4 (Code 54)
    Bit4 = VCLfault5 (Code 55)
    Bit5 = VCLfault6 (Code 56)
    Bit6 = VCLfault7 (Code 57)
    Bit7 = VCLfault8 (Code 58)

```

UserFault2

```

    Bit0 = VCLfault9 (Code 59)
    Bit1 = VCLfault10 (Code 61)
    Bit2 = VCLfault11 (Code 62)
    Bit3 = VCLfault12 (Code 63)
    Bit4 = VCLfault13 (Code 64)
    Bit5 = VCLfault14 (Code 65)
    Bit6 = VCLfault15 (Code 66)
    Bit7 = VCLfault16 (Code 67)

```

The “Code” numbers result in the flashing of the controller status LEDs that help identify the fault without use of a 1313/1314 programmer. Setting the fault in VCL is done by setting the associated fault bit in the UserFault1 and 2 variables. Clearing the fault must also be handled by VCL and is done by clearing the associated fault bit. For example:

```

    if (BDI_Percentage < 10)
    {
        UserFault1.2 = ON ;           Set VCL Fault bit
    }
    else
    {
        UserFault1.2 = OFF ;          Clear VCL Fault bit
    }

```

This VCL will check to see if the Battery Discharge Indicator is less than 10%. If it is, UserFault1 Bit1 (Code 52) is set. If the BDI is not less than 10%, the fault is cleared. Using just the VCL above in a program will only result in the flashing of a code 52 on controller status LEDs and no fault actions will result nor will the 1313/1314 programmer display any text about the fault.

To add automatic fault actions to the VCL faults, the VCL programmer must define the desired fault actions by using the sixteen VCL variables: User_Fault_Action_01 through User_Fault_Action_16. Each of the UserFault bits has a corresponding User_Fault_Action_xx variable (where “xx” is the number of the VCL fault bit). When a VCL fault bit is set, the actions defined in the corresponding User_Fault_Action_xx variable will be automatically executed by the operating system. For any user fault, single or multiple actions can be specified. The available fault actions in the User_Fault_Actions_xx variables are listed below:

Variable	Fault Action
Bit0 = ShutdownMotor	Disable the motor.
Bit1 = ShutdownMainContactor	Shut down the main contactor (only if Main Enable = On).
Bit2 = ShutdownEMBrake	Shut down the EM brake (only if EM Brake Disable Upon Fault = On).
Bit3 = ShutdownThrottle	Set the Throttle_Command = 0%.
Bit4 = ShutdownInterlock	Set the Interlock_State = Off.
Bit5 = ShutdownDriver1	Shut down Driver1.
Bit6 = ShutdownDriver2	Shut down Driver2.
Bit7 = ShutdownDriver3	Shut down Driver3.
Bit8 = ShutdownDriver4	Shut down Driver4.
Bit9 = ShutdownPD	Shut down Proportional Driver.
Bit10 = FullBrake	Set the Brake_Command = 100%.
Bit11 = [Reserved]	N/A.
Bit12 = TrimDisable	Disable Dual Drive trim calculation.
Bit13 = SevereDual	For Dual Drive system, one controller has a severe fault but the main contactor must stay closed so the other controller can continue to operate.
Bit14 = ShutdownSteer	Steer angle = 0°.
Bit15 = LOSDual	For Dual Drive system, set the max speed to Dual_LOS_Max_Speed parameter for operation in Limited Operating Strategy.

The User_Fault_Action_xx variables should be set up at the beginning of a VCL program (before the main loop) as these fault actions should be defined only once in a program. Here is another example:

```
User_Fault_Action_02 = 24 ;Set fault action to ShutdownInterlock
;and ShutdownThrottle
MainLoop:
if (BDI_Percentage < 10)
{
UserFault1.2 = ON ;          Set User Fault bit
Put_Spy_Text("BDI Low") ; Send message to Model 840 display
}
else
{
UserFault1.2 = OFF ;        Clear User Fault bit
}
goto MainLoop
```

This time when UserFault1.2 is set, the operating system will ShutdownInterlock and ShutdownThrottle (which will result in a Throttle_Command = 0%) in addition to flashing the code 52 on the controller status LEDs. An additional VCL line was added (Put_Spy_Text (“BDI Low”)) to show how additional actions beyond those provided in the User_Fault_Action_xx can be programmed using VCL. In this example the Put_Spy_Text(“BDI Low”) will result in the message “BDI Low” appearing on the model 840 display (presumably as a message to the vehicle operator). This example will still not result in any display on the 1313/1314 programmer System Faults and Fault History menus.

To add the fault text on the System Faults and Fault History menus it is necessary to create a Fault Definition. Creating 1313/1314 fault definitions is a subject that is covered in detail in the VCL Programmer’s Guide (Section 5, Support for the 1313/1314 Handheld Programmer). Here is an example of a fault definition:

```

User_Fault_Action_02 = 24 ;           Set fault action to ShutdownInterlock
                                     ;and ShutdownThrottle

MainLoop:
if (BDI_Percentage < 10)
{
UserFault1.2 = ON                     ;Set User fault bit
Put_Spy_Text("BDI Low")              ;Send message to Model 840 display
}
else
{
UserFault1.2 = OFF                   ;Clear User fault bit
}
goto MainLoop

; PARAMETER_ENTRY    "BDI Low Fault"
; TYPE                FAULTS
; WIDTH              8BIT
; ALT_ADDRESS         Hist_UserFault1
; ADDRESS             UserFault1
; BITSELECT           1
; BITACTIVELOW        NO
; END

```

This example will result in the exact same actions as the last example, except now the fault will be displayed in the programmer's System Faults menu (only while the fault is set) and this fault will be logged into the Fault History menu after being set. The text displayed in either of these menus will be the text defined in the fault definition (in this example "BDI Low Fault" will be displayed). The variable Hist_UserFault1 is listed as an ALT_ADDRESS. This line controls whether the fault gets logged into fault history (and thus appears in the programmer's Fault History menu). The two variables that can be used in the fault definitions for ALT_ADDRESS are Hist_UserFault1 and Hist_UserFault2; these should be used in the fault definitions with the corresponding UserFault1 and UserFault2 variables. To use VCL to clear fault history, use the VCL function Clear_Diaghist(). Also note that this example fault definition was for bit 1 of UserFault1. The VCL example set and cleared this bit by using the UserFault1.2 notation (".2" being the mask that defines bit 1).

CAN Indexes for the User Fault Variables

The CAN indexes for the various OEM-defined VCL user fault variables are shown below.

0x3238	0x00	UserFault1
0x389A	0x00	UserFault1_History
0x3231	0x00	Hist_UserFault1
0x3239	0x00	UserFault2
0x389B	0x00	UserFault2_History
0x3232	0x00	Hist_UserFault2
0x323B	0x00	User_Fault_Action_01
0x323C	0x00	User_Fault_Action_02
0x323D	0x00	User_Fault_Action_03
0x323E	0x00	User_Fault_Action_04
0x323F	0x00	User_Fault_Action_05
0x3240	0x00	User_Fault_Action_06
0x3241	0x00	User_Fault_Action_07
0x3242	0x00	User_Fault_Action_08
0x3243	0x00	User_Fault_Action_09
0x3244	0x00	User_Fault_Action_10
0x3245	0x00	User_Fault_Action_11
0x3246	0x00	User_Fault_Action_12
0x3247	0x00	User_Fault_Action_13
0x3248	0x00	User_Fault_Action_14
0x3249	0x00	User_Fault_Action_15
0x324A	0x00	User_Fault_Action_16

VCL FUNCTIONS SPECIFIC TO 1232E/SE, 1234E/SE, 1236E/SE, AND 1238E/SE CONTROLLERS

Function descriptions are provided here for the functions that are unique to these controllers. They are presented in the same format that is used in the VCL Common Functions Manual for the common functions.

ENABLE_PRECHARGE()

This function is designed to precharge the capacitor bank before engaging a main contactor thereby preventing current surges and to protect controller internal components and main contactor tips. This function turns on the request for precharge of the capacitor bank from KSI.

When the precharge function is enabled, power will be supplied to the capacitor bank until the voltage is within the factory-set range of KSI, or the precharge time limit has expired, or the precharge resistor energy range has been exceeded. The current state of precharge is shown by the precharge variable (Precharge_State), which has the following values:

- 0 – Precharge has not yet been done.
- 1 – Precharge is in progress.
- 2 – Precharge has passed.
- 3 – Precharge has been aborted by the Disable_Precharge() function.
- 4 – Precharge has exceeded the precharge resistor energy limit.
- 5 – Precharge has exceeded the one-second time limit.

Syntax: **Enable_Precharge()**

Parameters: None.

Returns:

- 0 – Precharge not enabled.
- 1 – Precharge successfully enabled.

Error Codes: None.

Example: Enable_Precharge()

This will attempt to precharge the capacitor bank.

Note: Use this function only when the precharge is under VCL control.

DISABLE_PRECHARGE()

This function is designed to abort the precharge function and clear any precharge fault. This function aborts the request for precharge of the capacitor bank from KSI. The resultant state of the precharge variable (Precharge_State) will be set to = 3 (for precharge aborted). The precharge states are:

- 0 – Precharge has not yet been done.
- 1 – Precharge is in progress.
- 2 – Precharge has passed.
- 3 – Precharge has been aborted by the Disable_Precharge() function.
- 4 – Precharge has exceeded the precharge resistor energy limit.
- 5 – Precharge has exceeded the one-second time limit.

Syntax: **Disable_Precharge()**

Parameters: None.

Returns:

- 0 – Precharge not enabled.
- 1 – Precharge successfully enabled.

Error Codes: None.

Example: Disable_Precharge()

This will attempt to abort the precharge of the capacitor bank and will clear any precharge fault.

SET_DIGOUT()

This function turns on the selected digital output. The digital outputs are active low (On = driver on and pulled to ground, Off = open circuit at the pin).

Syntax: `Set_Digout (DigOut_ID)`

Parameters: DigOut_ID is the digital output identification.
DigOut6 = Digital Output 6 (pin 19).
DigOut7 = Digital Output 7 (pin 20).

Returns:

0 – Selected digital output not set.
1 – Selected digital output successfully set.

Error Codes: Bad_ID is returned when DigOut_ID is not in the range of DigOut6 to DigOut7.

Example: `Set_Digout (DigOut6)`

This example will set Digital Output 6 (pin 19) On (active low, pulled to ground).

CLEAR_DIGOUT()

This function turns off the selected digital output. The digital outputs are active low (On = driver on and pulled to ground, Off = open circuit at the pin).

Syntax: `Clear_Digout (DigOut_ID)`

Parameters: DigOut_ID is the digital output identification.
DigOut6 = Digital Output 6 (pin 19).
DigOut7 = Digital Output 7 (pin 20).

Returns:

0 – Selected digital output not cleared.
1 – Selected digital output successfully cleared.

Error Codes: Bad_ID is returned when DigOut_ID is not in the range of DigOut6 to DigOut7.

Example: `Clear_Digout (DigOut6)`

This example will set Digital Output 6 (pin 19) Off (open circuit).

ENABLE_EMER_REV()

This function is used to engage emergency reverse using VCL. The EMR Type parameter must be set to = 1 in order for the Enable_Emer_Rev() function to operate. If the system emergency reverse state is enabled (EMR_State bit variable = On), the emergency reverse function will operate according to the Emergency Reverse parameter settings. To view the current emergency reverse state, see Monitor » Inputs: Emer Rev on the 1313/1314 programmer.

When the EMR_Type is set to = 1 and neither the Enable_Emer_Rev() nor the Disable_Emer_Rev() function has been called, the state for emergency reverse is Off (EMR_State bit variable = Off).

Syntax: `Enable_Emer_Rev()`

Parameters: None.

Returns:

0 – Emergency reverse not enabled.

1 – Emergency reverse successfully enabled.

Error Codes: None.

Example: `Enable_Emer_Rev()`

This will enable the emergency reverse function.

DISABLE_EMER_REV()

This function is used to disengage emergency reverse using VCL. The EMR Type parameter must be set to = 1 in order for the Disable_Emer_Rev() function to operate. If the system emergency reverse state is disabled (EMR_State bit variable = Off), the emergency reverse function will stop operating and normal motor control function will resume (including an HPD/SRO check if the HPD/SRO Enable parameter is set to On). To view the current emergency reverse state, see Monitor » Inputs: Emer Rev on the 1313/1314 programmer.

When the EMR_Type is set to = 1 and neither the Enable_Emer_Rev() nor the Disable_Emer_Rev() function has been called, the state for emergency reverse is Off (EMR_State bit variable = Off).

Syntax: `Disable_Emer_Rev()`

Parameters: None.

Returns:

0 – Emergency reverse not disabled.

1 – Emergency reverse successfully disabled.

Error Codes: None.

Example: `Disable_Emer_Rev()`

This will disable the emergency reverse function.

SET_INTERLOCK()

This function is used to engage the system interlock using VCL. The Interlock Type parameter must be set to = 1 in order for the Set_Interlock() function to operate. If the system interlock is set (Interlock_State bit variable = On), the throttle input signal is allowed to pass along the throttle chain; see Figure 15. Additionally, if the main contactor is used (Main Enable parameter = On), setting the interlock will request the main closed state from the main contactor state machine. To view the current interlock state, see the Monitor » Inputs: Interlock. To view the current main contactor state, see Monitor » Controller: Main State.

When the Interlock_Type is set to = 1 and neither the Set_Interlock() nor the Clear_Interlock() function has been called, the state for the interlock is Off (Interlock_State bit variable = Off).

Syntax: `Set_Interlock()`

Parameters: None.

Returns:

0 – Interlock not set.

1 – Interlock successfully set.

Error Codes: None.

Example: `Set_Interlock()`

This will engage the system interlock.

CLEAR_INTERLOCK()

This function is used to disengage the system interlock using VCL. The Interlock Type parameter must be set to = 1 in order for the Set_Interlock() function to operate. If the system interlock is cleared (Interlock_State bit variable = Off), the throttle input signal is not allowed to pass along the throttle chain; see Figure 15. Additionally, if the main contactor is used (Main Enable parameter = On), clearing the interlock will request the main open state from the main contactor state machine. To view the current interlock state, see Monitor » Inputs: Interlock. To view the current main contactor state, see Monitor » Controller: Main State.

When the Interlock_Type is set to = 1 and neither the Set_Interlock() nor Clear_Interlock() function has been called, the default state for the interlock is Off (Interlock_State bit variable = Off).

Syntax: `Clear_Interlock()`

Parameters: None.

Returns:

0 – Interlock not cleared.

1 – Interlock successfully cleared.

Error Codes: None.

Example: `Clear_Interlock()`

This will disengage the system interlock.

SETUP_POT()

This function determines the type of electrical connection of the Throttle Wiper (J1-16) and Pot2 High (J1-27) potentiometers (pot inputs).

One wire pots are a 0 – 5 V input.

Two wire pots are rheostats.

Three wire pots use the pot-high and pot-low as their high and low references.

Syntax: **Setup_Pot (Pot_ID, Type)**

Parameters:

POT_ID Pot (ID) Selector (THROTTLE_POT, or BRAKE_POT)

Type Pot type (ONE_WIRE, TWO_WIRE, or THREE_WIRE)

Returns:

0 Setup not successful.

1 Setup successful.

Errors

BAD_ID POT_ID out of Range.

BAD_P_TYPE Pot Type out of Range.

GET_POT()

This function returns the value of the selected pot; 0 – 32767 equating to 0 – 100% over the 0 – 5.00V input range.

Syntax: **Get_Pot (POT_ID)**

Parameters:

POT_ID Pot Selector (THROTTLE_POT, or BRAKE_POT)

Returns:

0 Pot = 0 or error.

n Pot value.

Errors

BAD_ID POT_ID out of Range.

SETUP_POT_FAULTS()

This function re-sets the upper and lower wiper fault voltages for the specified pot-input that will be used to activate a pot-wiper voltage fault. The valid range for the function parameters are 0 – 6.25 V (0 – 400 counts). If this function is not run, the default thresholds depend on the Throttle Type (or Brake Type) parameter setting; see table below. If the Throttle Type = 5 (or Brake Type = 5), the VCL function Setup_Pot will determine what fault thresholds are used.

Throttle Type	Low Fault Threshold	High Fault Threshold
1	0.1V	5.5V
2	NONE	5.5V
3	0.1V	5.5V
4	0.1V	5.5V
5 (ONE_WIRE)	NONE	5.5V
5 (TWO_WIRE)	0.1V	5.5V
5 (THREE_WIRE)	0.1V	5.5V

Syntax: `Setup_Pot_Faults(Pot_ID, Low_Fault, High_Fault, Fault_Value)`

Parameters: `Pot_ID` identifies the throttle whose fault limits are being set:

`Throttle_Pot`

`Brake_Pot`

`Low_Fault`: Specifies the lower threshold voltage limit.

Scaling: 1V = 64 counts.

`High_Fault`: Specifies the upper threshold voltage limit.

Scaling: 1V = 64 counts.

`Fault_Value`: The value that is used for the pot input when there is a fault (0–32767).

Scaling: $\pm 32767 = \pm 100\%$.

Returns:

0 – Setup did not execute.

1 – Setup successful.

Error Codes: `Bad_ID` is returned when an incorrect pot ID is used.
`Param_Range` is returned when the voltage value is not within range.

Example: `Setup_Pot_Faults(THROTTLE_POT, 19, 320, 4000)`

For the throttle pot, this will set the lower pot voltage to 0.3V (19/64) and the upper pot voltage to 5.0V (320/64). When there is a pot fault, the value of 4000 will be used. That is 4000/32767 of the full output, or roughly 12%.

START_PUMP()

This function is used to turn on the pump lift contactor (pin 4). If the parameter Drivers » Driver3 » Contactor Enable = On (VCL HydraulicContactorEnable bit variable = On), the VCL function Start_Pump() will turn on the pump lift contactor (Driver 3).

When HydraulicContactorEnable = On and neither Start_Pump() nor Stop_Pump() has been called, the pump lift contactor is Off.

Syntax: **Start_Pump()**

Parameters: None.

Returns:

0 – Pump start function not enabled.

1 – Pump start function successfully enabled.

Error Codes: None.

Example: Start_Pump()

This will turn on the pump lift contactor.

STOP_PUMP()

This function is used to turn off the pump lift contactor (pin 4). If the parameter Drivers » Driver3 » Contactor Enable = On (VCL HydraulicContactorEnable bit variable = On), the VCL function Start_Pump() will turn off the pump lift contactor (Driver 3).

When HydraulicContactorEnable = On and neither Start_Pump() nor Stop_Pump() has been called, the pump lift contactor is Off.

Syntax: **Stop_Pump()**

Parameters: None.

Returns:

0 – Pump stop function not enabled.

1 – Pump stop function successfully enabled.

Error Codes: None.

Example: Stop_Pump()

This will turn off the pump lift contactor.

AUTOMATE_FREQUENCY_OUTPUT()

This function sets up the PD Driver (pin 2) PWM output to yield a frequency proportional to the input variable at an execution rate of 16 ms. This output can be used to drive an electronic speedometer or tachometer.

An additional VCL variable called `Frequency_Output_Duty_Cycle` works in conjunction with the `Automate_Frequency_Output()` function to modify the duty cycle of the output. The default value of `Frequency_Output_Duty_Cycle` is 50% (or

Syntax: `Automate_Frequency_Output()`

Parameters:

<code>@Source</code>	Index of source value to be the input.
<code>MinInput</code>	Holds the minimum value for the input.
<code>MaxInput</code>	Holds the maximum value for the input.
<code>MinOutput</code>	Holds the minimum value for the output in Hz.
<code>MaxOutput</code>	Holds the maximum value for the output in Hz.

NOTE: The frequency range is 0–4000 Hz, with 4 Hz being the minimum active frequency, and 0–3 Hz = Off.

Returns:

- 0 – PWM not automated.
- 1 – PWM automated.

Error Codes: `Param_Range` is returned when a parameter is out of range.
`PT_Range` is returned when the Parameter Table Index is out of range.

Example: To set up the PD Driver to output 500 Hz to 1500 Hz (at duty cycle = 25%) for motor speed in the range 100 rpm to 4000 rpm:
`Frequency_Output_Duty_Cycle = 8192 ;set duty cycle = 25%`
`Automate_Frequency_Output(@ABS_Motor_Speed,100,4000,500,1500)`

OVERRIDE_EM_BRAKE_PWM()

This function is designed to override EM Brake. When this function is enabled, it resets the hill hold memory value.

Syntax: **Override_EM_Brake_PWM(Kill_Hill_Hold)**

Parameters: Kill_Hill_Hold

0 – Retain hill hold memory.

1 – Reset hill hold memory.

Returns:

0 – Failed to override EM brake.

1 – Succeeded in overriding EM brake.

Error Codes: None.

RESTORE_EM_BRAKE_PWM()

This function returns control of the EM brake PWM to the EM brake state machine.

Syntax: **Restore_EM_Brake_PWM()**

Parameters: None.

Returns:

0 – Failed.

1 – Success.

Error Codes: None.

MAP_TWO_POINTS()

This function interpolates values between two points, Y1 and Y2; based upon an X input parameter. Interpolating based on X1 and X2.

Typical Usage:

1. Calculate a value framed between two X-axis points projected across two Y points, where the functions' output is the value of the Y-axis intersection.
2. Extend the 7 pair limitation of the Setup_Map function by the use of multiple If, If Else statements on segments of an XY array.

Syntax: **Map_Two_Points(X,X1,X2,Y1,Y2)**

Parameters:

- X Input.
- X1 Input point X1.
- X2 Input point X2.
- Y1 Output point Y1.
- Y2 Output point Y2.

Returns:

- n Mapped value.

Error Codes: None.

Example: Convert a controller value, such as Steer_Angle, to a voltage. The input (X) is the Steer_Angle variable, where X1-X2 is an angle between 0 and 90 Degrees. The output is the interpolated value across Y-axis points in Volts.

Enable the expansion of the Setup_Map(16) function beyond the seven (7) point-pair by the use of multiple If, If Else statements on segments of a two point XY array. For Example:

```
If (X < 1)
{
Y = Map_2_Points(X, X1, X2, Y1, Y2)
}
Else If (0<= X <= 1)
{
Y = Map_two_Points(X, X1, X2, Y1, Y2)
}
Else If (1< X <= 2)
{
Y = Map_Two_Points(X, X1, X2, Y1, Y2)
}
Else If (2< X <= 3)
{
Y = Map_Two_Points(X, X1, X2, Y1, Y2)
}
Etc.
```

BATTERY_COMPENSATE()

This function is used to compensate a variable with battery voltage using the nominal battery setting.

Syntax: `Battery_Compensate (Input)`

Parameters:

Input Variable to be compensated.

Returns:

n Output.

Error Codes: None.

Updated since os32.0, a string length argument has been added to the *Put_Message_To_String()* function. This allows the function to create strings up to 64 characters (the previous limit was 12).

PUT_MESSAGE_TO_STRING()

Use this function to construct a message and parse it into a list of ASCII character equivalents. The output formats an ASCII message with pre-text on the left, a variable's value in the middle, and post-text on the right.

The variable's value can have a decimal place added using the NFormat specifier.

If the resulting message is longer than 64 characters, truncation occurs,

- The variable's value is truncated.
- The post-text is truncated second.
- The pre-text is truncated last.
- All truncation occurs on the right of a section.
- All unused array positions will be filled with a space (ASCII 32).

Syntax: `Put_Message_To_String(PreText, Variable, PostText, NFormat, @ArrayStart, MsgLength)`

Parameters:

Pretext String to concatenate on the left.

Variable Numeric value to convert to ASCII and concatenate in the middle.

PostText String to concatenate on the right.

NFormat Numerical format specifier (see NFormat specifiers, below and in Sys Info).

@ArrayStart Pointer to the User or Autouser variable that is the start of the output array.

Msglength Number of characters in the string (up to 64). This is a new argument.

Set this to 12 to maintain backwards compatibility with VCL written for previous OS's.

Returns:

- 1 Invalid string format.
- 0 If message NOT output.
- 1 If the message was generated successfully.
- 2 If the message was generated successfully, but the numeric value was truncated.

Error Codes: None.

Example: LEFT-MOTOR 1234.5 RPM [Msglength = 21]

If the variable = 12345, the pre-text is "LEFT-MOTOR", the post-text is "RPM", the decimal format is PSM_DECIMAL_1 and the ArrayStart is User101.

Then set User101 = 76 "L" User102 = 69 "E" User103 = 70 "F"
 User104 = 84 "T" User105 = 45 "-" User106 = 77 "M" User107
 = 79 "0" User108 = 84 "T" User109 = 79 "O" User110 = 82
 "R" User111 = 32 " "(space) User112 = 49 "1" User113 =
 50 "2" User114 = 51 "3" User115 = 52 "4" User116 = 46 "."
 User117 = 53 "5" User118 = 32 " "(space) User119 = 82 "R"
 User120 = 80 "P" User121 = 77 "M"

NFormat Numerical format specifier

PSM_DECIMAL = Decimal, no decimal point.

PSM_DECIMAL_0 = Decimal, decimal point at far right.

PSM_DECIMAL_1 = Decimal, decimal point 1st digit from far right.

PSM_DECIMAL_2 = Decimal, decimal point 2nd digit from far right.

PSM_DECIMAL_3 = Decimal, decimal point 3rd digit from far right.

PSM_BINARY, = Binary.

PSM_HEX, = Hexadecimal.

PSM_TEXT_ONLY = Do not use the value of the variable in the message.

AUTOMATE_PWM_INPUT()

Configures a switch input to measure the duty cycle of a PWM signal applied to the input.

Syntax: `Automate_PWM_Input(sw#, Period_Ticks)`

Parameters:

- SW#** A number 1 – 16 indicating which switch input of SW_1 to SW_16 to measure the duty cycle.
- Period_Ticks** The *Automate_PWM_Input* function runs at a maximum 8ms rate in the OS. This value is the expected period of the input wavefore/8ms per measurement.
For example, if the PWM signal to be measured has a period of 200ms, the set the *Period_Ticks* to $200/8 = 25$.
Set to 0 to disable the *Automate_PWM_Input* function.

Returns:

- 0 Function did not successfully execute.
- 1 Function successfully executed.

Error Codes:

BAD_ID Switch # out of range.

The outputs of these functions are as follows:

Pwm_Input – The duty cycle of the PWM input. (0-32767 = 0-100% PWM).

SETUP_TIMER_LONG()

Allows a 32-bit variable to be used to set the value of a timer with a single function call.

Syntax: `Setup_Timer_Long(TMR#, TIME)`

Parameters:

- TMR#** ID of Timer to initialize (TMR1, TMR2, or TMR3).
- TIME** Reference to 32bit number, time in 0.1hr increments.

Returns:

- 0 Function did not successfully execute.
- 1 Function successfully executed.

Error Codes:

BAD_ID TMR# out of range.

PARAM_RANGE Parameter is out of range.

SETUP_MAILBOX_MASK()

Sets up a mask to allow messages with multiple COB ID's to be received by the same mailbox. For each bit in the mask, a “1” indicates that the bit in the incoming message must match the bit in the COB ID setup in the corresponding *Setup_Mailbox()* call. A “0” means that messages will be received into the mailbox regardless of the state of the bit in the COB ID of messages on the bus. This function must be called before *Setup_Mailbox()*.

Syntax: **Setup_Mailbox_Mask(CAN#, Mask)**

Parameters:

CAN# The mailbox ID (e.g., CAN3).
Mask 11 bits for mask.

Returns:

0 Mask not setup.
 1 Mask setup.

Error Codes:

BAD_ID CAN# out of range.

SETUP_MAILBOX_EXTENDED_MASK()

Sets up a mask to allow messages with multiple 29-bit COB ID's to be received by the same mailbox. For each bit in the mask, a “1” indicates that the bit in the incoming message must match the bit in the COB ID setup in the corresponding *Setup_Mailbox_Extended_ID()* call. A “0” means that messages will be received into the mailbox regardless of the state of the bit in the COB ID of messages on the bus. This function must be called before *Setup_Mailbox_Extended_ID()*.

Syntax: **Setup_Mailbox_Extended_Mask(CAN#, Mask_High, Mask_Low)**

Parameters:

CAN# The mailbox ID (e.g., CAN3).
Mask-High High 13 bits of mask.
Mask_Low Low 16 bits of mask.

Returns:

0 Mask not setup.
 1 Mask setup.

Error Codes:

BAD_ID CAN# out of range.

AUTOMATE_ENCODER()

Automate_Encoder() allows processing of low bandwidth Sin/Cos signals for use in VCL. This function runs in the 4 ms queue.

Syntax: `Automate_Encoder(ENC#, @ChannelA, @ChannelB, V_Min_A, V_Max_A, V_Min_B, V_Max_B, Encoder_Type)`

Parameters:

ENC#	ENC1 or ENC2 (Holds the ID of the desired encoder block).
@ChannelA	Pointer to source variable 1 (-32767 - 32767).
@ChannelB	Pointer to source variable 2 (-32767 - 32767).
V_Min_A	Minimum Voltage of encoder channel A (-32767 - 32767).
V_Max_A	Maximum Voltage of encoder channel A (-32767 - 32767).
V_Min_B	Minimum Voltage of encoder channel B (-32767 - 32767).
V_Max_B	Maximum Voltage of encoder channel B (-32767 - 32767).
Encoder_Type	Only Sin/Cos encoders are supported, this should be set to 0.

Returns:

- 0 Function did not successfully execute.
- 1 Function successfully executed.

Error Codes:

BAD_ID	ENC# out of Range.
PT_RANGE	Invalid source or destination parameter.

The outputs of these functions are as follows:

ENC1_rev_output – ENC1's output in revolutions
(-32767 - 32767)

ENC1_Angle – ENC1's output angle in degrees (0 - 4096 = 0 - 360 degrees)

ENC2_rev_output – ENC2's output in revolutions
(-32767 - 32767)

ENC2_Angle – ENC2's output angle in degrees (0 - 4096 = 0 - 360 degrees)

Example:

```
Return = Automate_Encoder(ENC#, @ChannelA, @ChannelB, V_Min_A, V_Max_A, V_Min_B, V_Max_B, Encoder_Type)
```

11 — DIAGNOSTICS AND TROUBLESHOOTING

These controllers detect a wide variety of faults or error conditions. Faults can be detected by the operating system or by the VCL code. This chapter describes the faults detected by the operating system.

Faults detected by VCL code (faults 51–67 in Table 6) cannot be defined here as they will vary from application to application. Refer to the appropriate OEM documentation for information on these faults.

DIAGNOSTICS

Diagnostics information can be obtained in either of two ways: (1) by reading the display on a 1313 handheld or 1314 PC programmer or (2) by observing the fault codes issued by the Status LEDs. See Table 5 for a summary of LED display formats.

The 1313 /1314 programmer will display 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 1313/1314 displays the faults by name.

The pair of LEDs built into the controller (one red, one yellow) produce flash codes displaying all the currently set faults in a repeating cycle. Each code consists of two digits. The red LED flashes once to indicate that the first digit of the code will follow; the yellow LED then flashes the appropriate number of times for the first digit. The red LED flashes twice to indicate that the second digit of the code will follow; the yellow LED flashes the appropriate number of times for the second digit.

Example:

B+ Undervoltage Cutback (code 23).

In the Fault menu of the 1313/1314 programmer, the words B+ Undervoltage Cutback will be displayed; the real-time battery voltage is displayed in the Monitor menu (“Capacitor Voltage”).

The controller’s two LEDs will display this repeating pattern:

Red	Yellow	Red	Yellow
*	* *	* *	* * *
(first digit)	(2)	(second digit)	(3)

The numerical codes used by the yellow LED are listed in the troubleshooting chart (Table 6), which also lists possible fault causes and describes the conditions that set and clear each fault.

Identifying Prototype Software

In recent years, many global functional safety standards have changed to include requirements for formal verification/validation of software for production vehicles. Curtis has included a feature in the software to allow both Curtis internal software development teams and **OEM development engineers** to make it clear to someone working with a prototype vehicle that the software is not fully verified. If a prototype OS coming from Curtis is not fully verified, the LED blink pattern when no faults are present will change to a fast flashing red LED rather than the standard slow yellow flashing LED.

In addition, there is now a VCL variable called *VCL_Prototype_Software*. Setting this variable to a non-zero value in the initialization section of a VCL program will cause the same fast flashing red LED

blink pattern as a prototype OS. This will allow a VCL developer to indicate to someone working with a prototype vehicle that the VCL software has not been fully verified.

Curtis recommends that OEM engineers, or anyone modifying the VCL, whether on a new/development vehicle or an existing vehicle, implement this fast flashing red LED until the VCL code and the vehicle is fully tested and verified.

Summary of LED Display Formats

The two LEDs have four different display modes, indicating the type of information they are providing.

Table 5 Types of LED Display

Display	Status
Neither LED illuminated	Controller is not powered on; or vehicle has dead battery; or severe damage.
Yellow LED flashing	Controller is operating normally.
Yellow and red LEDs both on solid	Controller is in Flash program mode.
Red LED on solid	Internal hardware fault detected by the Supervisor or Primary microprocessor. Missing or corrupt software. Interrupting a software download may cause corruption of the software. Cycle KSI to clear. Reload software or replace controller if necessary.
Red LED and yellow LED flashing alternately	Controller has detected a fault. 2-digit code flashed by yellow LED identifies the specific fault; one or two flashes by red LED indicate whether first or second code digit will follow.

TROUBLESHOOTING

The troubleshooting chart, Table 6, provides the following information on all the controller faults:

- fault code.
- fault name as displayed on the programmer's LCD.
- the effect of the fault.
- possible causes of the fault.
- fault set conditions.
- fault clear conditions.

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.

The fault actions (effect of fault) in Table 6 use the same bit-structure as the User_Fault_Action_xx listed in the OEM-defined User Faults. The variable System_Action (available in WinVCL Monitor or TACT) returns the decimal number corresponding to the active fault action bit(s). The Dual Drive fault actions, Bits 12–15, are listed in the supplemental Dual Drive Operation manual.

System_Action Bit	Action
No defined System_Action	Effect of fault as noted in Table 6.
Bit0 = ShutdownMotor	Disable the motor.
Bit1 = ShutdownMainContactor	Shut down the main contactor (only if Main Enable = On).
Bit2 = ShutdownEMBrake	Shut down the EM brake (only if EM Brake Disable Upon Fault = On).
Bit3 = ShutdownThrottle	Set the Throttle_Command = 0%.
Bit4 = ShutdownInterlock	Set the Interlock_State = Off.
Bit5 = ShutdownDriver1	Shut down Driver1.
Bit6 = ShutdownDriver2	Shut down Driver2.
Bit7 = ShutdownDriver3	Shut down Driver3.
Bit8 = ShutdownDriver4	Shut down Driver4.
Bit9 = ShutdownPD	Shut down Proportional Driver.
Bit10 = FullBrake	Set the Brake_Command = 100%.
Bit11 = [Reserved]	N/A (for 1232–1238E/SE, and 1239E controllers).
Bit12 = TrimDisable	Disable Dual Drive trim calculation.
Bit13 = SevereDual	For Dual Drive system, one controller has a severe fault but the main contactor must stay closed so the other controller can continue to operate.
Bit14 = ShutdownSteer	Steer angle = 0° (DD applicable).
Bit15 = LOSDual	For Dual Drive system, set the max speed to Dual_LOS_Max_Speed parameter for operation in Limited Operating Strategy.

Example:

HPD/Sequencing Fault, Flash Code 47.

Effect of fault = Shutdown Throttle.

System_Action = 8 (equates to Bit3 set: 0000 0000 0000 1000).

Changes to the fault handler and main bridge operation with PMAC motors

In non-1239E models the controller will activate the main power bridge automatically when the overvoltage point is reached, the controller will then command zero torque producing current, and command only enough negative flux producing current to keep the capacitor voltage below the overvoltage threshold. This will occur even if there is no power applied to KSI.

The Shutdown motor fault action has been modified for safe operation with a PMAC motor. If one of the following faults is declared, the bridge will open unless the Severe Overvoltage threshold is exceeded at which time the bridge will short the motor until the capacitor voltage falls below the undervoltage threshold:

- Controller Overcurrent (fault code 12).
- Current Sensor Fault (fault code 13).
- Severe KSI Undervoltage (fault code 17).
- Severe B+ Overvoltage (fault code 18).
- Sin/Cos Sensor Fault (fault code 36).
- Motor Open (fault code 37).
- OS General (fault code 71).
- Stall Detected (fault code 73).
- Supervisor Fault (fault code 77).
- Driver Supply (fault code 83).
- Sin/Cos Misalignment Fault (fault code 88).

For any other fault with a ShutdownMotor fault action, the controller will command zero torque producing current, and command only enough negative flux producing current to keep the capacitor voltage below the overvoltage threshold. If no flux producing current is necessary for 64 ms, the bridge will be disabled.

Table 6 TROUBLESHOOTING CHART

CODE (status#,bit)	PROGRAMMER LCD DISPLAY EFFECT OF FAULT	POSSIBLE CAUSE	SET / CLEAR CONDITIONS
12 (Status4,3)	Controller Overcurrent <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	1. External short of phase U,V, or W motor connections. 2. Motor parameters are mistuned. 3. Controller defective. 4. Speed encoder noise problems.	<i>Set:</i> Phase current exceeded the current measurement limit. <i>Clear:</i> Cycle KSI or perform a VCL reset.
13 (Status4,4)	Current Sensor Fault <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	1. Leakage to vehicle frame from phase U, V, or W (short in motor stator). 2. Controller defective.	<i>Set:</i> Controller current sensors have invalid offset reading. <i>Clear:</i> Cycle KSI or perform a VCL reset.
14 (Status4,0)	Precharge Failed <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	1. See Monitor menu » Battery: Capacitor Voltage. 2. External load on capacitor bank (B+ connection terminal) that prevents the capacitor bank from charging.	<i>Set:</i> The precharge failed to charge the capacitor bank. <i>Clear:</i> Cycle Interlock input or use VCL function <i>Enable_Precharge()</i> .
15 (Status2,7)	Controller Severe Undertemp <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	1. See Monitor menu » Controller: Temperature. 2. Controller is operating in an extreme environment.	<i>Set:</i> Heatsink temperature below -40°C. <i>Clear:</i> Bring heatsink temperature above -40°C, and cycle interlock or KSI.
16 (Status3,0)	Controller Severe Overtemp <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	1. See Monitor menu » Controller: Temperature. 2. Controller is operating in an extreme environment. 3. Excessive load on vehicle. 4. Improper mounting of controller.	<i>Set:</i> Heatsink temperature above +95°C. <i>Clear:</i> Bring heatsink temperature below +95°C, and cycle interlock or KSI.
17 (Status2,1)	Severe B+ Undervoltage <i>No drive torque.</i>	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: Capacitor Voltage. 6. Blown B+ fuse or main contactor did not close.	<i>Set:</i> Capacitor bank voltage dropped below the Severe Undervoltage limit (see page 25) with FET bridge enabled. <i>Clear:</i> Bring capacitor voltage above Severe Undervoltage limit.
17 (Status8,4)	Severe KSI Undervoltage <i>No action.</i>	1. See Monitor menu » Battery: Keyswitch Voltage. 2. Non-controller system drain on battery/ KSI circuit wiring. 3. KSI disconnected while driving. 4. Blown KSI fuse.	<i>Set:</i> When below Brownout Voltage for 2 seconds (see Table D-1). <i>Clear:</i> Bring KSI voltage above Brownout Voltage.
18 (Status2,2)	Severe B+ Overvoltage <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	1. See Monitor menu » Battery: Capacitor Voltage. 2. Battery parameters are misadjusted. 3. Battery resistance too high for given regen current. 4. Battery disconnected while regen braking.	<i>Set:</i> Capacitor bank voltage exceeded the Severe Overvoltage limit (see page 25) with FET bridge enabled. <i>Clear:</i> Bring capacitor voltage below Severe Overvoltage limit, and then cycle KSI.

Table 6 TROUBLESHOOTING CHART, cont'd

CODE (status#,bit)	PROGRAMMER LCD DISPLAY EFFECT OF FAULT	POSSIBLE CAUSE	SET / CLEAR CONDITIONS
19 (Status9, bit3)	PMAC Commissioning Needed Hold all three phases at 50% PWM and disables main weld and DNC faults.	The PMAC Commissioning Procedure has not yet been performed to set up the controller for a new PMAC motor.	<i>Set:</i> PMAC Commissioning Needed parameter set to 1. <i>Clear:</i> PMAC Commissioning Needed parameter set to 0.
22 (Status2,6)	Controller Overtemp Cutback <i>Reduced drive and brake torque.</i>	1. See Monitor menu » Controller: Temperature. 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.	<i>Set:</i> Heatsink temperature exceeded 85°C. <i>Clear:</i> Bring heatsink temperature below 85°C.
23 (Status2,3)	B+ Undervoltage Cutback <i>Reduced drive torque.</i>	1. Normal operation. Fault indicates the batteries need recharging. Controller is performance limited at this voltage. 2. Battery parameters are misadjusted. 3. Non-controller system drain on battery. 4. Battery resistance too high. 5. Battery disconnected while driving. 6. See Monitor menu » Battery: Capacitor Voltage. 7. Blown B+ fuse or main contactor did not close.	<i>Set:</i> Capacitor bank voltage dropped below the Undervoltage limit (see page 25) with the FET bridge enabled. <i>Clear:</i> Bring capacitor voltage above the Undervoltage limit.
24 (Status2,4)	B+ Overvoltage Cutback <i>Reduced brake torque. Note: This fault is declared only when the controller is running in regen.</i>	1. Normal operation. Fault shows that regen braking currents elevated the battery voltage during regen braking. Controller is performance limited at this voltage. 2. Battery parameters are misadjusted. 3. Battery resistance too high for given regen current. 4. Battery disconnected while regen braking. 5. See Monitor menu » Battery: Capacitor Voltage.	<i>Set:</i> Capacitor bank voltage exceeded the Overvoltage limit (see page 25) with the FET bridge enabled. <i>Clear:</i> Bring capacitor voltage below the Overvoltage limit.
25 (Status5,3)	+5V Supply Failure <i>None, unless a fault action is programmed in VCL.</i>	1. External load impedance on the +5V supply (pin 26) is too low. 2. See Monitor menu » outputs: 5 volts and Ext Supply Current.	<i>Set:</i> +5V supply (pin 26) outside the 5V±10% range. <i>Clear:</i> Bring voltage within range.
26 (Status4,1)	Digital Out 6 Overcurrent <i>Digital Output 6 driver will not turn on.</i>	1. External load impedance on Digital Output 6 driver (pin 19) is too low.	<i>Set:</i> Digital Output 6 (pin 19) current exceeded 1 Amp. <i>Clear:</i> Remedy the overcurrent cause and use the VCL function <i>Set_DigOut()</i> to turn the driver on again.
27 (Status4,2)	Digital Out 7 Overcurrent <i>Digital Output 7 driver will not turn on.</i>	1. External load impedance on Digital Output 7 driver (pin 20) is too low.	<i>Set:</i> Digital Output 7 (pin 20) current exceeded 1 Amp. <i>Clear:</i> Remedy the overcurrent cause and use the VCL function <i>Set_DigOut()</i> to turn the driver on again.

Table 6 TROUBLESHOOTING CHART, cont'd

CODE (status#,bit)	PROGRAMMER LCD DISPLAY EFFECT OF FAULT	POSSIBLE CAUSE	SET / CLEAR CONDITIONS
28 (Status4,5)	Motor Temp Hot Cutback <i>Reduced drive torque.</i>	1. Motor temperature is at or above the programmed Temperature Hot setting, and the current is being cut back. 2. Motor Temperature Control Menu parameters are mistuned. 3. See Monitor menu » Motor: Temperature and » Inputs: Analog2. 4. If the application doesn't use a motor thermistor, Temp Compensation and Temp Cutback should be programmed Off.	<i>Set:</i> Motor temperature is at or above the Temperature Hot parameter setting. <i>Clear:</i> Bring the motor temperature within range.
29 (Status5,1)	Motor Temp Sensor Fault <i>MaxSpeed reduced (LOS, Limited Operating Strategy), and motor temperature cutback disabled.</i>	1. Motor thermistor is not connected properly. 2. If the application doesn't use a motor thermistor, Motor Temp Sensor Enable should be programmed Off. 3. See Monitor menu » Motor: Temperature and » Inputs: Analog2.	<i>Set:</i> Motor thermistor input (pin 8) is at the voltage rail (0V or 10V). <i>Clear:</i> Bring the motor thermistor input voltage within range.
31 (Status3,1)	Coil1 Driver Open/Short <i>ShutdownDriver1.</i>	1. Open or short on driver load. 2. Dirty connector pins. 3. Bad crimps or faulty wiring.	<i>Set:</i> Driver 1 (pin 6) is either open or shorted. This fault can be set only when Main Enable = Off. <i>Clear:</i> Correct open or short, and cycle driver.
31 (Status3,6)	Main Open/Short <i>ShutdownMotor; ShutdownMainContactor; ShutdownEMBrake; ShutdownThrottle; FullBrake.</i>	1. Open or short on driver load. 2. Dirty connector pins. 3. Bad crimps or faulty wiring.	<i>Set:</i> Main contactor driver (pin 6) is either open or shorted. This fault can be set only when Main Enable = On. <i>Clear:</i> Correct open or short, and cycle driver.
32 (Status3,2)	Coil2 Driver Open/Short <i>ShutdownDriver2.</i>	1. Open or short on driver load. 2. Dirty connector pins. 3. Bad crimps or faulty wiring.	<i>Set:</i> Driver 2 (pin 5) is either open or shorted. This fault can be set only when EM Brake Type = 0. <i>Clear:</i> Correct open or short, and cycle driver.
32 (Status3,7)	EMBrake Open/Short <i>ShutdownEMBrake; ShutdownThrottle; FullBrake.</i>	1. Open or short on driver load. 2. Dirty connector pins. 3. Bad crimps or faulty wiring.	<i>Set:</i> Electromagnetic brake driver (pin 5) is either open or shorted. This fault can be set only when EM Brake Type >0. <i>Clear:</i> Correct open or short, and cycle driver.
33 (Status3,3)	Coil3 Driver Open/Short <i>ShutdownDriver3.</i>	1. Open or short on driver load. 2. Dirty connector pins. 3. Bad crimps or faulty wiring.	<i>Set:</i> Driver 3 (pin 4) is either open or shorted. <i>Clear:</i> Correct open or short, and cycle driver.
34 (Status3,4)	Coil4 Driver Open/Short <i>ShutdownDriver4.</i>	1. Open or short on driver load. 2. Dirty connector pins. 3. Bad crimps or faulty wiring.	<i>Set:</i> Driver 4 (pin 3) is either open or shorted. <i>Clear:</i> Correct open or short, and cycle driver.
35 (Status3,5)	PD Open/Short <i>ShutdownPD.</i>	1. Open or short on driver load. 2. Dirty connector pins. 3. Bad crimps or faulty wiring.	<i>Set:</i> Proportional driver (pin 2) is either open or shorted. <i>Clear:</i> Correct open or short, and cycle driver.
36 (Status5,6)	Encoder Fault <i>ShutdownEMBrake; Motor disabled.</i>	1. Motor encoder failure. One or both inputs. 2. Bad crimps or faulty wiring. 3. See Monitor menu » Motor: Motor RPM.	<i>Set:</i> Motor encoder phase failure detected. <i>Clear:</i> Either cycle KSI, or if parameter LOS Upon Encoder Fault= On and Interlock has been cycled, then the Encoder Fault is cleared and Encoder LOS fault (code 93) is set, allowing limited motor control.

Table 6 TROUBLESHOOTING CHART, cont'd

CODE (status#,bit)	PROGRAMMER LCD DISPLAY EFFECT OF FAULT	POSSIBLE CAUSE	SET / CLEAR CONDITIONS
37 (Status4,7)	Motor Open <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	1. Motor phase is open. 2. Bad crimps or faulty wiring.	<i>Set:</i> Motor phase U, V, or W detected open. <i>Clear:</i> Cycle KSI.
38 (Status1,0)	Main Contactor Welded <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	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).	<i>Set:</i> 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. <i>Clear:</i> Cycle KSI.
39 (Status1,1)	Main Contactor Did Not Close <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	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.	<i>Set:</i> With the main contactor commanded closed, the capacitor bank voltage (B+ connection terminal) did not charge to B+. <i>Clear:</i> 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.
41 (Status1,4)	Throttle Wiper High <i>ShutdownThrottle.</i>	1. See Monitor menu » Inputs: Throttle Pot. 2. Throttle pot wiper voltage too high.	<i>Set:</i> Throttle pot wiper (pin 16) voltage is higher than the high fault threshold (can be changed with the VCL function <i>Setup_Pot_Faults()</i>). <i>Clear:</i> Bring throttle pot wiper voltage below the fault threshold.
42 (Status1,3)	Throttle Wiper Low <i>ShutdownThrottle.</i>	1. See Monitor menu » Inputs: Throttle Pot. 2. Throttle pot wiper voltage too low.	<i>Set:</i> Throttle pot wiper (pin 16) voltage is lower than the low fault threshold (can be changed with the VCL function <i>Setup_Pot_Faults()</i>). <i>Clear:</i> Bring throttle pot wiper voltage above the fault threshold.
43 (Status1,6)	Pot2 Wiper High <i>No Action</i>	1. See Monitor menu » Inputs: Pot2 Raw. 2. Pot2 wiper voltage too high.	<i>Set:</i> Pot2 wiper (pin 17) voltage is higher than the high fault threshold (can be changed with the VCL function <i>Setup_Pot_Faults()</i>). <i>Clear:</i> Bring Pot2 wiper voltage below the fault threshold.
44 (Status1,5)	Pot2 Wiper Low <i>No Action</i>	1. See Monitor menu » Inputs: Pot2 Raw. 2. Pot2 wiper voltage too low.	<i>Set:</i> Pot2 wiper (pin 17) voltage is lower than the low fault threshold (can be changed with the VCL function <i>Setup_Pot_Faults()</i>). <i>Clear:</i> Bring Pot2 wiper voltage above the fault threshold.
45 (Status1,2)	Pot Low OverCurrent <i>No Action.</i> (Does not shut down the pot's low-current source.)	1. See Monitor menu » Outputs: Pot Low. 2. Combined pot resistance connected to pot low is too low.	<i>Set:</i> Pot low (pin 18) current exceeds 10 mA. <i>Clear:</i> Clear pot low overcurrent condition and/or cycle KSI.
46 (Status1,7)	EEPROM Failure <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>ShutdownInterlock;</i> <i>ShutdownDriver1;</i> <i>ShutdownDriver2;</i> <i>ShutdownDriver3;</i> <i>ShutdownDriver4;</i> <i>ShutdownPD;</i> <i>FullBrake.</i>	1. Failure to write to EEPROM memory. This can be caused by EEPROM memory writes initiated by VCL, by the CAN bus, by adjusting parameters with the programmer, or by loading new software into the controller.	<i>Set:</i> Controller operating system tried to write to EEPROM memory and failed. <i>Clear:</i> Download the correct software (OS) and matching parameter default settings into the controller and cycle KSI.

Table 6 TROUBLESHOOTING CHART, cont'd

CODE (status#,bit)	PROGRAMMER LCD DISPLAY EFFECT OF FAULT	POSSIBLE CAUSE	SET / CLEAR CONDITIONS
47 (Status2,0)	HPD/Sequencing Fault <i>ShutdownThrottle.</i>	1. KSI, interlock, direction, and throttle inputs applied in incorrect sequence. 2. Faulty wiring, crimps, or switches at KSI, interlock, direction, or throttle inputs. 3. See Monitor menu » Inputs.	<i>Set:</i> HPD (High Pedal Disable) or sequencing fault caused by incorrect sequence of KSI, interlock, direction, and throttle inputs. <i>Clear:</i> Reapply inputs in correct sequence.
47 (Status6,2)	Emer Rev HPD <i>ShutdownThrottle;</i> <i>ShutdownEMBrake.</i>	4. Emergency Reverse operation has concluded, but the throttle, forward and reverse inputs, and interlock have not been returned to neutral.	<i>Set:</i> At the conclusion of Emergency Reverse, the fault was set because various inputs were not returned to neutral. <i>Clear:</i> If EMR_Interlock = On, clear the interlock, throttle, and direction inputs. If EMR_Interlock = Off, clear the throttle and direction inputs.
49 (Status4,6)	Parameter Change Fault <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	1. 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 Type this fault will appear and require cycling KSI before the vehicle can operate.	<i>Set:</i> Adjustment of a parameter setting that requires cycling of KSI. <i>Clear:</i> Cycle KSI.
51-67	OEM Faults <i>(See OEM documentation.)</i>	1. These faults can be defined by the OEM and are implemented in the application-specific VCL code. See OEM documentation.	<i>Set:</i> See OEM documentation. <i>Clear:</i> See OEM documentation.
68 (Status5,2)	VCL Run Time Error <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>ShutdownInterlock;</i> <i>ShutdownDriver1;</i> <i>ShutdownDriver2;</i> <i>ShutdownDriver3;</i> <i>ShutdownDriver4;</i> <i>ShutdownPD;</i> <i>FullBrake.</i>	1. VCL code encountered a runtime VCL error. 2. See Monitor menu » Controller: VCL Error Module and VCL Error. This error can then be compared to the runtime VCL module ID and error code definitions found in the specific OS system information file.	<i>Set:</i> Runtime VCL code error condition. <i>Clear:</i> Edit VCL application software to fix this error condition; flash the new compiled software and matching parameter defaults; cycle KSI.
69 (Status5,0)	External Supply Out of Range <i>None, unless a fault action is programmed in VCL.</i>	1. External load on the 5V and 12V supplies draws either too much or too little current. 2. Fault Checking Menu parameters Ext Supply Max and Ext Supply Min are mistuned. 3. See Monitor menu » Outputs: Ext Supply Current.	<i>Set:</i> The external supply current (combined current used by the 5V supply [pin 26] and 12V supply [pin 25]) is either greater than the upper current threshold or lower than the lower current threshold. The two thresholds are defined by the External Supply Max and External Supply Min parameter settings (page 55). <i>Clear:</i> Bring the external supply current within range.
71 (Status5,4)	OS General <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>ShutdownInterlock;</i> <i>ShutdownDriver1;</i> <i>ShutdownDriver2;</i> <i>ShutdownDriver3;</i> <i>ShutdownDriver4;</i> <i>ShutdownPD;</i> <i>FullBrake.</i>	1. Internal controller fault.	<i>Set:</i> Internal controller fault detected. <i>Clear:</i> Cycle KSI.
72 (Status5,5)	PDO Timeout <i>ShutdownThrottle;</i> CAN NMT State set to Pre-operational.	1. Time between CAN PDO messages received exceeded the PDO Timeout Period.	<i>Set:</i> Time between CAN PDO messages received exceeded the PDO Timeout Period. <i>Clear:</i> Cycle KSI or receive CAN NMT message.

Table 6 TROUBLESHOOTING CHART, cont'd

CODE (status#,bit)	PROGRAMMER LCD DISPLAY EFFECT OF FAULT	POSSIBLE CAUSE	SET / CLEAR CONDITIONS
73 (Status5,7)	Stall Detected <i>ShutdownEMBrake;</i> Motor disabled; Control Mode changed to LOS (Limited Operating Strategy).	1. Stalled motor (motor current insufficient to move the vehicle, for example up a steep hill or over-loaded). 2. Motor encoder failure (no feedback received at the controller). 3. Bad crimps or faulty wiring. 4. Problems with power supply for the motor encoder. 5. See Monitor menu » Motor: Motor RPM.	<i>Set:</i> No motor encoder movement detected (no motor speed), yet controller is requesting current (throttle command). <i>Clear:</i> Either cycle KSI, or if parameter LOS Upon Encoder Fault = On and Interlock has been cycled, then the Stall Detected fault is cleared and Encoder LOS fault (code 93) is set, allowing limited motor control.
74 (Status7,6)	Fault On Other Traction Controller	1. Dual Drive fault: see Dual Drive manual.	
75 (Status7,5)	Dual Severe Fault	1. Dual Drive fault: see Dual Drive manual.	
77 (Status6,5)	Supervisor Fault <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>ShutdownInterlock;</i> <i>ShutdownDriver1;</i> <i>ShutdownDriver2;</i> <i>ShutdownDriver3;</i> <i>ShutdownDriver4;</i> <i>ShutdownPD;</i> <i>FullBrake.</i>	1. The Supervisor has detected a mismatch in redundant readings. 2. Internal damage to Supervisor microprocessor. 3. Switch inputs allowed to be within upper and lower thresholds for over 100 milliseconds (for recurring errors, check the switches for moisture).	<i>Set:</i> Mismatched redundant readings; damaged Supervisor; illegal switch inputs. <i>Clear:</i> Check for noise or voltage drift in all switch inputs; check connections; Cycle KSI or VCL reset.
78 (Status9,0)	Supervisor Incompatible <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>ShutdownInterlock;</i> <i>ShutdownDriver1;</i> <i>ShutdownDriver2;</i> <i>ShutdownDriver3;</i> <i>ShutdownDriver4;</i> <i>ShutdownPD;</i> <i>FullBrake.</i>	1. The main OS is not compatible with the Supervisor OS.	<i>Set:</i> Incompatible software. <i>Clear:</i> Load properly matched OS code or update the Supervisor code; cycle KSI.
82 (Status6,0)	Bad Calibrations <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	1. Internal controller fault.	<i>Set:</i> Internal controller fault detection. <i>Clear:</i> Cycle KSI or perform a VCL reset.
83 (Status9,6)	Driver Supply <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	1. Internal controller fault in the voltage supply for the driver circuits.	<i>Set:</i> Internal controller fault detection. <i>Clear:</i> Cycle KSI or perform a VCL reset.

Table 6 TROUBLESHOOTING CHART, cont'd

CODE (status#,bit)	PROGRAMMER LCD DISPLAY EFFECT OF FAULT	POSSIBLE CAUSE	SET / CLEAR CONDITIONS
87 (Status6,6)	Motor Characterization Fault <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	Motor characterization failed during characterization process. See Monitor menu » Controller: Motor Characterization Error for cause: 0 = Sequencing error. Normally caused by turning off Motor Characterization Test Enable before running the test. 1 = Encoder signal seen but step size not auto-detected; set up Encoder Steps manually. 2 = Motor temp sensor fault. 3 = Motor temp hot cutback fault. 4 = Controller overtemp cutback fault. 5 = Controller undertemp cutback fault. 6 = Undervoltage cutback fault. 7 = Severe overvoltage fault. 8 = Encoder signal not seen, or one or both channels missing. 9 = Motor parameters out of characterization range. 20 = Sin/Cos sensor not found. 21 = Phasing not detected. 22 = Sin/Cos sensor characterization failure. 23 = Started characterization procedure while motor rotating.	<i>Set:</i> Motor characterization failed during the motor characterization process. Normally caused by turning off Motor_Characterization_Test_Enable before running test. Needs controller reset. <i>Clear:</i> Correct fault; Cycle KSI or perform a VCL reset.
88 (Status8,7)	Encoder Pulse Count Fault <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>ShutdownInterlock;</i> <i>ShutdownDriver1;</i> <i>ShutdownDriver2;</i> <i>ShutdownDriver3;</i> <i>ShutdownDriver4;</i> <i>ShutdownPD;</i> <i>FullBrake.</i>	Encoder Steps parameter does not match the actual motor encoder.	<i>Set:</i> Detected wrong setting of the Encoder Steps parameter. <i>Clear:</i> Ensure the Encoder Steps parameter matches the actual encoder; cycle KSI.
88 (Status2, bit5)	Sin/Cos Sensor Fault <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i> — SinCos Sensor Misaligned Sin_Cos_Motor_Feedback_Fault	The relationship between the rotor magnets in a PMAC motor and the Sin/Cos sensor has shifted away from the value determined by the Commissioning process.	<i>Set:</i> The battery current significantly disagrees with the commanded torque and speed. <i>Clear:</i> Cycle KSI and Repeat the PMAC commissioning process or repair the Sin/Cos sensor.
89 (Status6,4)	Motor Type Fault <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	The <i>Motor_Type</i> parameter value is out of range for the loaded OS (e.g., os37.0).	<i>Set:</i> <i>Motor_Type</i> parameter is set to an incorrect (for the loaded OS) value. <i>Clear:</i> Set <i>Motor_Type</i> to correct value and cycle KSI.

Table 6 TROUBLESHOOTING CHART, cont'd

CODE (status#,bit)	PROGRAMMER LCD DISPLAY EFFECT OF FAULT	POSSIBLE CAUSE	SET / CLEAR CONDITIONS
91 (Status7,1)	VCL/OS Mismatch <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>ShutdownInterlock;</i> <i>ShutdownDriver1;</i> <i>ShutdownDriver2;</i> <i>ShutdownDriver3;</i> <i>ShutdownDriver4;</i> <i>ShutdownPD;</i> <i>FullBrake.</i>	1. The VCL software in the controller does not match the OS software in the controller.	<i>Set:</i> VCL and OS software do not match; when KSI cycles, a check is made to verify that they match and a fault is issued when they do not. <i>Clear:</i> Download the correct VCL and OS software into the controller.
92 (Status7,2)	EM Brake Failed to Set Position Hold is engaged when Interlock = On. Follows the Position Hold parameter settings, including the Position Hold Timeout Time.	1. Vehicle movement sensed after the EM Brake has been commanded to set. 2. EM Brake will not hold the motor from rotating.	<i>Set:</i> After the EM Brake was commanded to set and time has elapsed to allow the brake to fully engage, vehicle movement has been sensed. <i>Clear:</i> 1. Activate the Throttle (EM Brake type 2). 2. Activate the Interlock (EM Brake type 1).
93 (Status7,3)	Encoder LOS (Limited Operating Strategy) Enter LOS control mode. NO_ACTION, in LOS mode.	1. Limited Operating Strategy (LOS) control mode has been activated, as a result of either an Encoder Fault (Code 36) or a Stall Detected fault (Code 73). 2. Motor encoder failure. 3. Bad crimps or faulty wiring. 4. Vehicle has stalled.	<i>Set:</i> Encoder Fault (code 36) or Stall Detected (code 73) was activated, if parameter LOS Upon Encoder Fault = On and Interlock has been cycled, then the Encoder LOS (code 93) control mode is activated, allowing limited motor control. <i>Clear:</i> Cycle KSI or, if LOS mode was activated by the Stall Detected fault, clear by ensuring encoder senses proper operation, Motor RPM = 0, and Throttle Command = 0.
94 (Status7,4)	Emer Rev Timeout <i>ShutdownEMBrake;</i> <i>ShutdownThrottle.</i>	1. Emergency Reverse was activated and concluded because the EMR Timeout timer has expired. 2. The emergency reverse input is stuck On.	<i>Set:</i> Emergency Reverse was activated and ran until the EMR Timeout timer expired. <i>Clear:</i> Turn the emergency reverse input Off.
98 (Status7,7)	Illegal Model Number <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	1. The model number (<i>Model_Number</i>) must be 1232, 1234, 1236, 1238, 1239 or 1298 or E or SE variant. See model list in Appendix D, Table D-1. 2. Software and hardware do not match. 3. Controller defective.	<i>Set:</i> Illegal Model_Number variable; when KSI cycles, a check is made to confirm a legal Model_Number, and a fault is issued if one is not found. <i>Clear:</i> Download the appropriate software for the controller model (may require a newer version of the OS, beyond this manual's OS version).
99 (Status8,3)	Parameter Mismatch <i>ShutdownMotor;</i> <i>ShutdownMainContactor;</i> <i>ShutdownEMBrake;</i> <i>ShutdownThrottle;</i> <i>FullBrake.</i>	1. Dual drive enabled on only one controller. 2. Incorrect position feedback type chosen for motor technology in use. 3. Dual drive is enabled in torque mode. 4. Controller hardware incompatible with PMAC operation or with the selected PMAC motor. 5. This Control Mode does not support EM Brake Control. 6. This Control Mode does not support Interlock Braking. 7. Reserved. 8. Controller/motor incompatibility. PMAC motor short circuit current too high for controller. 9. This controller not compatible with released IPM PMAC software (no xconnect, and not a 1239E). 10. Can't run Restricted Operation Mode on this motor (incomplete Motor Type).	<i>Set:</i> Multiple set conditions, see: <i>Monitor » Controller » Parameter Mismatch Error.</i> <i>Clear:</i> Adjust parameters to appropriate values and cycle KSI.

12 — MAINTENANCE

There are no user serviceable parts in Curtis E/SE controllers. No attempt should be made to open, repair, or otherwise modify the controller. Doing so may damage the controller and will void the warranty.

It is recommended that the controller and connections be kept clean and dry and that the controller's fault history file be checked and cleared periodically.

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.

When working around any battery powered system, proper safety precautions should be taken. These include, but are not limited to: proper training, wearing eye protection, and avoiding loose clothing and jewelry.

CAUTION

Use the following cleaning procedure for routine maintenance. Never use a high pressure washer to clean the controller.

1. Remove power by disconnecting the battery.
2. Discharge the capacitors in the controller by connecting a load (such as a contactor coil) across the controller's B+ and B– terminals.
3. Remove any dirt or corrosion from the power and signal connector areas. The controller should be wiped clean with a moist rag. Dry it before reconnecting the battery.
4. Make sure the connections are tight. Refer to Chapter 2 for maximum tightening torque specifications for the battery and motor connections.

FAULT HISTORY

The 1313 handheld or 1314 PC programmer can be used to access the controller's fault history file. The programmer will read out all the faults the controller has experienced since the last time the fault history file was cleared. Faults such as contactor faults may be the result of loose wires; contactor wiring should be carefully checked. Faults such as overtemperature may be caused by operator habits or by overloading.

After a problem has been diagnosed and corrected, it is a good idea to clear the fault history file. This allows the controller to accumulate a new file of faults. By checking the new fault history file at a later date, a technician can readily determine whether the problem was indeed fixed.

APPENDIX A

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. Therefore, emission reduction techniques include making the battery and motor cables as short as possible. Minimize the lengths of the AMPseal connector's 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 ferrite beads at the RF noise source(s) 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 4 for the controller ESD ratings.

Improve ESD immunity by either 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. As a component, it has no function unless installed as part of a vehicle's 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 B

EN13849 COMPLIANCE

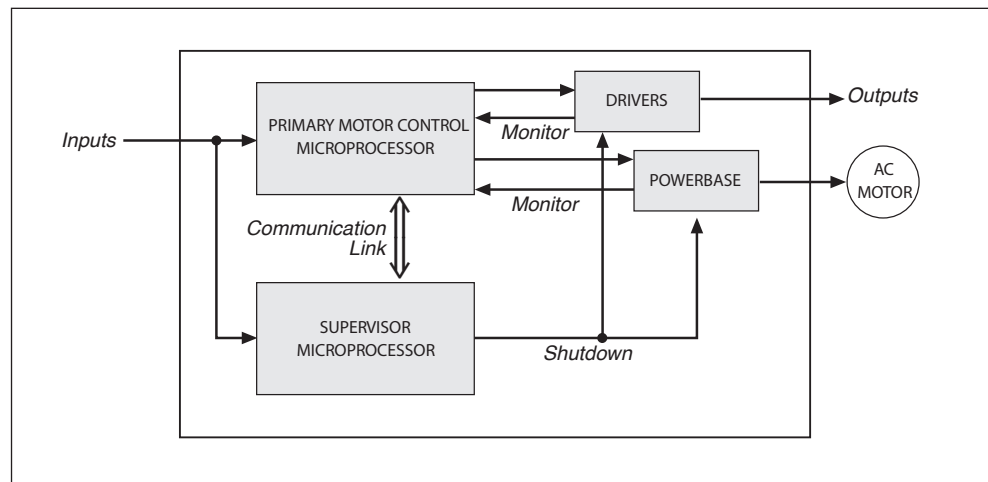
Since January 1, 2012, conformance to the European Machinery Directive has required that the Safety Related Parts of the Control System (SRPCS) be designed and verified upon the general principles outlined in EN13849. EN13849 supersedes the EN954 standard and expands upon it by requiring the determination 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). These figures are used by the OEM to calculate the overall PL for each of the safety functions of their vehicle or machine.

The OEM must determine the hazards that are applicable to their vehicle design, operation, and environment. Standards such as EN13849-1 provide guidelines that must be followed in order 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 Enhanced AC Motor Controllers comply with these directives using advanced active supervisory techniques. The basic “watchdog” test circuits have been replaced with a Supervisor microcontroller that continuously tests the safety related parts of the control system; see the simplified block diagram in Figure B-1.

Figure B-1

Supervisory system in Curtis enhanced AC motor controllers.



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 ensured 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 Enhanced AC Motor Controllers: Uncommanded Powered Motion and Motor Braking 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 start-up. The Braking Torque safety function provides detection and safe shutdown in the event of the loss of braking torque, position/hill hold, or emergency reverse.

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 safety-related performance of the 1232E controller is summarized as follows:

Safety Function	Designated Architecture	MTTFd	DC	CCF	PL
Uncommanded Powered Motion	2	>40 yrs	>90%	Pass	d
Motor Braking Torque	2	>16 yrs	>90%	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 (up to PL=d).

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.

APPENDIX C

PROGRAMMING DEVICES

Curtis programmers provide programming, diagnostic, and test capabilities for the controller. The power for operating the programmer is supplied by the host controller via a 4-pin connector. When the programmer powers up, it gathers information from the controller.

Two types of programming devices are available: the 1314 PC Programming Station and the 1313 handheld programmer. The Programming Station has the advantage of a large, easily read screen; on the other hand, the handheld programmer (with its 45×60mm 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. In addition, the Curtis 1309 Interface box with a computer-side USB and controller-side 4-pin serial Molex connection is required for the 1314 PC Programmer.

HANDHELD PROGRAMMER (1313)

The 1313 handheld programmer is functionally equivalent to the PC Programming Station; operating instructions are provided in the 1313 manual. This programmer replaces the 1311, an earlier model with fewer functions.

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 also a means to clear the fault history file.

Programming — allows the user to save/restore custom parameter settings.

Favorites — allows the user to create shortcuts to the frequently-used adjustable parameters and monitor variables (1313 only).

APPENDIX D

CONTROLLER SPECIFICATIONS

Table D-1 SPECIFICATIONS: 1232E/SE, 1234E/SE, 1236E/SE, 1238E/SE CONTROLLERS¹

Nominal input voltage: 24 V, 24–36 V, 36–48 V, 48–80 V, 72–96 V

NOMINAL VOLTAGE	MINIMUM VOLTAGE	MAXIMUM VOLTAGE	BROWNOUT VOLTAGE
24 V	14 V	30 V	12 V
24–36 V	16.8 V	50 V	15 V
36–48 V	22.2 V	60 V	20 V
48–80 V	33.6 V	105 V	20 V
72–96 V	50.4 V	120 V	30 V

PWM operating frequency: 10 kHz

Maximum encoder frequency: 10 kHz

Maximum controller output frequency: 450 Hz

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: 1238E-75XX, 1238E-76XX, and 1238SE-79XX: 5,000 hours
All other models: 20,000 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 35-pin connector header.

Weight: 1232E: 1.70 kg 1232SE: 1.92 kg
1234E: 2.84 kg 1234SE: 3.12 kg
1236E: 4.12 kg 1236SE: 3.79 kg
1238E: 6.82 kg 1238SE: 6.00 kg

Dimensions, W×L×H: 1232E: 140 × 180 × 71 mm 1232SE: 140 × 180 × 74 mm
1234E: 155 × 212 × 75 mm 1234SE: 155 × 212 × 79 mm
1236E: 165 × 232 × 85 mm 1236SE: 165 × 232 × 85 mm
1238E: 275 × 232 × 85 mm 1238SE: 275 × 232 × 85 mm

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.

¹ Controller models 123X/SE -xx7x are the only controllers compatible for PMAC applications.

Reference the PMAC motor parameter and commissioning sections for further information.

These models are not specifically listed in Table D-1.

Table D-1 SPECIFICATIONS: 1232E/SE, 1234E/SE, 1236E/SE, 1238E/SE CONTROLLERS, cont'd

MODEL CHART				
MODEL NUMBER	NOMINAL BATTERY VOLTAGE (V)	MAXIMUM CURRENT (A)	S2-60 MINUTE CONTINUOUS CURRENT (A)	CONTINUOUS CURRENT FOR DESIGN LIFE (A)
1232E-21XX	24	180	90	80
1232E-23XX	24	250	125	100
1232E-41XX	24-36	150	75	79
1232E-51XX	36-48	150	75	79
1232E-52XX	36-48	200	100	90
1232E-62XX	48-80	175	80	45
1232E-72XX	72-96	175	80	50
1234E-23XX	24	350	175	160
1234E-52XX	36-48	275	135	90
1234E-53XX	36-48	350	175	90
1234E-62XX	48-80	250	110	60
1236E-44XX	24-36	400	175	175
1236E-45XX	24-36	500	240	210
1236E-54XX	36-48	450	205	210
1236E-55XX	36-48	550	250	140
1236E-64XX	48-80	350 (1 min)	100	100
1238E-46XX	24-36	650	300	325
1238E-48XX	24-36	800	355	365
1238E-56XX	36-48	650	295	325
1238E-64XX	48-80	450	155	175
1238E-65XX	48-80	550	190	175
1238E-66XX	48-80	650 (1 min)	195	175
1238E-75XX	72-96	550	175	175
1238E-76XX	72-96	650	200	175
1232SE-24XX	24	375	185	160
1232SE-53XX	36-48	350	175	160
1232SE-63XX	48-80	250	144	125
1234SE-45XX	24-36	500	235	210
1234SE-54XX	36-48	450	215	210
1234SE-63XX	48-80	350	149	175
1236SE-46XX	24-36	650	285	220
1236SE-56XX	36-48	600	260	210
1236SE-65XX	48-80	450	185	175
1238SE-49XX	24-36	1000*	425	376
1238SE-59XX	36-48	900	395	350
1238SE-6971	48-80	900	305	300
1238SE-7971	72-96	900	305	305

Notes: All current ratings are rms values per motor phase.

S2-60 minute current is the typical current achievable before thermal cutback occurs, with an ambient temperature of 25°C and the controller mounted to a 6 mm thick vertical steel plate with 6 km/h (1.7 m/s) airflow perpendicular to the plate.

Continuous current for design life is the maximum long-term current, at an internal heatsink long-term temperature of 70°C (158°F), at which the controller will achieve its design life. These numbers are often calculated for a given application by determining an rms duty cycle current and heatsink temperature. For applications that exceed this, please contact the Curtis distributor or support engineer.

*1238SE-49XX. Please contact the Curtis distributor or support engineer regarding the duty-cycle of this controller.