1236/38 Manual

MANUAL

MODELS

1236 & 1238

AC INDUCTION MOTOR CONTROLLERS with VCL

DESIGN OF CURTIS PMC 1200 SERIES CONTROLLERS PROTECTED BY U.S. PATENT NO. 4626750.

1236/38 Manual, p/n 37022
Rev. B: October 2005

CURTIS INSTRUMENTS, INC.
200 Kisco Avenue
Mt. Kisco, New York 10509 USA
Tel. 914.666.2971
Fax 914.666.2188
www.curtisinstruments.com

CURTIS

CONTENTS

1. OVERVIEW 1

2. INSTALLATION AND WIRING...3
   - Mounting the Controller 3
   - High Current Connections and Wiring Guidelines..............6
   - Low Current Connections and Wiring Guidelines...............8
   - Controller Wiring: Basic Configuration 12
   - Switch Input Wiring 13
   - Throttle Wiring 13
   - Input/Output Specifications...................................18
     - Digital inputs 18
     - Analog inputs 19
     - Digital outputs 19
     - Analog output 19
     - High power outputs 20
     - Power supply outputs 20
     - KSI and coil return.......................................21
     - Throttle and brake inputs.................................21
     - Communications ports 22
     - Speed encoder inputs......................................22
     - Program input 22

3. PROGRAMMABLE PARAMETERS 23
   - Control Mode Select Parameter 26
   - Speed Controller Parameters *(Speed Control Mode)* 27
     - Max Speed, Kp, Ki
   - Velocity Feedforward Parameters *(Speed Control Mode)*......28
     - Kvff, Build Rate, Release Rate
   - Acceleration Feedforward Parameters *(Spd Cntrl Mode)* 29
     - Kaff, Kbff, Build Rate, Release Rate
   - Response Parameters *(Speed Control Mode)* 30
     - Full Accel Rate HS, Full Accel Rate LS, Low Accel Rate, Neutral Decel Rate HS, Neutral Decel Rate LS, Full Brake Rate HS, Full Brake Rate LS, Low Brake Rate
   - Fine Tuning Parameters *(Speed Control Mode)*....................31
     - Partial Decel Rate, HS, LS, Reversal Soften, Max Speed Accel, Max Speed Decel, Accel Release Rate
   - Pump Enable Parameter *(Speed Control Mode)*..................31
   - Speed Limiter Parameters *(Torque Control Mode)* 32
     - Max Speed, Kp, Ki, Kd
   - Response Parameters *(Torque Control Mode)*.......................33
     - Accel Rate, Accel Release Rate, Brake Rate, Brake Release Rate, Neutral Braking, Neutral Taper Speed

Fine Tuning Parameters *(Torque Control Mode)* 34
Creep Torque, Gear Soften, Brake Taper Speed,
Reversal Soften, Max Speed Decel
Restraint Parameters..36
Restraint Forward, Restraint Back, Position Hold Enable
Position Hold Parameters...36
Kp, Kp Deadband, Kd, Entry Rate
Current Limit Parameters...37
Drive Current Limit, Regen Current Limit,
Brake Current Limit
Power Limiting Map Parameters 38
Base Speed, Delta Speed
Drive Limiting Map Parameters...38
Nominal, Base Plus Delta, Base Plus 2xDelta,
Base Plus 4xDelta, Base Plus 8xDelta
Regen Limiting Map Parameters 39
Nominal, Base Plus Delta, Base Plus 2xDelta,
Base Plus 4xDelta, Base Plus 8xDelta
Throttle Parameters..40
Throttle Type, Forward Deadband, Forward Map,
Forward Max, Forward Offset, Reverse Deadband,
Reverse Map, Reverse Max, Reverse Offset,
HPD/SRO Enable, Sequencing Delay
Brake Parameters..42
Brake Pedal Enable, Brake Type, Brake Deadband,
Brake Map, Brake Max, Brake Offset
Main Contactor Parameters 43
Main Enable, Pull In PWM, Holding PWM,
Interlock Type, Open Delay, Checks Enable,
Precharge Enable
Proportional Driver Parameters..44
PD Enable, Hyd Lower Enable, PD Max Current,
PD Min Current, PD Dither %, PD Dither Period,
PD Kp, PD Ki
Fault Checking Parameters...45
Driver1–4 Checks Enable, PD Checks Enable,
EM Brake Disable Upon Fault, External Supply Max,
External Supply Min
Motor Parameters...46
Typical Max Speed, Swap Encoder Direction,
Swap Two Phases, Encoder Steps,
Encoder SW Fault Enable
Motor Temperature Parameters..47
Sensor Enable, Temperature Hot, Temperature Max,
Current Source, MotorTemp LOS Max Speed,
Sensor Type, Sensor Temp Offset
User Defined Sensor 48
Sensor 0–4, Temp 0–4

Battery Parameters 49
Nominal Voltage, User Overvoltage,
User Undervoltage, Reset Volts Per Cell,
Full Volts Per Cell, Empty Volts Per Cell,
Discharge Time, BDI Reset Percent
Vehicle Parameters 52
Metric Units, Speed to RPM, Capture Speed,
Capture Distance 1-2-3
Emergency Reverse Parameters..53
EMR Enable, EMR Type, EMR Current,
EMR Speed, EMR Accel Rate,
EMR Speed Decel Rate, EMR Torque Decel Rate
CANOpen Interface Parameters..54
CANopen Interlock, Master ID, Slave ID,
Baud Rate, Heartbeat Rate, PDO Timeout Period,
Emergency Message Rate, Suppress CANopen Init
Motor Control Tuning Parameters 55
Motor Type, Base Speed
Controller Cloning..55

4a. MONITOR MENU 56
Inputs, Outputs, Battery, Motor, Controller,
Vehicle, CAN Status

4b. CONTROLLER INFORMATION MENU 64

5. INITIAL SETUP 65

6. VEHICLE CONTROL LANGUAGE....................................71
Variable Types 72
VCL Runtime Rates...73
I/O Control with VCL 74
Interfacing the Controller's Throttle Command.................77
Interfacing the Proportional Current Driver 84
Using the Fault Handler in VCL..85
VCL Functions Specific to the 1236/38 Controller 90

7. DIAGNOSTICS AND TROUBLESHOOTING...................96

8. MAINTENANCE 104

APPENDIX A Theory of Operation, 1236/38 Controller..........A-1
APPENDIX B Vehicle Design Considerations............................B-1
APPENDIX C 1311 Programmer Operation............................. C-1
APPENDIX D Specifications, 1236/38 Controller..................... D-1

FIGURES

FIG. 1: Curtis 1236 and 1238 controllers 1
FIG. 2a: Mounting dimensions, Curtis 1236 controller 3
FIG. 2b: Mounting dimensions, Curtis 1238 controller 4
FIG. 3: Basic wiring diagram 12
FIG. 4: Wiring for Type 1 throttles 14
FIG. 5: Wiring for Type 2 throttles 15
FIG. 6: Wiring for Type 3 throttles 16
FIG. 7: Neutral braking, torque control mode.................................... 35
FIG. 8: Effect of gear soften parameter, torque control mode............. 35
FIG. 9: Effect of brake taper speed parameter, torque control mode... 35
FIG. 10: Drive current limiting map 38
FIG. 11: Regen current limiting map 39
FIG. 12: Throttle adjustment 41
FIG. 13: VCL motor command diagram... 78
FIG. 14: VCL control mode mapping 82
FIG. 15: VCL proportional driver processing 84

FIG. A-1: IFO diagram A-2
FIG. A-2: Power section topology...A-3

FIG. C-1: Curtis 1311 handheld programmer..................................... C-1

TABLES

TABLE 1: High current connections... 6
TABLE 2: Low current connections 9
TABLE 3: Programmable parameter menus 24
TABLE 4: Types of LED display 99
TABLE 5: Troubleshooting chart... 100

TABLE D-1: Specifications, 1236/38 controller D-1

1 OVERVIEW

Curtis 1236 and 1238 AC induction motor controllers with VCL deliver smooth power unlike any previous vehicle control system. The 1236 and 1238 provide unprecedented flexibility and power through inclusion of a field-programmable logic controller embedded in a state-of-the-art motor controller.

The embedded logic controller uses an innovative software programming language developed by Curtis, called Vehicle Control Language (VCL). Many electric vehicle functions are uniquely built into the VCL code, and additional functions can be added by downloading custom application code. VCL opens new avenues of customization, product differentiation, and response to the market.

The CAN bus communications included in the 1236 and 1238, as well as in many other Curtis products, allow these AC induction motor controllers to be part of an efficient distributed system. Inputs and outputs can be optimally shared throughout the system, minimizing wiring and creating integrated functions that often reduce the cost of the system.

Curtis 1236 and 1238 controllers are the ideal solution for traction, hoist, dual drive, and other motor drive and vehicle control needs. Applications include reach trucks, order pickers, stackers, large pallet movers, boom and scissor lifts, personnel carriers, light on-road, tug, counterbalance trucks, and other industrial vehicles.

Fig. 1 *Curtis 1236 and 1238 AC induction motor controllers with VCL. The more powerful 1238 is 110 mm wider. Both models have the same standard features.*

Like all Curtis controllers, the 1236 and 1238 offer superior operator control of motor drive performance. **Features include:**

✓ High efficiency, field-oriented motor control algorithms*

✓ Advanced Pulse Width Modulation technology for efficient use of battery voltage, low motor harmonics, low torque ripple, and minimized switching losses

✓ Extremely wide torque/speed range including full regeneration capability

✓ Smooth low speed control, including zero speed

✓ Adaptation of control algorithm to motor temperature variation so optimal performance is maintained under widely varying conditions

More Features ☞

- ✓ Real-time battery current, motor torque, and power estimates available
- ✓ Power limiting maps allow performance customization for reduced motor heating and consistent performance over varying battery state-of-charge
- ✓ Powerful operating system allows parallel processing of vehicle control tasks, motor control tasks, and user configurable programmable logic
- ✓ A wide range of I/O can be applied wherever needed, for maximum destributed system control
- ✓ Internal battery-state-of-charge, hourmeter, and maintenance timers
- ✓ High frequency silent operation
- ✓ Models available for 24V to 80V battery systems, with 300A RMS to 650A RMS 2-minute current ratings
- ✓ Easily programmable through the Curtis 1311 handheld programmer and 1314 PC Programming Station
- ✓ CANopen communications for integration into distributed control systems; other 11-bit identifier field CAN protocols can be custom configured through VCL
- ✓ Field-programmable, with flash downloadable main operating code
- ✓ Thermal cutback, warning, and automatic shutdown provide protection to motor and controller
- ✓ Rugged sealed housing and connectors meet IP65 environmental sealing standards for use in harsh environments
- ✓ Insulated metal substrate power base provides superior heat transfer for increased reliability.

Familiarity with your Curtis 1236/38 controller will help you install and operate it properly. We encourage you to read this manual carefully. If you have questions, please contact the Curtis office nearest you.

---

* For more information on the 1236/38 controller's motor control algorithms and 3-phase power section implementation, see Appendix A: Theory of Operation.

☞ Using the 1311 handheld programmer, you can set up the 1236/38 controller to perform all the basic operations—such as acceleration control, throttle shaping, and HPD. In this manual, we first show you how to wire your system and adjust its performance characteristics without the use of VCL. Then, in Section 6, we show you how to adjust the system using VCL, which interacts with a second, independent software realm resident in a powerful logic controller embedded within the 1236/38 controller.

2

INSTALLATION AND WIRING

MOUNTING THE CONTROLLER

The outline and mounting hole dimensions for the 1236 and 1238 controllers are shown in Figures 2a and 2b. The controller meets the IP65 requirements for environmental protection against dust and water. Nevertheless, in order to prevent external corrosion and leakage paths from developing, **the mounting location should be carefully chosen to keep the controller as clean and dry as possible**.

It is recommended that the controller be fastened to a clean, flat metal surface with four 6mm (1/4”) diameter bolts, using the holes provided. A thermal joint compound can be used to improve heat conduction from the controller heatsink to the mounting surface. Additional heatsinking or fan cooling may be necessary to meet the desired continuous ratings.

Fig. 2a *Mounting dimensions, Curtis 1236 motor controller.*

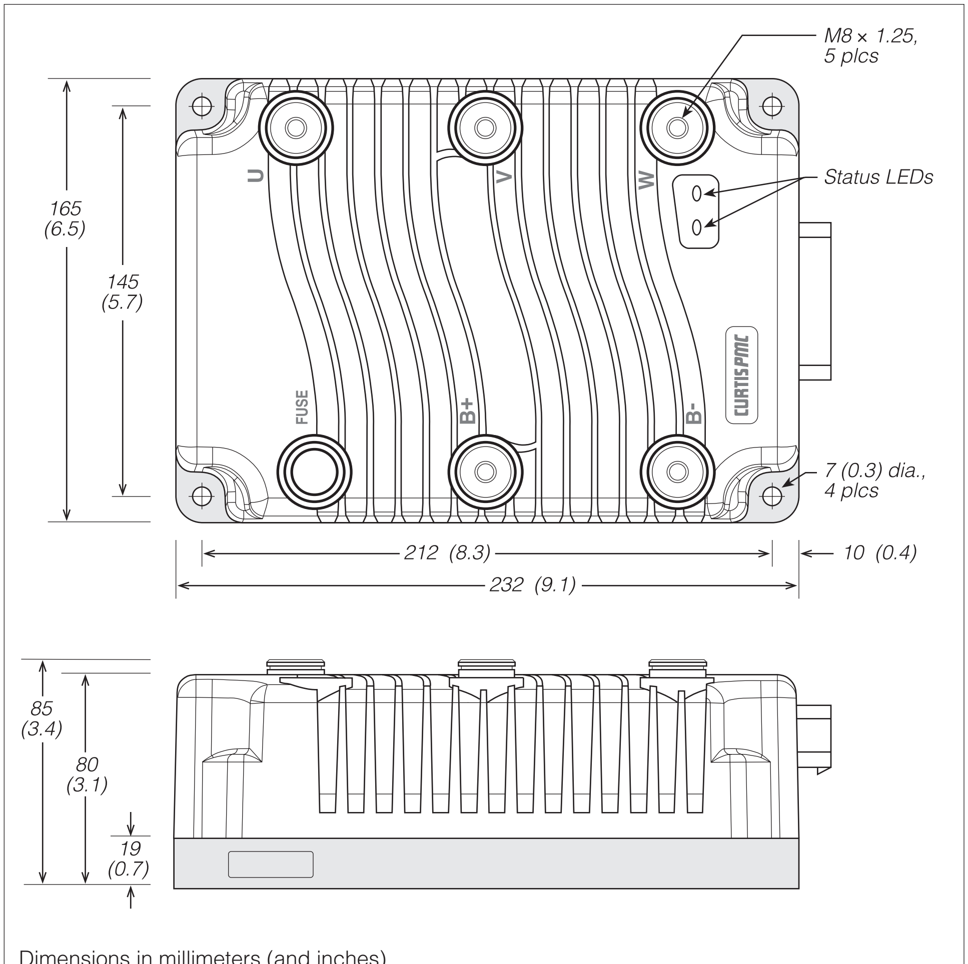
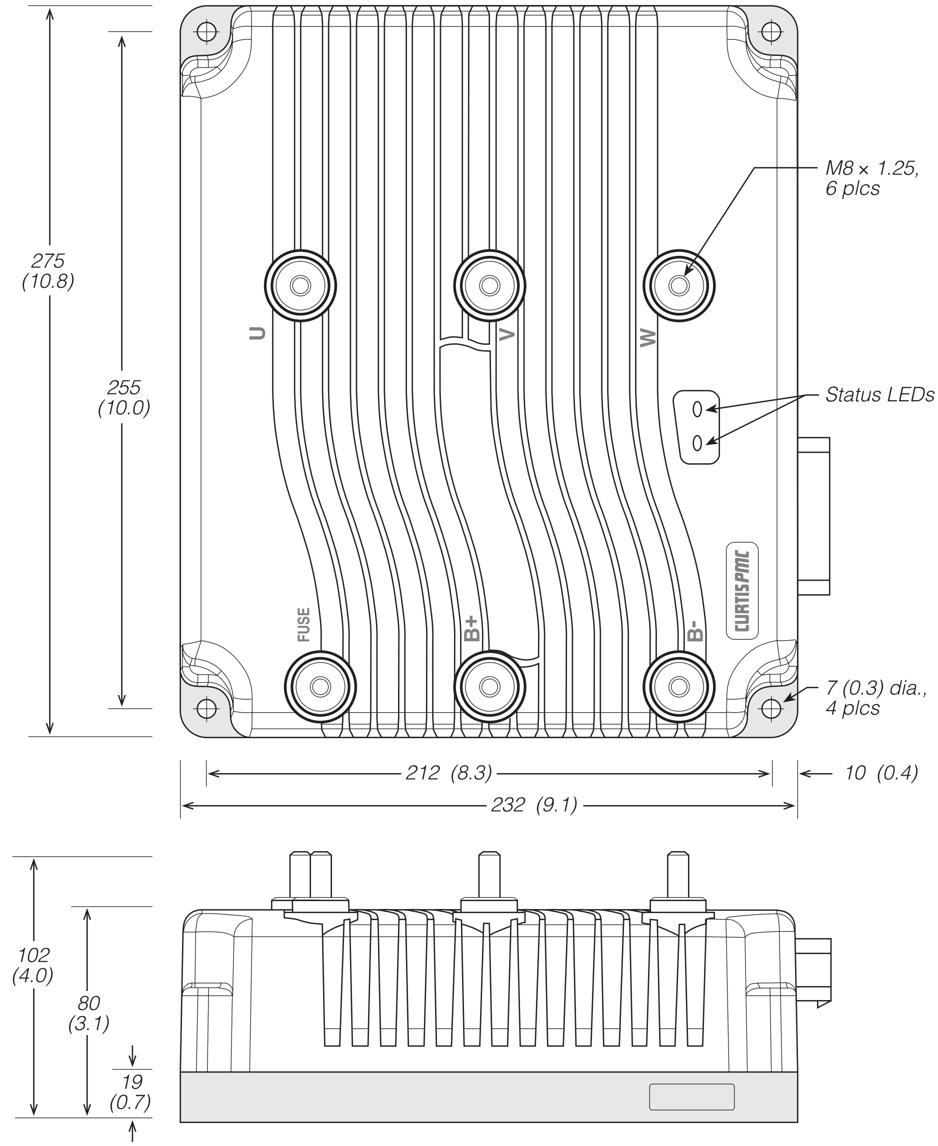


Fig. 2b *Mounting dimensions, Curtis 1238 motor controller.*



Dimensions in millimeters (and inches)

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

The 1236 and 1238 controllers contain **ESD-sensitive components**. Use appropriate precautions in connecting, disconnecting, and handling the controller. See installation suggestions in Appendix B for protecting the controller from ESD damage.

CAUTION

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

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

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

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

HIGH CURRENT CONNECTIONS

There are five high-current terminals, identified on the controller housing as **B+**, **B-**, **U**, **V**, and **W**. The terminal specifications are provided in Table 1. Electrically, the terminals are the same on the 1236 and 1238. The connections, however, are different.

Table 1 High Current Connections

TERMINAL	TYPE	FUNCTION
B+	Input	Positive battery to controller.
B-	Input	Negative battery to controller.
U	I/O	Motor phase U.
V	I/O	Motor phase V.
W	I/O	Motor phase W.

1236 Connections

Five brass M8 female studs are provided on the 1236 for the high current connections. Lugs should be installed as follows, using M8 bolts:

- Place the lug on top of the brass stud, followed by a high-load 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 stud, stack them so the lug carrying the least current is on top.
- Tighten the assembly to 9.6 ±0.9 N·m (85 ±8 in-lbs).

Note: The female studs may rotate up to ±5° in the cover.

1238 Connections

Six brass M8 studs are provided on the 1238 for the high current connections. In addition to the five standard connections, there is a convenience terminal for adding a fuse between the battery and B+; it has no internal connection.

Lugs should be installed as follows:

- Place the lug on top of the brass nut, followed by a washer and then another nut.
- If two lugs are used on the same stud, stack them so the lug carrying the least current is on top.
- Tighten the assembly to 9.6 ±0.9 N·m (85 ±8 in-lbs).

High current wiring recommendations

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.

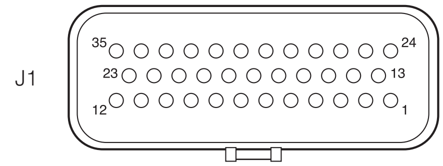
Motor wiring (**U**, **V**, **W**)

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 best 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-** stud at the controller. Typical installations will readily pass the emissions standards without a shield. Low current signal wires should not be run next to the motor cables. When necessary they should cross the motor cables at a right angle to minimize noise coupling.

LOW CURRENT 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 contact pins are Amp P/N 770520-3. The connector will accept 20 to 16 AWG wire with a 1.7 to 2.7 mm diameter thin-wall insulation.

The 35 individual pins are characterized in Table 2.



Low current wiring recommendations

Motor encoder (Pins 31, 32)

All four encoder wires 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.

CAN bus (Pins 21, 23, 34, 35)

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

All other low current wiring

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

Table 2 Low Current Connections

PIN	NAME	DESCRIPTION	RELATED VCL*	
			FUNCTIONS	REFERENCES
1	KSI	Keyswitch input. Provides logic power for the controller and power for the coil drivers.	Setup_BDI	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	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.	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	Sw_9 PWM1 PWM1_Output
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
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

* The related VCL columns are vital when writing VCL code (see Section 6). VCL "functions" are used to access the various I/Os; VCL "references" are predefined names for specific pins.

Table 2 Low Current Connections, cont'd

PIN	NAME	DESCRIPTION	RELATED VCL	
			FUNCTIONS	REFERENCES
12	Switch 6	Generic switch input #6.		Sw_6
13	Coil Return	This is the coil return pin for all the contactor coils.		
14	Program	This is a switch input to manually put the controller into flash program mode.		
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	Brake Pot Wiper	Pot wiper connection for the brake pot.	Setup_Pot Setup_Pot_Faults	Brake_Pot Brake_Pot_Output
18	Pot Low	Common pot low connection for the throttle and brake pots.		Pot_Low_Output
19	Digital Out 6	A low power open collector digital output. Can also be used as a digital input.	Set_DigOut Clear_DigOut	Sw_14 DigOut6 Dig6_Output
20	Digital Out 7	A low power open collector digital output. 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	CANH	CAN bus 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	Brake Pot High	Pot high connection for a 3-wire brake pot.		

Table 2 Low Current Connections, cont'd

PIN	NAME	DESCRIPTION	RELATED VCL FUNCTIONS	RELATED VCL REFERENCES
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 0–10V analog output.	Automate_PWM Put_PWM	PWM6 Analog_Output
31	Encoder A	Quadrature encoder input phase A.		Motor_RPM
32	Encoder B	Quadrature encoder input phase B.		Motor_RPM
33	Switch 8	Generic switch input #8. Typically used as the Reverse switch.		Sw_8
34	CAN Term L	Low connection for the CAN bus termination jumper.		
35	CANL	CAN bus low.	Setup_CAN Setup_Mailbox Send_Mailbox etc.	

CONTROLLER WIRING: BASIC CONFIGURATION

A basic wiring diagram for the 1236/38 controller is shown in Figure 3 + Thro and brake are shown in the diagram as 3-wire potentiometers; other ty throttle and brake inputs are easily accommodated) and are discuss following throttle wiring section +

The main contactor coil must be wired directly to the controller as sho in Figure 3 to meet EEC safety re n uirements + The controller can be progra

Fig. 3 *Basic wiring diagram, Curtis 1236/38 motor controller.*

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 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 your system. The 1236/38 controller has very flexible I/O and wiring configurations; you may wish to contact your local Curtis representative to discuss your 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 (see page 53).
Switch 3:	Interlock input if Interlock Type = 0 (see page 43).
Switch 7:	Forward input if Throttle Type = 1–3 (see page 40).
Switch 8:	Reverse input if Throttle Type = 1–3 (see page 40).

THROTTLE WIRING

In this manual, the term *throttle* is used in two senses: as another name for the drive throttle, and 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 braking or for acceleration.

Various throttles can be used with the 1236/38 controller. They are characterized as one of five types in the programming menu of the 1311 programmer.

Type 1:	2-wire 5kΩ–0 potentiometers
Type 2:	single-ended 0–5V throttles, current source throttles, 3-wire potentiometers, and electronic throttles
Type 3:	2-wire 0–5kΩ potentiometers
Type 4:	wigwag 0–5V throttles and 3-wire potentiometers
Type 5:	VCL input (VCL_Throttle or VCL_Brake)

The 1236/38's two throttle inputs (drive throttle, brake throttle) are programmed independently.

For potentiometers, the 1236/38 provides complete throttle fault protection that meets all applicable EEC regulations. For voltage throttles, the 1236/38 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 13). Throttle types 4 and 5 do not use the forward and reverse inputs.

Wiring for the most common throttles is described in the following text and shown in the accompanying illustrations. If a throttle you are planning to use is not covered, contact the Curtis office nearest you.

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.

Fig. 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). 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 (B-).

Throttle Type 2

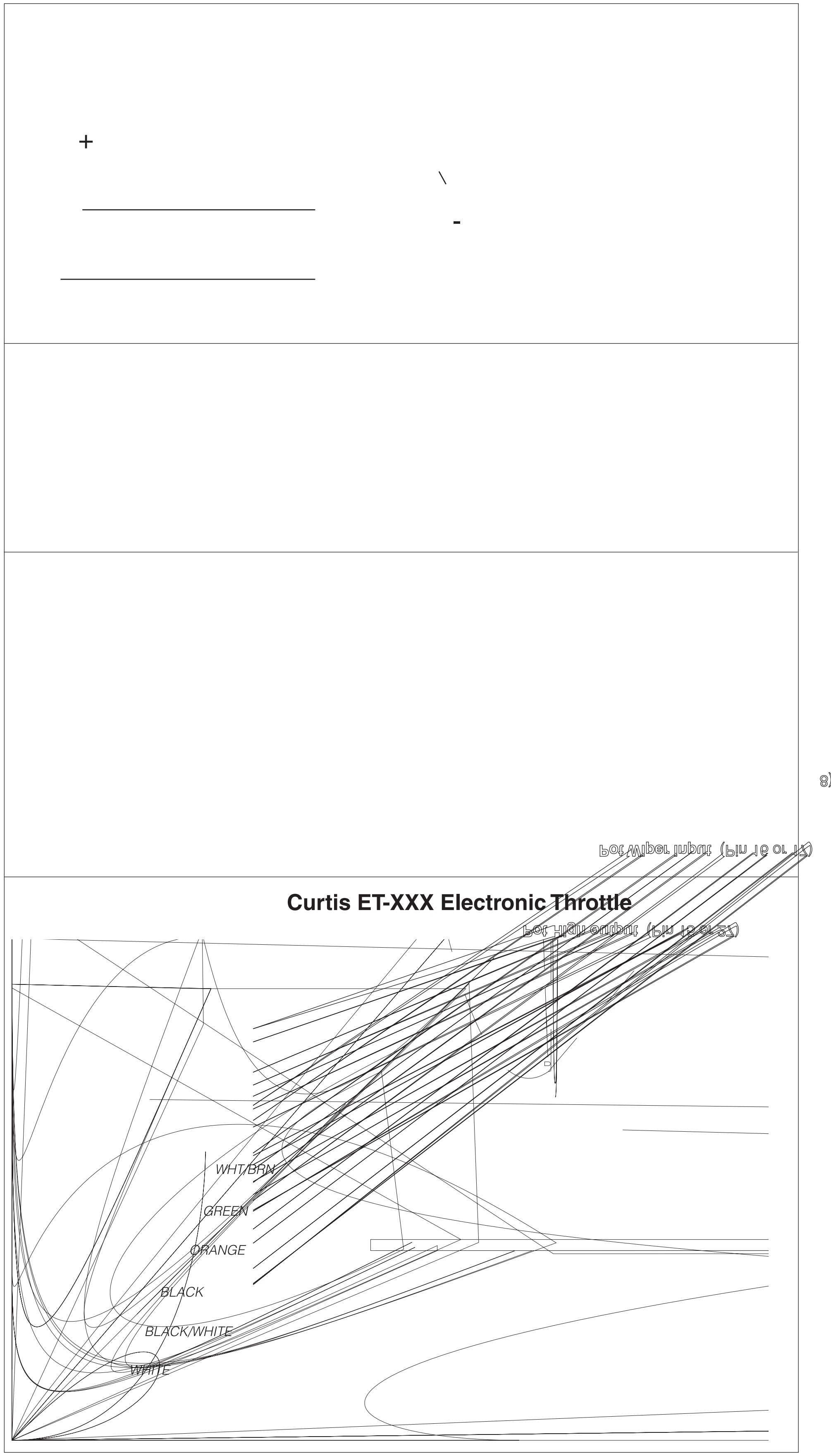
With these throttles, the controller looks for a voltage signal at the wiper input. Zero throttle request corresponds to 0 V and full throttle request to 5 V. A variety of devices can be used with this throttle input type, including voltage sources, current sources, 3-wire pots, and electronic throttles. 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

Fig. 5 *Wiring for Type 2 throttles.*



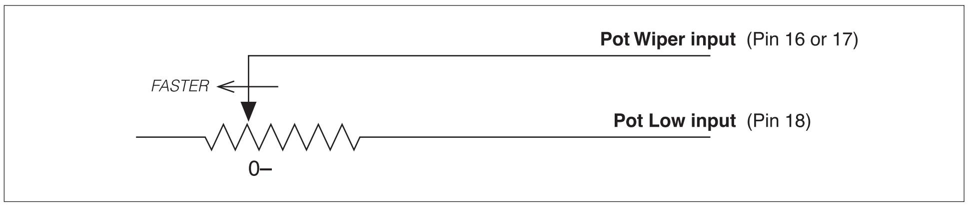
divider mode, with the controller providing the voltage source and return. Pot High provides a current limited 5V source to the pot, and Pot Low provides the return path. This is the throttle shown in the basic wiring diagram (Figure 3) for the drive throttle and for the brake throttle.

The ET-XXX electronic throttle is typically used only as a drive throttle. The ET-XXX 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.

Throttle Type 3

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

Fig. 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). 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 (B-).

Throttle Type 4

Type 4 throttles operate in wigwag style, and are appropriate only for the drive throttle. 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 Type 2 throttles; see Figure 5. The neutral point will be with the wiper at 2.5 V, measured between pot wiper input (pin 16) and I/O ground return (pin 7). The controller will provide increasing forward speed as the wiper input value is increased, and increasing reverse speed as the wiper input value is decreased.

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 (see Figure 13). This throttle type can be used for either the drive throttle or the brake throttle by using the VCL variables VCL_Throttle and VCL_Brake. How the VCL program is written will determine where the throttle signal originates from, making this a very flexible throttle input method. VCL can be written to use the throttle pot inputs, switch inputs, or CAN communitcation messages as the source of the throttle signals. If you have questions regarding this throttle type, contact the Curtis office nearest you.

Setting the Throttle Type to Type 5 also allows the throttle and brake pot inputs to be redefined by a VCL program for uses other than throttle or brake input. The variable names that VCL can use to interface with these two inputs are Throttle_Pot_Output (see page 80) and Brake_Pot_Output (see page 83).

INPUT/OUTPUT SIGNAL SPECIFICATIONS

The input/output signals wired to the 35-pin connector can be grouped by type as follows:

— digital inputs
— analog inputs
— digital outputs
— analog output
— high power outputs
— power supply outputs
— KSI and coil return
— throttle and brake
— communications ports
— encoder inputs
— program input.

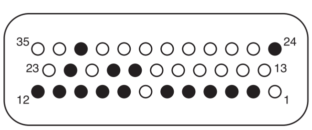
Their electrical characteristics are discussed below.

Digital inputs

Fifteen 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-.

Eight of these lines are designed to pull current to keep switch contacts clean and prevent leakage paths from causing false signals. The remaining seven lines are digital inputs associated with driver outputs; note that they have much higher input impedances. The two digital output lines can also be read as inputs, and are therefore included in this group; note that their threshold levels are not the same as for the other lines in this group.

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



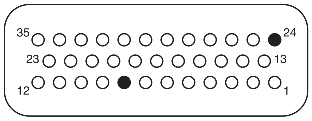
DIGITAL INPUT SPECIFICATIONS

SIGNAL NAME	PIN	LOGIC THRESHOLDS	INPUT IMPEDANCE	VOLTAGE RANGE	ESD TOLERANCE
Switch 1	24	Rising edge= 4.4 V max Falling edge= 1.5 V min	*24-36V models:* about 7.1 kΩ *36-48V models:* about 11.0 kΩ *48-80V models:* about 26.0 kΩ	-10 to 105 V	±8 kV (air discharge)
Switch 2	8				
Switch 3	9				
Switch 4	10				
Switch 5	11				
Switch 6	12				
Switch 7	22				
Switch 8	33				
Digital Out 6	19	Rising edge= 29.5 V max Falling edge= 10.1 V min	Below 5.5 V= 134 kΩ Above 5.5 V= 124 kΩ	-5 to 105 V	
Digital Out 7	20				
Driver 1	6	Rising edge= 4.4 V max Falling edge= 1.5 V min	Below 10 V= 300 kΩ Above 10 V= 150 kΩ	-0.5 to 105 V	
Driver 2	5				
Driver 3	4				
Driver 4	3				
Prop Driver	2				

Analog inputs

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

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

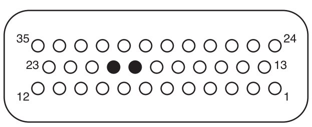


ANALOG INPUT SPECIFICATIONS

SIGNAL NAME	PIN	OPERATING VOLTAGE	INPUT IMPEDANCE	PROTECTED VOLTAGE	ESD TOLERANCE
Analog 1	24	0 to 10 V in 1024 steps	*24-36V models:* 7.2 kΩ *36-48V models:* 11.3 kΩ *48-80V models:* 28.3 kΩ	-10 to 105 V	±8 kV (air discharge)
Analog 2	8				

Digital outputs

Two control lines are available as low power digital outputs. These are open collector drivers that can only sink current, not source it, and are intended to drive LEDs or other low current loads connected to the +5V or +12V external power supplies; see power supply output group specs. Fault protection will turn off these outputs if output voltage exceeds about 15 V when the output is On (low output). Both outputs are protected against shorts to B+ or B-.

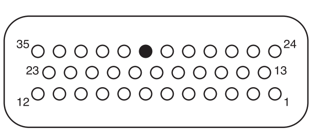


DIGITAL OUTPUT SPECIFICATIONS

SIGNAL NAME	PIN	ALLOWED VOLTAGE	OUTPUT IMPEDANCE	PROTECTED VOLTAGE	ESD TOLERANCE
Digital Out 6	19	0 to 15 V	On: 1 kΩ to B- Off: 134 kΩ	-5 to 105 V	±8 kV (air discharge)
Digital Out 7	20				

Analog output

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 <25ms for a 0–5V step and <30 ms for a 0–10V step. This output line is protected against shorts to B+ or B-.



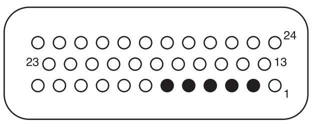
ANALOG OUTPUT SPECIFICATIONS

SIGNAL NAME	PIN	OUTPUT VOLTAGE	OUTPUT IMPEDANCE	PROTECTED VOLTAGE	ESD TOLERANCE
Analog Out	30	0 to 10 V	Source: 100 Ω Sink: 66 kΩ	-1 to 105 V	±8 kV (air discharge)

High power outputs

Five control lines can be used as high power output drivers. One of these drivers (the proportional driver) can be operated in a current control mode for driving a proportional valve or similar load. 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 five outputs are protected against shorts to B+ or B-.

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

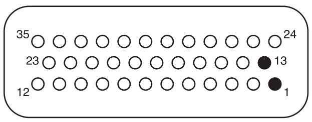


KSI and coil return

KSI input provides power for all low power control circuits, power capacitor precharge (before main contactor turn on), power supply outputs, and high power output drivers. Battery voltage is sensed on the input for the VCL battery discharge function.

Coil Return 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 (vehicle wiring correct, battery terminals reversed).



KSI AND COIL RETURN INPUT SPECIFICATIONS

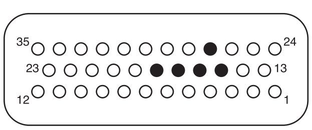
SIGNAL NAME	PIN	OPERATING VOLTAGE	INPUT CURRENT	PROTECTED VOLTAGE	ESD TOLERANCE
KSI	1	Between under- and overvoltage cutbacks	75 mA min, 1.0 A max *	-0.3 V to Severe Overvoltage	±8 kV (air discharge)
Coil Return	13		12 A max	-105 V to Severe Overvoltage	

* Min for 48–80V models at 105 V; max for 24–36V models at 16 V.
Additionally must carry the current supplied to the driver loads by the coil return (pin 13).

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 Section 6.



THROTTLE INPUT SPECIFICATIONS

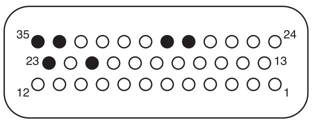
SIGNAL NAME	PIN	OPERATING VOLTAGE	INPUT IMPEDANCE	S/SINK CURRENT	PROTECTED VOLTAGE	ESD TOLERANCE
Throttle Pot High Brake Pot High	15 27	0 V (shorted to Pot Low) 5 V (open circuit)	n/a	7 mA nominal (source)	-50 to 105 V	±8 kV (air discharge)
Throttle Pot Wiper Brake Pot Wiper	16 17	0 to 6.25 V	290 kΩ (voltage and 3-wire)	0.76 mA nominal (source, 2-wire)		
Pot Low	18	0 to 10 V	20 Ω nom.	Faults if above 11 mA (sink)	-1 to 105 V	

Communications ports

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

The Curtis 1311 handheld programmer plugs into a connector wired to pins 28 and 29, along with ground (pin 7) and the +12V power supply (pin 25); see wiring diagram, Figure 3. The Curtis Model 840 display can plug into the same 4-pin connector.

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.

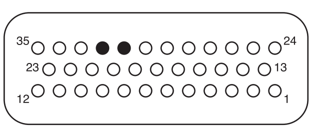


COMMUNICATIONS PORT SPECIFICATIONS

SIGNAL NAME	PIN	SUPPORTED PROTOCOL/DEVICES	DATA RATE	PROTECTED VOLTAGE	ESD TOLERANCE
CANH	23	CANopen, NODES 2.0, Other 11-bit identifier field CAN protocols	up to 500 kbps	Continuous= -36 to 105 V Transient= ±200 V	±8 kV (air discharge)
CANL	35				
CAN Term H	21			(no connection to external wiring)	±8 kV (air discharge)
CAN Term L	34				
Serial TX	28	Curtis 840 Display, 1311 Handheld Programmer, 1314 PC-based Programmer	as required, 9.6 to 56 kbps	0 to 12 V	±8 kV (air discharge)
Serial RX	29			±15 V	

Encoder inputs

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

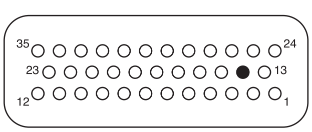


ENCODER INPUT SPECIFICATIONS

SIGNAL NAME	PIN	LOGIC THRESHOLDS	INPUT IMPEDANCE	MAX FREQ.	PROTECTED VOLTAGE	ESD TOLERANCE
Encoder A	31	Rising edge= 2.8 V max Falling edge= 2.2 V min	720 Ω (internal pull-up to +4V)	10 kHz	-5 to 105 V	±8 kV (air discharge)
Encoder B	32					

Program input

A single control line enables flash program download.



PROGRAM INPUT SPECIFICATIONS

SIGNAL NAME	PIN	LOGIC THRESHOLDS	INPUT IMPEDANCE	PROTECTED VOLTAGE	ESD TOLERANCE
Program	14	Rising edge= 11 V max Falling edge= 3 V min	150 kΩ	±105 V	±8 kV (air discharge)

3

PROGRAMMABLE PARAMETERS

The 1236/38 controller has a number of parameters that can be programmed using a Curtis 1311 handheld programmer. These programmable parameters allow the vehicle's performance to be customized to fit the needs of specific applications. For information on programmer operation, see Appendix C.

PROGRAMMING MENUS (for setting the programmable parameters)

The 1236/38's programmable parameters are grouped into hierarchical menus; see Table 3.

As shown in the table, there are two ways of tuning motor response characteristics: torque control mode and speed control mode. Use the Control Mode Select parameter (page 26) to select which tuning mode you will use. Note: Only one tuning mode can be used when programming; a combination is not allowed. If you adjust a parameter belonging to the other mode, the programmer will show the new setting but it will have no effect. Only those parameters belonging to the mode selected by the Control Mode Select parameter are active.

We urge you to read Section 5, Initial Setup, before adjusting any of the parameters.

CAUTION

Even if you opt to leave most of the parameters at their default settings, **it is imperative that you perform the procedures outlined in Section 5, which set up the basic system characteristics for your application**.

Table 3 Programmable Parameter Menus: 1311 Programmer

CONTROL MODE SELECT .... p. 26

1 - SPEED CONTROL MODE MENU
- **Speed Controller** .... p. 27
  - Max Speed
  - Kp
  - Ki
  - **Vel Feedforward** .... p. 28
    - Kvff
    - Build Rate
    - Release Rate
  - **Acc Feedforward** .... p. 29
    - Kaff
    - Kbff
    - Build Rate
    - Release Rate
- **Response** .... p. 30
  - Full Accel Rate HS
  - Full Accel Rate LS
  - Low Accel Rate
  - Neutral Decel Rate HS
  - Neutral Decel Rate LS
  - Full Brake Rate HS
  - Full Brake Rate LS
  - Low Brake Rate
  - **Fine Tuning** .... p. 31
    - Partial Decel Rate
    - HS (High Speed)
    - LS (Low Speed)
    - Reversal Soften
    - Max Speed Accel
    - Max Speed Decel
    - Accel Release Rate
- **Pump Enable** .... p. 31

2 - TORQUE CONTROL MODE MENU
- **Speed Limiter** .... p. 32
  - Max Speed
  - Kp
  - Ki
  - Kd
- **Response** .... p. 33
  - Accel Rate
  - Accel Release Rate
  - Brake Rate
  - Brake Release Rate
  - Neutral Braking
  - Neutral Taper Speed
  - **Fine Tuning** .... p. 34
    - Creep Torque
    - Gear Soften
    - Brake Taper Speed
    - Reversal Soften
    - Max Speed Decel

RESTRAINT MENU .... p. 36
- Restraint Forward
- Restraint Back
- Position Hold Enable
- **Position Hold** .... p. 36
  - Kp
  - Kp Deadband
  - Kd
  - Entry Rate

CURRENT LIMITS MENU .... p. 37
- Drive Current Limit
- Regen Current Limit
- Brake Current Limit
- **Power Limiting Map** .... p. 38
  - Base Speed
  - Delta Speed
  - **Drive Limiting Map** ..... p. 38
    - Nominal
    - Base Plus Delta
    - Base Plus 2xDelta
    - Base Plus 4xDelta
    - Base Plus 8xDelta
  - **Regen Limiting Map** .... p. 39
    - Nominal
    - Base Plus Delta
    - Base Plus 2xDelta
    - Base Plus 4xDelta
    - Base Plus 8xDelta

THROTTLE MENU .... p. 40
- Throttle Type
- Forward Deadband
- Forward Map
- Forward Max
- Forward Offset
- Reverse Deadband
- Reverse Map
- Reverse Max
- Reverse Offset
- HPD/SRO Enable
- Sequencing Delay

BRAKE MENU .... p. 42
- Brake Pedal Enable
- Brake Type
- Brake Deadband
- Brake Map
- Brake Max
- Brake Offset

Table 3 Programmable Parameter Menus: 1311 Programmer, cont'd

DRIVERS MENU

- **Main Contactor** .... p. 43
  - Main Enable
  - Pull In PWM
  - Holding PWM
  - Interlock Type
  - Open Delay
  - Checks Enable
  - Precharge Enable
- **Proportional Driver** .... p. 44
  - PD Enable
  - Hyd Lower Enable
  - PD Max Current
  - PD Min Current
  - PD Dither %
  - PD Dither Period
  - PD Kp
  - PD Ki
- **Fault Checking** .... p. 45
  - Driver1 Checks Enable
  - Driver2 Checks Enable
  - Driver3 Checks Enable
  - Driver4 Checks Enable
  - PD Checks Enable
  - EM Brake Disable Upon Fault
  - External Supply Max
  - External Supply Min

MOTOR MENU .... p. 46

- Typical Max Speed
- Swap Two Phases
- Swap Encoder Direction
- Encoder Steps
- Encoder SW Fault Enable
- **Temperature Control** .... p. 47
  - Sensor Enable
  - Temperature Hot
  - Temperature Max
  - Current Source
  - MotorTemp LOS Max Speed
  - Sensor Type
  - Sensor Temp Offset
  - **User Defined Sensor** ... p. 48
    - Sensor 0
    - Temp 0
    - Sensor 1
    - Temp 1
    - Sensor 2
    - Temp 2
    - Sensor 3
    - Temp 3
    - Sensor 4
    - Temp 4

BATTERY MENU .... p. 49

- Nominal Voltage
- User Overvoltage
- User Undervoltage
- Reset Volts Per Cell
- Full Volts Per Cell
- Empty Volts Per Cell
- Discharge Time
- BDI Reset Percent

VEHICLE MENU .... p. 52

- Metric Units
- Speed to RPM
- Capture Speed
- Capture Distance 1
- Capture Distance 2
- Capture Distance 3

EMERGENCY REVERSE MENU ..... p. 53

- EMR Enable
- EMR Type
- EMR Current
- EMR Speed
- EMR Accel Rate
- EMR Speed Decel Rate
- EMR Torque Decel Rate

CANOPEN INTERFACE MENU .... p. 54

- CANopen Interlock
- Master ID
- Slave ID
- Baud Rate
- Heartbeat Rate
- PDO Timeout Period
- Emergency Message Rate
- Suppress CANopen Init

MOTOR CONTROL TUNING MENU .. p. 55

- Motor Type
- Base Speed

☞ Individual parameters are presented in the following menu charts as shown in this example:

Parameter name as it appears in the programmer display ⇓	Allowable range in the programmer's units ⇓	Description of the parameter's function and, where applicable, suggestions for setting it ⇓
Max Speed *Max_Speed_SpdM*	50–6000 rpm *50–6000*	Defines the maximum allowed motor rpm at full throttle.
⇑ *Parameter name in VCL*	⇑ *Allowable range in VCL units*	

CONTROL MODE SELECT

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Control Mode Select *Control_Mode_Select*	1–2 *1–2*	This parameter determines which speed control method will be in effect when programming motor response: 1 = SPEED MODE 2 = TORQUE MODE. Contact Curtis if you are interested in a custom speed control method. Note: Do not change this parameter while the controller is powering the motor. Doing so will initiate a parameter change fault that must be cleared by cycling power; this protects the controller and the operator.

Note: Motor Speed Constraints

The maximum motor speed is a programmable parameter in both control modes. However—regardless of control mode—the maximum motor speed the controller will allow is constrained by the number of motor poles, the number of encoder pulses per motor revolution, and the maximum speed constraint imposed by the firmware.

Electrical frequency constraint

The maximum electrical frequency the controller will output is 300Hz. To determine how fast this constraint will allow your motor to spin, use the equation

Max Motor RPM = 36000 / Number of Motor Poles

(e.g., a 6-pole motor can run up to 6000 RPM).

Encoder pulses/revolution constraint

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

Max Motor RPM = 600000 / Encoder Size

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

Firmware max speed constraint

The maximum motor speed the controller will allow is 6000 RPM.

☞ The overall maximum motor speed allowed is the least of these three constraints.

1 – SPEED CONTROL MODE **SPEED CONTROLLER MENU**

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Max Speed *Max_Speed_SpdM*	50–6000 rpm *50–6000*	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. Note: The maximum motor rpm is subject to the constraints on page 26.
Kp *Kp_SpdM*	0–100 % *0–12288*	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, you 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 *Ki_SpdM*	5–100 % *100–2000*	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, you 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.

1 – SPEED CONTROL MODE **VELOCITY FEEDFORWARD MENU**

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Kvff *Kvff_SpdM*	0–200 A *0–2000*	This velocity feedforward term is designed to improve throttle responsiveness and speed controller performance, especially at low speeds. For traction systems, set it to slightly less than 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 *Vel_FF_Build_Rate_SpdM*	0.1–5.0 sec *100–5000*	Determines how fast the Kvff term builds up. For traction systems, if you feel or hear the mechanical slop pick up abruptly when you move 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. See Section 5, Initial Setup, for a more detailed adjustment procedure.
Release Rate *Vel_FF_Release_Rate_SpdM*	0.1–2.0 sec *100–2000*	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.

1 – SPEED CONTROL MODE **ACCELERATION FEEDFORWARD MENU**

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Kaff *Kaff_SpdM*	0–500 A *0–5000*	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 your present accel and decel rates, observe the average current you are running at full throttle at low speeds while accelerating, and set Kaff to that value.
Kbff *Kbff_SpdM*	0–500 A *0–5000*	This braking feedforward term is designed to improve braking responsiveness at all speeds. Using your present decel rates, observe the average current you are running at full throttle braking, and set Kbff to that value.
Build Rate *Acc_FF_Build_Rate_SpdM*	0.1–5.0 sec *100–5000*	Determines how fast the Kaff term builds up. For traction systems, if you feel or hear the mechanical slop pick up abruptly when you move 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 over-shoot if too much feedforward has been commanded. See Section 5, Initial Setup, for a more detailed adjustment procedure.
Release Rate *Acc_FF_Release_Rate_SpdM*	0.1–2.0 sec *100–2000*	Determines how fast the Kaff term releases. 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 CONTROL MODE **RESPONSE MENU**

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Full Accel Rate HS *Full_Accel_Rate_HS_SpdM*	0.5–30.0 sec. *500–30000*	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.
Full Accel Rate LS *Full_Accel_Rate_LS_SpdM*	0.5–10.0 sec. *500–10000*	Sets the rate (in seconds) at which the speed command increases when full throttle is applied at low vehicle speeds. Larger values represent slower response.
Low Accel Rate *Low_Accel_Rate_SpdM*	0.5–30.0 sec. *500–30000*	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 *Neutral_Decel_Rate_HS_SpdM*	1.0–30.0 sec. *1000–30000*	Sets the rate (in seconds) that is used to slow down the vehicle when the throttle is released to neutral at high vehicle speeds. Larger values represent slower response.
Neutral Decel Rate LS *Neutral_Decel_Rate_LS_SpdM*	1.0–30.0 sec. *1000–30000*	Sets the rate (in seconds) that is used to slow down the vehicle when the throttle is released to neutral at slow vehicle speeds. Larger values represent slower response.
Full Brake Rate HS *Full_Brake_Rate_HS_SpdM*	0.5–10.0 sec. *500–10000*	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. Larger values represent slower response.
Full Brake Rate LS *Full_Brake_Rate_LS_SpdM*	1.0–10.0 sec. *1000–10000*	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. Larger values represent slower response.
Low Brake Rate *Low_Brake_Rate_SpdM*	2.0–30.0 sec. *2000–30000*	Sets the rate (in seconds) at which the vehicle slows down at all speeds when a small amount of brake is applied or a small amount of throttle is applied in the opposite direction. Larger values represent slower response.

1 – SPEED CONTROL MODE **FINE TUNING MENU**

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Partial Decel Rate *Partial_Decel_Rate_SpdM*	5.0–30.0 sec. *5000–30000*	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) *HS*	0–100 % *0–32767*	Sets the percentage of the Typical Max Speed (page 46) at which the motor speed value will be captured and used in all subsequent interpolation whenever one of the “HS” parameters is invoked.
LS (Low Speed) *LS*	0–100 % *0–32767*	Sets the percentage of the Typical Max Speed (page 46) at which the motor speed value will be captured and used in all subsequent interpolation whenever one of the “LS” parameters is invoked.
Reversal Soften *Reversal Soften_SpdM*	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 or there is a big disparity between the brake rates and the accel rates.
Max Speed Accel *Max_Speed_Accel_SpdM*	0.1–30.0 sec *100–30000*	In some applications, the Max Speed value is changed frequently, through VCL or over the CAN bus. 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 * (4000–1000) / 5000 = 6.0 seconds to ramp from 1000 rpm to 4000 rpm.
Max Speed Decel *Max_Speed_Accel_SpdM*	0.1–30.0 sec *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 lowered. For example, suppose you change Max Speed 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 * (4500–2500) / 5000 = 2.0 seconds to ramp from 4500 rpm to 2500 rpm.
Accel Release Rate *Accel_Release_Rate_SpdM*	0–100 % *0–32767*	Determines how quickly deceleration will be initiated when the throttle is released while the vehicle is still accelerating. This parameter will reduce the “surge forward” that otherwise would result when the throttle is released, and which can give the operator the sense that the vehicle is not responding to the released throttle request.

1 – SPEED CONTROL MODE **PUMP ENABLE**

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Pump Enable *Pump_Enable_SpdM_Bit0*	On/Off *On/Off*	When this parameter is programmed On, the Speed Control mode is used to operate a pump motor rather than a vehicle drive motor. Pump applications turn the motor only in the forward direction.

2 – TORQUE CONTROL MODE **SPEED LIMITER MENU**

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Max Speed *Max_Speed_TrqM*	500–6000 rpm *500–6000*	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 on page 26.
Kp *Kp_TrqM*	0–100 % *0–12288*	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, you 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*	0–100 % *0–500*	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, you 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. Always start your speed limiter tuning by setting Ki = 0 and adjusting only the Kp terms and Current at Speed.
Kd *Kd_TrqM*	0–100 % *0–30000*	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.

2 – TORQUE CONTROL MODE **RESPONSE MENU**

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Accel Rate *Accel_Rate_TrqM*	0.1–5.0 sec. *100–5000*	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 *Accel_Release_Rate_TrqM*	0.1–2.0 sec. *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 *Brake_Rate_TrqM*	0.1–5.0 sec. *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 *Brake_Release_Rate_TrqM*	0.1–2.0 sec. *100–2000*	Adjusts the rate (in seconds) at which braking torque releases as as the vehicle transitions from braking to drive.
Neutral Braking *Neutral_Braking_TrqM*	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 (see Current Limits menu, page 37).
Neutral Taper Speed *Neutral_Taper_Speed_TrqM*	200–6000 rpm *200–6000*	Determines the motor speed below which the neutral braking current is linearly reduced from 100% to the Creep Torque current at zero rpm motor speed, when throttle is reduced; see Figure 7. Setting the taper speed too low relative to the neutral braking parameter may cause oscillations in the motor.

2 – TORQUE CONTROL MODE **FINE TUNING MENU**

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Creep Torque *Creep_Torque_TrqM*	0–100 % *0–32767*	Determines the amount of torque to move the vehicle from a stop with no throttle input; see Figure 7. ***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 (see page 42), creep torque is progressively disabled as brake is applied so as to prevent the motor from driving into the brakes and thus wasting energy.
Gear Soften *Gear_Soften_TrqM*	0–100 % *0–10000*	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 8.
Brake Taper Speed *Brake_Taper_Speed_TrqM*	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 9. 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.
Reversal Soften *Reversal_Soften_TrqM*	0–100 % *256–32767*	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.
Max Speed Decel *Max_Speed_Accel_TrqM*	0.1–30.0 sec *100–30000*	In some applications, the Max Speed value is changed frequently, through VCL or over the CAN bus. 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 lowered. The rate set by this parameter is the time to ramp from Typical Max Speed rpm to 0 rpm. For example, suppose you change Max Speed 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 * (4500–2500) / 5000 = 2.0 seconds to ramp from 4500 rpm to 2500 rpm.

Fig. 7 *Neutral braking (torque control mode).*

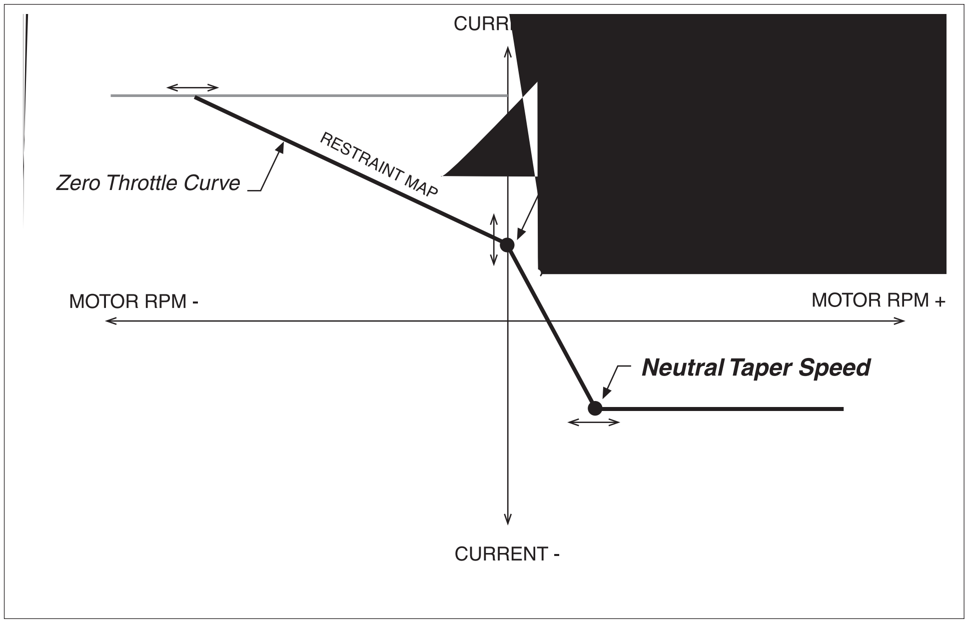
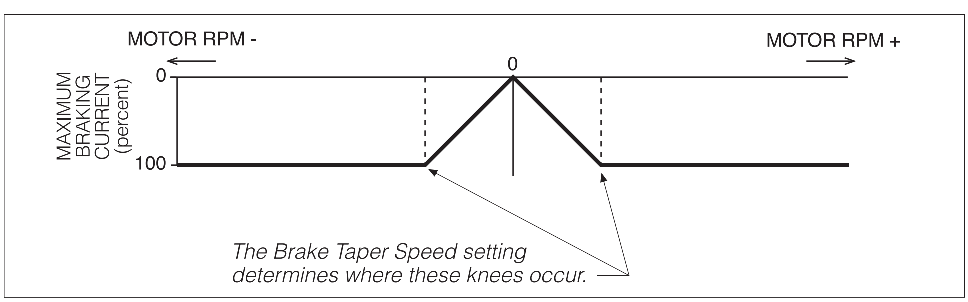


Fig. 8 *Effect of Gear Soften parameter (torque control mode).*

Fig. 9 *Effect of Brake Taper Speed parameter (torque control mode).*



RESTRAINT MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Restraint Forward *Restraint_Forward*	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 *Restraint_Back*	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.
Position Hold Enable *Position_Hold_Enable_Bit0*	On/Off *On/Off*	Allows the Position Hold mode to be entered at zero throttle when the vehicle reaches zero speed.

POSITION HOLD MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Kp *Kp_Position_Hold*	2–100 % *82–4096*	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.
Kp Deadband *Kp_Deadband_Position_Hold*	0–100 % *0–8192*	Allows a position feedback deadband around the setpoint, to help avoid instability caused by gear slop.
Kd *Kd_Position_Hold*	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.
Entry Rate *Entry_Rate_Position_Hold*	5–100 % *100–2000*	When the vehicle transitions from forward to reverse or from reverse to forward (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.

CURRENT LIMITS MENU		
PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Drive Current Limit *Drive_Current_Limit*	0–100 % *0–32767*	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.
Regen Current Limit *Regen_Current_Limit*	0–100 % *0–32767*	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 *Brake_Current_Limit*	0–100 % *0–32767*	Sets the maximum RMS regen current during braking when the brake throttle is applied, 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.

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

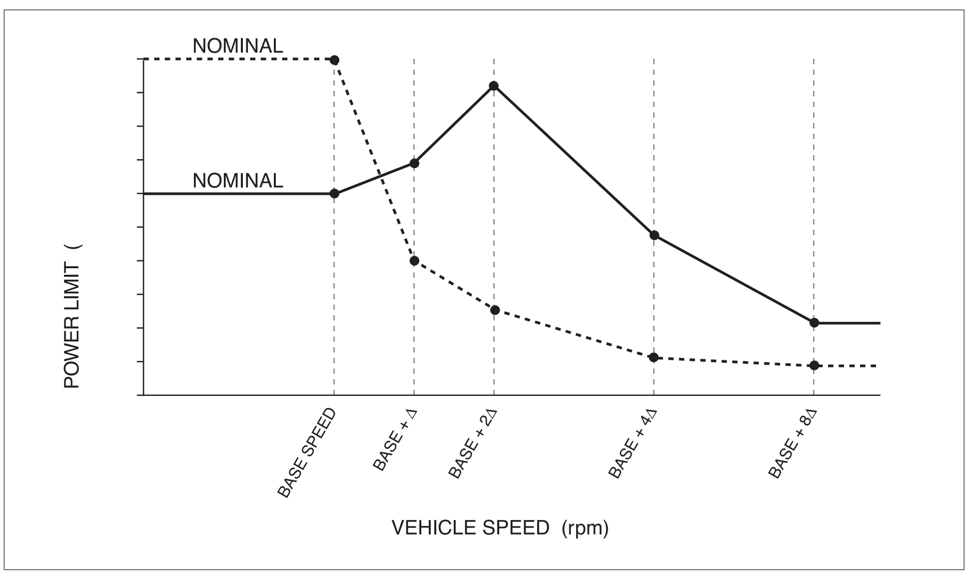
POWER LIMITING MAP MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Base Speed *PL_Base_Speed*	100–4000 rpm *100–4000*	Sets the base speed that will be used in the drive limiting map and regen limiting map.
Delta Speed *PL_Delta_Speed*	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.

DRIVE LIMITING MAP MENU

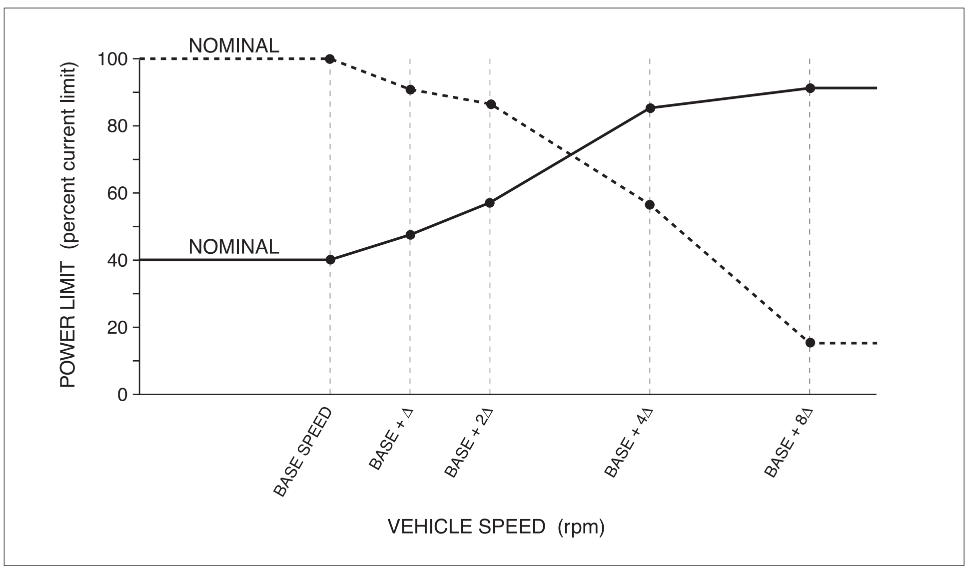
PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Nominal *PL_Drive_Nominal*	0–100 % *0–32767*	Sets
Base Plus Delta *PL_Drive_Base_Plus_Delta*	0–100 % *0–32767*	Sets
Base Plus 2xDelta *PL_Drive_Base_Plus_2xDelta*	0–100 % *0–32767*	Sets
Base Plus 4xDelta *PL_Drive_Base_Plus_4xDelta*	0–100 % *0–32767*	Sets
Base Plus 8xDelta *PL_Drive_Base_Plus_8xDelta*	0–100 % *0–32767*	Sets

Fig. 10 *Drive current limiting map (two examples).*



REGEN LIMITING MAP MENU		
PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Nominal *PL_Regen_Nominal*	0–100 % *0–32767*	These parameters define the percentage of regen current limit that will be applied at the speeds defined by the base 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.
Base Plus Delta *PL_Regen_Base_Plus_Delta*	0–100 % *0–32767*	
Base Plus 2xDelta *PL_Regen_Base_Plus_2xDelta*	0–100 % *0–32767*	
Base Plus 4xDelta *PL_Regen_Base_Plus_4xDelta*	0–100 % *0–32767*	
Base Plus 8xDelta *PL_Regen_Base_Plus_8xDelta*	0–100 % *0–32767*	

Fig. 11 *Regen current limiting map (two examples).*



THROTTLE MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Throttle Type *Throttle_Type*	1–5 *1–5*	The 1236/38 controller accepts a variety of throttle inputs. The throttle type parameter can be programmed as follows: 1 2-wire rheostat, 5kΩ–0 input 2 <u>single-ended</u> 3-wire 1kΩ–10kΩ potentiometer, or 0–5V voltage source 3 2-wire rheostat, 0–5kΩ input 4 <u>wigwag</u> 3-wire 1kΩ–10kΩ potentiometer, or 0–5V voltage source 5 VCL input (*VCL_Throttle*) Note: Do not change this parameter while the controller is powering the motor. Doing so will initiate a parameter change fault that must be cleared by cycling power; this protects the controller and the operator.
Forward Deadband *Forward_Deadband*	0–5.00 V *0–32767*	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 *Forward_Map*	0–100 % *0–32767*	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 *Forward_Max*	0–5.00 V *0–32767*	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 *Forward_Offset*	0–100 % *0–32767*	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.

☞ Note: All four throttle adjustment parameters — Deadband, Map, Max, Offset — condition the raw throttle voltage into a single % throttle command, as shown in Figure 12.

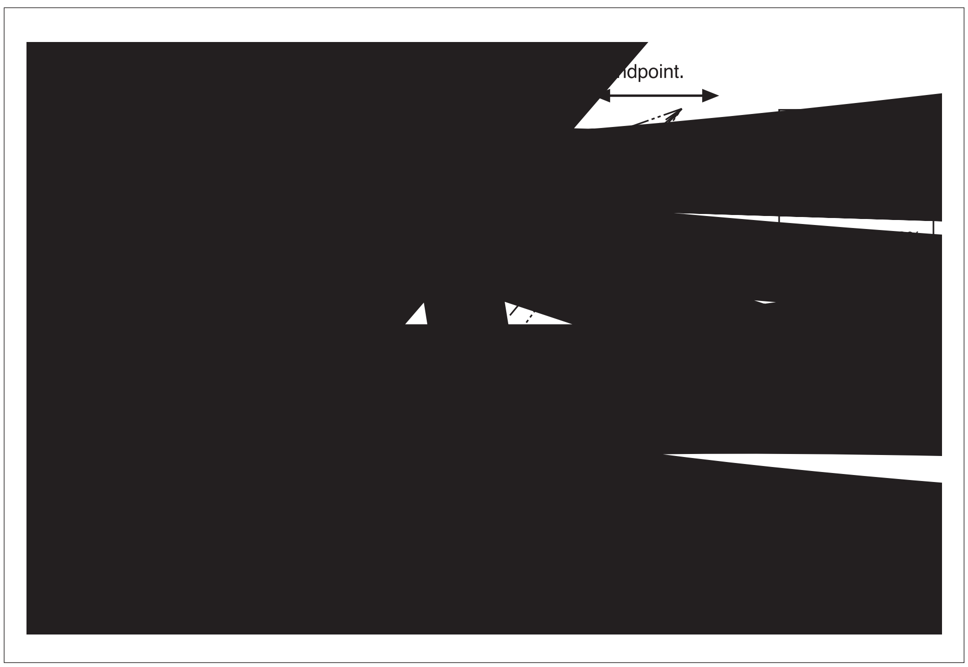
THROTTLE MENU, cont'd

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Reverse Deadband *Reverse_Deadband*	0–5.00 V *0–32767*	
Reverse Map *Reverse_Map*	0–100 % *0–32767*	
Reverse Max *Reverse_Max*	0–5.00 V *0–32767*	
Reverse Offset *Reverse_Offset*	0–100 % *0–32767*	
HPD/SRO Enable *HPD_SRO_Enable*	On/Off *On/Off*	Determines whether the HPD/SRO feature will be active. When programmed On, the controller will not start the vehicle unless these inputs are in the proper order: KSI first, then interlock, must be received before a direction input or a throttle input > 25%.
Sequencing Delay *Sequencing_Delay*	0.0–5.0 sec. *0–312*	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 [redacted]

Fig. 12 *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.



BRAKE MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Brake Pedal Enable *Brake_Pedal_Enable*	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 *Brake_Type*	1–5 *1–5*	The 1236/38 controller accepts a variety of brake inputs. The brake type parameter can be programmed as follows: 1 2-wire rheostat, 5kΩ–0 input 2 single-ended 3-wire 1kΩ–10kΩ potentiometer, 0–5V voltage source, or current source 3 2-wire rheostat, 0–5kΩ input 4 (not applicable) 5 VCL input (*VCL_Brake*) Note: Do not change this parameter while the controller is powering the motor. Doing so will initiate a parameter change fault that must be cleared by cycling power; this protects the controller and the operator.
Brake Deadband *Brake_Deadband*	0–5.00 V *0–32767*	The four Brake throttle adjustment parameters are the same as their Drive throttle counterparts; see descriptions on page 40 and Figure 12, page 41.
Brake Map *Brake_Map*	0–100 % *0–32767*	
Brake Max *Brake_Max*	0–5.00 V *0–32767*	
Brake Offset *Brake_Offset*	0–100 % *0–32767*	

MAIN CONTACTOR MENU		
PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Main Enable *Main_Enable*	On/Off *On/Off*	When programmed On, the controller's native software controls the main contactor; when programmed Off, the contactor is controlled by VCL. Note: When Main Enable is 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.
Pull In PWM *Pull_In_PWM*	0–100 % *0–32767*	The main contactor pull-in PWM parameter allows a high initial voltage when the main contactor driver first turns on, to ensure contactor closure. After 0.1 second, this peak voltage drops to the contactor holding voltage. Note: The two PWM parameters are battery-voltage-compensated relative to the set Nominal Voltage (see Battery Menu, page 49). In other words, the output PWM is adjusted to compensate for swings in battery voltage, so the percentage is relative to the set Nominal Voltage—not to the actual voltage.
Holding PWM *Holding_PWM*	0–100 % *0–32767*	The main contactor holding PWM parameter allows a reduced average voltage to be applied to the contactor once it has closed. The parameter must be set high enough to hold the contactor closed under all shock and vibration conditions the vehicle will be subjected to. See Note above. Here's an example. Suppose Nominal Voltage is set to 48V and Holding PWM is set to 75%. When the actual bus voltage is 48V, the driver will send 75% PWM (36V) to the main contactor. Now suppose the bus voltage dips to 40V; the driver will now send 90% PWM (48/40 × Holding_PWM) to the main contactor, to keep the output voltage at 36V.
Interlock Type *Interlock_Type*	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 KSI.
Open Delay *Open_Delay*	0–40 sec. *0–2500*	Applicable only when the Interlock Type parameter is set to zero. 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.
Checks Enable *Checks_Enable*	On/Off *On/Off*	When programmed On, the controller performs ongoing checks to ensure that the main contactor has closed properly each time it is commanded to do so, and that it has not welded closed. These checks are not performed if this parameter is Off. The main contactor driver, however, is always protected from short circuits.
Precharge Enable *Precharge_Enable*	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.

PROPORTIONAL DRIVER MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
PD Enable *PD_Enable*	On/Off *On/Off*	Determines how the PWM of the proportional driver is controlled. When programmed On, it is controlled by the controller's PID software. When programmed Off, it is controlled by the VCL function *Put_PWM* (PWM5, value); see Figure 15, page 84.
Hyd Lower Enable *Hyd_Lower_Enable*	On/Off *On/Off*	When programmed On, lowering is controlled by throttle position. When programmed Off, lowering is controlled by the VCL variable *VCL_PD_Throttle*; see Figure 15, page 84.
PD Max Current *PD_Max_Current*	0.0–2.0 A *0–607*	* The Lower 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*	0.0–2.0 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*	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 none-add%-none-subtract%.
PD Dither Period *PD_Dither_Period*	16–112 mS *1–7*	* Sets the period for proportional valve dither.
PD Kp *PD_Kp*	0–100 % *0–32767*	Sets the proportional gain of the current feedback controller.
PD Ki *PD_Ki*	0–100 % *0–32767*	Sets the integral gain of the current feedback controller. Integral gain tries to force the error to zero. Higher gains force the motor to respond quickly but may cause oscillations.

* These parameter descriptions assume the proportional driver is being used to drive a proportional valve.

FAULT CHECKING MENU		
PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Driver1 Checks Enable *Driver1_Checks_Enable*	On/Off *On/Off*	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 fault is detected, the controller opens the driver and issues a fault code. This fault detection is active only at 100% and 0% PWM. Note: Short circuit protection is always active at these five drivers, regardless of how checks enable is set.
Driver2 Checks Enable *Driver2_Checks_Enable*	On/Off *On/Off*	
Driver3 Checks Enable *Driver3_Checks_Enable*	On/Off *On/Off*	
Driver4 Checks Enable *Driver4_Checks_Enable*	On/Off *On/Off*	
PD Checks Enable *PD_Checks_Enable*	On/Off *On/Off*	
EM Brake Disable Upon Fault *EM_Brake_Disable_Upon_Fault*	On/Off *On/Off*	When programmed On, the controller's operating system will override VCL control and drop the electromagnetic brake when a severe fault occurs.
External Supply Max *External_Supply_Max*	5–200 mA *52–800*	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 *External_Supply_Min*	5–200 mA *52–800*	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.

MOTOR MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Typical Max Speed *Typical_Max_Speed*	500–6000 rpm *500–6000*	Set this parameter to the typical maximum motor speed of the vehicle. This value does not need to be set precisely; a ballpark estimate will do. It is used to scale various other parameters.
Swap Encoder Direction *Swap_Encoder_Direction*	On/Off *On/Off*	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. Doing so will initiate a parameter change fault that must be cleared by cycling power; this protects the controller and the operator. ***Adjusting this parameter can be hazardous. For instructions, see Section 5, Step ⑨ (pages 67–68).***
Swap Two Phases	On/Off	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. Doing so will initiate a parameter change fault that must be cleared by cycling power; this protects the controller and the operator. ***Adjusting this parameter can be hazardous. For instructions, see Section 5, Step ⑨ (pages 67–68).***
Encoder Steps *Encoder_Steps*	4–256 *4–256*	Sets the number of encoder pulses per revolution. This must be set to match the encoder; see motor nameplate. Note: Do not change this parameter while the controller is powering the motor. Doing so will initiate a parameter change fault that must be cleared by cycling power; this protects the controller and the operator.
Encoder SW Fault Enable *Encoder_SW_Fault_Enable_Bit0*	On/Off *On/Off*	When programmed On, the software algorithm for detecting a failed motor encoder is enabled. The motor encoder failure algorithm can trigger the vehicle will be disabled. Optionally, it is possible to setup a vehicle to operate in an LOS (Limited Operating Strategy) mode without the motor encoder in order to "limp home" at a reduced speed for maintenance. Customers interested in the LOS mode should contact their local Curtis applications engineer for details.

MOTOR TEMPERATURE CONTROL MENU		
PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Sensor Enable *Sensor_Enable*	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 cutback the drive 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 consistant performance.
Temperature Hot *Temperature_Hot*	0–250 °C *0–2500*	Defines the temperature at which drive current cutback begins.
Temperature Max *Temperature_Max*	0–250 °C *0–2500*	Defines the temperature at which drive current is cut back to zero.
Current Source *Current_Source*	On/Off *On/Off*	For resistive-type temperature sensors (Types 1–4 and maybe Type 0), the current source must be set to On.
MotorTemp LOS Max Speed *MotorTemp_LOS_Max_Speed*	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_SpdM or Max_Speed_TrqM) ***or*** the MotorTemp_LOS_Max_Speed, whichever is lower.
Sensor Type *Sensor_Type*	0–5 *0–5*	Type 0 is a user-defined sensor; see menu on next page. Types 1–5 are predefined in the software: Type 1 KTY83–122 Type 2 2× Type 1, in series Type 3 KTY84–130 Type 4 2× Type 3, in series Type 5 PT1000.
Sensor Temp Offset *Sensor_Temp_Offset*	-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.

USER-DEFINED TEMPERATURE SENSOR MENU (Sensor Type = 0)

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Sensor 0 *Sensor_0*	0–10 V *0–1023*	These parameters can be used to create a Temperature Sensor Voltage vs. Actual Temperature map to describe your motor's temperature sensor response curve. The map is defined by five calibration points, which are set by five sensor-temp pairs. You will want to select at least one point near your Temperature Hot value so the controller will accurately regulate motor temperature. The points must be entered into the 1311 so that the Sensor 0 point is the lowest voltage point and each sensor point after is increasing in voltage. The corresponding temperature values can be increasing or decreasing. Note: To use the user-defined temperature sensor menu, the Sensor Type must be set to 0.
Temp 0 *Temp_0*	-50 – 250 °C *-500–2500*	
Sensor 1 *Sensor_1*	0–10 V *0–1023*	
Temp 1 *Temp_1*	-50 – 250 °C *-500–2500*	
Sensor 2 *Sensor_2*	0–10 V *0–1023*	
Temp 2 *Temp_2*	-50 – 250 °C *-500–2500*	
Sensor 3 *Sensor_3*	0–10 V *0–1023*	
Temp 3 *Temp_3*	-50 – 250 °C *-500–2500*	
Sensor 4 *Sensor_4*	0–10 V *0–1023*	
Temp 4 *Temp_4*	-50 – 250 °C *-500–2500*	

BATTERY MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Nominal Voltage *Nominal_Voltage*	24–84 V *1536–5376*	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. 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, User Overvoltage, and User Undervoltage parameter settings and the controller's severe undervoltage, minimum voltage, and maximum voltage ratings:

VOLTAGE RATINGS			
CONTROLLER	SEVERE UNDERVOLTAGE	MIN VOLTAGE	MAX VOLTAGE
24–36V	12V	16.8V	45V
36–48V	15V	25.2V	60V
48–80V	15V	33.6V	105V

<u>Overvoltage</u> = Either Max Voltage (see voltage ratings table) *or* User Overvoltage × Nominal Voltage, whichever is lower.

<u>Severe Overvoltage</u> = Overvoltage (see previous item) + 10V.

<u>Undervoltage</u> = Either Min Voltage (see voltage ratings table) *or* User Undervoltage × Nominal Voltage, whichever is higher.

<u>Severe Undervoltage</u> = See voltage ratings table.

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
User Overvoltage *User_Overvoltage*	115–200 % *293–512*	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 48V 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 *User_Undervoltage*	50–80 % *128–204*	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.

BDI Algorithm

The BDI (battery discharge indicator) algorithm continuously calculates the battery state-of-charge whenever KSI is on. The result of the BDI algorithm is the variable BDI Percentage, which is viewable in the 1311 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 and sealed maintenance-free 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 your type of batteries as the starting point in setting the reset, full, and empty volts-per-cell parameters.

BATTERY MENU, cont'd		
PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Reset Volts Per Cell *Reset_Volts_Per_Cell*	0.90–3.00 V *90–300*	The reset voltage level is checked only once, when KSI is first turned on. 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. Reset Voltage Level = Reset Volts Per Cell × number of cells in the battery pack.*
Full Volts Per Cell *Full_Volts_Per_Cell*	0.90–3.00 V *90–300*	The full voltage level sets the Keyswitch Voltage that is considered to be 100% state-of-charge; when a loaded battery drops below this voltage, it begins to lose charge. Keyswitch Voltage is viewable in the 1311 menu Monitor » Battery. Full Voltage Level = Full Volts Per Cell × number of cells in the battery pack.*
Empty Volts Per Cell *Empty_Volts_Per_Cell*	0.90–3.00 V *90–300*	The empty voltage level sets the Keyswitch_Voltage that is considered to be 0% state-of-charge. Empty Voltage Level = Empty Volts Per Cell × number of cells in the battery pack.*
Discharge Time *Discharge_Time*	0–600 min. *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 keyswitch 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. State of Charge Level = ((Full Voltage Level - Empty Voltage Level) × BDI Percentage) + Empty Voltage Level.
BDI Reset Percent *BDI_Reset_Percent*	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 your battery pack, divide your Nominal Voltage setting (page 45) by 2.

VEHICLE MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Metric Units *Metric_Units*	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). The distance variables displayed in the Monitor » Vehicle menu, page 61.
Speed to RPM *Speed_to_RPM*	10.0–3000.0 *100–30000*	This parameter affects the vehicle speed displayed in the Monitor » Motor menu (see page 59), and also modifies the VCL variable *Vehicle_Speed*; it does <u>not</u> 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)*530.5, where G = gear ratio, d = tire diameter [cm]. MPH to RPM: (G/d)*336.1, where G = gear ratio, d = tire diameter [in].
Capture Speed *Capture_Speed*	0–8000 rpm *0–8000*	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” in the Monitor » Vehicle menu (page 61). This timer starts every time the motor accelerates from zero speed.
Capture Distance 1 *Capture_Distance_1*	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 (page 61). 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 *Capture_Distance_2*	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 *Capture_Distance_3*	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.

EMERGENCY REVERSE MENU		
PARAMETER	ALLOWABLE RANGE	DESCRIPTION
EMR Enable *EMR_Enable*	On/Off *On/Off*	Determines whether the emergency reverse function is active.
EMR Type *EMR_Type*	0–1 *0–1*	Two emergency reverse options are available: 0 = emergency reverse activated by emergency reverse switch input (switch 1). 1 = emergency reverse controlled by VCL.
EMR Current *EMR_Current*	0–100 % *0–32767*	Defines the maximum braking current provided through the motor when the optional emergency reverse function is engaged. The emergency reverse current limit is a percentage of the controller's full rated current.*
EMR Speed *EMR_Speed*	50–6000 rpm *50–6000*	Defines the maximum reverse speed of the motor, when the optional emergency reverse function is engaged.
EMR Accel Rate *EMR_Accel_Rate*	0.1–3.0 sec. *100–3000*	Sets the rate (in seconds) at which the vehicle accelerates in the opposite direction during emergency reverse. Larger values represent slower acceleration and therefore gentler response. More abrupt response can be achieved by decreasing the emergency reverse acceleration rate (i.e., by setting the rate to a smaller value).
EMR Speed Decel Rate *EMR_Speed_Decel_Rate*	0.1–3.0 sec. *100–3000*	Sets the rate (in seconds) at which the braking force builds upon activation of emergency reverse while the vehicle is still moving toward the operator. Larger values represent slower deceleration and therefore gentler response. More abrupt response can be achieved by decreasing the emergency reverse deceleration rate (i.e., by setting the rate to a smaller value). This parameter has no effect when the controller is operating in Torque Control Mode.
EMR Torque Decel Rate *EMR_Torque_Decel_Rate*	0–0.5 sec. *10–500*	Sets the rate (in seconds) at which torque builds to slow down the vehicle; this parameter is active in both Speed Control Mode and Torque Control Mode.

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

CANOPEN INTERFACE MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
CANopen Interlock *CANopen_Interlock*	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, page 63.
Master ID *Master_ID*	0–3 *0–3*	The CAN Master ID for incoming CAN messages to the CANopen Slave system.
Slave ID *Slave_ID*	0–31 *0–31*	The CAN Slave ID for outgoing CAN messages from the CANopen Slave system.
Baud Rate *Baud_Rate*	0–2 *0–2*	Sets the CAN baud rate for the CANopen Slave system: 0=125Kbps, 1=250Kbps, 2=500Kbps.
Heartbeat Rate *Heartbeat_Rate*	16–200 msec *4–50*	Sets the rate at which the CAN heartbeat messages are sent from the CANopen Slave system.
PDO Timeout Period *PDO_Timeout_Period*	0–200 msec *0–50*	Sets the PDO timeout period for the CANopen Slave system. After the slave controller has sent a PDO MISO, it will declare a PDO Timeout Fault if the master controller has not sent a reply PDO MOSI message within the set time. Either PDO1 MOSI or PDO2 MOSI will reset the timer. Setting the PDO Timeout Period = 0 will disable this fault check.
Emergency Message Rate *Emergency_Message_Rate*	16–200 msec *4–50*	Sets the minimum rate between CAN emergency messages from the CANopen Slave system. This prevents quickly changing fault states from generating so many emergency messages that they flood the CAN bus.
Suppress CANopen Init *Suppress_CANopen_Init*	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).

MOTOR CONTROL TUNING MENU

PARAMETER	ALLOWABLE RANGE	DESCRIPTION
Motor Type *Motor_Type*	0–200 *0–200*	This parameter references a predefined table of motor parameters for many AC motors. Consult your local Curtis applications engineer for information on how to set this parameter based on your application and motor.
Base Speed *MotorData1*	200–6000 rpm *200–6000*	This parameter needs to be reset each time the Motor Type is changed or the low speed current limit is changed. For example, if you lower Drive_Current_Limit (page 37) or PL_Drive_Nominal (page 38), you should consider changing this parameter. To determine the correct value, perform this tuning test. The test should be run with batteries that have a reasonable charge. In either Torque Control Mode or Speed Control Mode, set your accel rates to be fast—so that you'll be accelerating at full current during the test. From a stop, with the throttle in neutral, apply full throttle and accelerate to high speed and then stop. After stopping, note the value displayed in Monitor » Controller » Motor Tuning » Base Speed Captured, and enter this value for the Base Speed setting. The test restarts each time the vehicle comes to a stop and the throttle is released, so be sure to note the value before driving away.

CLONING (for copying parameter settings to multiple controllers)

Once a controller has been programmed to the desired settings, these settings can be transferred as a group to other controllers, thus creating a family of "clone" controllers with identical settings. **Cloning only works between controllers with the same model number and software version.** For example, the programmer can read all the information from a 1236-4415 controller and write it to other 1236-4415 controllers; however, it cannot write that same information to 1236-4416 controllers.

To perform cloning, plug the 1311 programmer into the controller that has the desired settings. Scroll down to the Functions menu; "Settings" is the only function included here. Select "Get settings from controller" to copy the settings into the programmer.

Plug the programmer into the controller that you want to have these same settings, and select "Write settings to controller."

4a

MONITOR MENU

Through its Monitor menu, the 1311 programmer provides access to real-time data during vehicle operation. This information is helpful during diagnostics and troubleshooting, and also while adjusting programmable parameters. The Monitor menu has seven submenus:

MONITOR MENU
—Inputs............. p. 56
—Outputs........... p. 58
—Battery............ p. 59
—Motor.............. p. 59
—Controller........ p. 60
—Vehicle............ p. 61
—CAN Status..... p. 63

Monitor Menu: INPUTS

VARIABLE	DISPLAY RANGE	DESCRIPTION
Throttle Command *Throttle_Command*	-100–100 % *-32768–32767*	Throttle request to slew rate block.
Mapped Throttle *Mapped_Throttle*	-100–100 % *-32768–32767*	Mapped throttle request.
Throttle Pot *Throttle_Pot_Raw*	0–5.5 V *0–36044*	Voltage at throttle pot wiper (pin 16).
Brake Command *Brake_Command*	0–100 % *0–32767*	Brake request to slew rate block.
Mapped Brake *Mapped_Brake*	0–100 % *0–32767*	Mapped brake request.
Brake Pot *Brake_Pot_Raw*	0–5.5 V *0–36044*	Voltage at brake pot wiper (pin 17).
Interlock *Interlock_State*	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 *EMR_State*	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 from VCL function if EMR Type = 1.
PD Throttle *PD_Throttle*	0–100 % *0–32767*	Proportional driver current request.
Analog 1 *Analog1_Input*	0–10.0 V *0–1023*	Voltage at analog 1 (pin 24).

Monitor Menu: INPUTS, cont'd

VARIABLE	DISPLAY RANGE	DESCRIPTION
Analog 2 *Analog2_Input*	0–10.0 V *0–1023*	Voltage at analog 2 (pin 8).
Switch 1 *Sw_1*	On/Off *On/Off*	Switch 1 on or off (pin 24).
Switch 2 *Sw_2*	On/Off *On/Off*	Switch 2 on or off (pin 8).
Switch 3 *Sw_3*	On/Off *On/Off*	Switch 3 on or off (pin 9).
Switch 4 *Sw_4*	On/Off *On/Off*	Switch 4 on or off (pin 10).
Switch 5 *Sw_5*	On/Off *On/Off*	Switch 5 on or off (pin 11).
Switch 6 *Sw_6*	On/Off *On/Off*	Switch 6 on or off (pin 12).
Switch 7 *Sw_7*	On/Off *On/Off*	Switch 7 on or off (pin 22).
Switch 8 *Sw_8*	On/Off *On/Off*	Switch 8 on or off (pin 33).
Driver 1 Input *Sw_9*	On/Off *On/Off*	Driver 1 input on or off (pin 6).
Driver 2 Input *Sw_10*	On/Off *On/Off*	Driver 2 input on or off (pin 5).
Driver 3 Input *Sw_11*	On/Off *On/Off*	Driver 3 input on or off (pin 4).
Driver 4 Input *Sw_12*	On/Off *On/Off*	Driver 4 input on or off (pin 3).
PD Input *Sw_13*	On/Off *On/Off*	Proportional driver on or off (pin 2).
DigOut6 Input *Sw_14*	On/Off *On/Off*	Digital Out 6 input on or off (pin 19).
DigOut7 Input *Sw_15*	On/Off *On/Off*	Digital Out 7 input on or off (pin 20).

Monitor Menu: OUTPUTS

VARIABLE	DISPLAY RANGE	DESCRIPTION
Analog Out *Analog_Output*	0–10.0 V *0–32767*	Voltage at Analog output (pin 30).
Digital Out 6 *Dig6_Output*	On/Off *On/Off*	Digital Out 6 output on or off (pin 19).
Digital Out 7 *Dig7_Output*	On/Off *On/Off*	Digital Out 7 output on or off (pin 20).
Driver 1 PWM *PWM1_Output*	0–100 % *0–32767*	Driver 1 PWM output (pin 6).
Driver 2 PWM *PWM2_Output*	0–100 % *0–32767*	Driver 2 PWM output (pin 5).
Driver 3 PWM *PWM3_Output*	0–100 % *0–32767*	Driver 3 PWM output (pin 4).
Driver 4 PWM *PWM4_Output*	0–100 % *0–32767*	Driver 4 PWM output (pin 3).
PD PWM *PD_Output*	0–100 % *0–32767*	Proportional driver PWM output (pin 2).
PD Current *PD_Current*	0–2.0 A *0–607*	Current at proportional driver (pin 2).
5 Volts *5_Volts_Output*	0–6.25 V *0–1023*	Voltage at +5V output (pin 26).
Ext Supply Current *Ext_Supply_Current*	5–200 mA *52–800*	Combined current of the external +12V and +5V voltage supplies (pins 25 and 26).
Pot Low *Pot_Low_Output*	0–6.25 V *0–1023*	Voltage at pot low (pin 18).

Monitor Menu: BATTERY

VARIABLE	DISPLAY RANGE	DESCRIPTION
BDI *BDI_Percentage*	0–100 % *0–100*	Battery state of charge.
Capacitor Voltage *Capacitor_Voltage*	0–105 V *0–6720*	Voltage of controller's internal capacitor bank at B+ stud.
Keyswitch Voltage *Keyswitch_Voltage*	0–105 V *0–10500*	Voltage at KSI (pin 1).

Monitor Menu: MOTOR

VARIABLE	DISPLAY RANGE	DESCRIPTION
Motor RPM *Motor_RPM*	-12000–12000 rpm *-12000–12000*	Motor speed in revolutions per minute.
Vehicle Speed *Vehicle_Speed*	-327.7–327.7 *-32768–32767*	Vehicle speed, in MPH or KPH; the value displayed is the Motor RPM divided by the value set for the Speed to RPM parameter (see Program » Vehicle menu, page 52).
Temperature *Motor_Temperature*	-100–300 °C *-1000–3000*	Temperature sensor readout.

Monitor Menu: CONTROLLER

VARIABLE	DISPLAY RANGE	DESCRIPTION
Current (RMS) *Current_RMS*	0–1000 A *0–10000*	RMS current of the controller, taking all three phases into account.
Modulation Depth *Modulation_Depth*	0–100 % *0–1182*	Percentage of available voltage being used.
Frequency *Frequency*	-300–300 Hz *-18000–18000*	Controller electrical frequency.
Temperature *Controller_Temperature*	-100–300 °C *-1000–3000*	Controller internal temperature.
Main State *Main_State*	0–10 *0–10*	Main contactor state: 0= open 1= precharge 2= weldcheck 3= closingdelay 4= missingcheck 5= closed (when Main Enable = On) 6= delay 7= arccheck 8= opendelay 9= fault 10= closed (when Main Enable = Off).
Regen *Regen_State*	On/Off *On/Off*	On when regen braking is taking place; Off when it is not.
VCL Error Module *Last_VCL_Error_Module*	0–65536 *0–65536*	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 (system information file), which should help pinpoint the VCL error that caused the runtime error.
VCL Error *Last_VCL_Error*	0–65536 *0–65536*	See description of VCL Error Module, above.

Monitor Menu: VEHICLE

VARIABLE	DISPLAY RANGE	DESCRIPTION
Vehicle Speed *Vehicle_Speed*	-327.7–327.7 *-32768–32767*	Vehicle speed, in units of MPH or KPH, depending on the setting of the Metric Units parameter (see Program » Vehicle menu, page 52). For accurate speed estimates, the Speed to RPM parameter be set correctly (see page 52).
Vehicle Odometer *Vehicle_Odometer*	0–42949672.9 *0–4294967295*	Vehicle distance traveled, in units of miles or km, depending on the setting of the Metric Units parameter (page 52). For accurate distance measurements, the Speed to RPM parameter must be set correctly (page 52).
Vehicle Acceleration *Vehicle_Acceleration*	0–10 g *0–1000*	Vehicle acceleration. The Speed to RPM parameter must be set correctly for accurate measurement.
Time to Speed *Time_to_Capture_Spd*	0–128 sec *0–32000*	Time taken for the vehicle to go from 0 rpm to the programmed Capture Speed (see Program » Vehicle menu, page 52) during its most recent such acceleration.
Time to Distance 1 *Time_to_Capture_Dist_1*	0–128 sec *0–32000*	Time taken for the vehicle to travel from 0 rpm to the programmed Capture Distance 1 (see Program » Vehicle menu, page 52) during its most recent such trip. For accurate distance measurements, the Speed to RPM parameter must be set correctly (page 52).
Time to Distance 2 *Time_to_Capture_Dist_2*	0–128 sec *0–32000*	Time taken for the vehicle to travel from 0 rpm to the programmed Capture Distance 2 (see Program » Vehicle menu, page 52) during its most recent such trip. For accurate distance measurements, the Speed to RPM parameter must be set correctly (page 52).
Time to Distance 3 *Time_to_Capture_Dist_3*	0–128 sec *0–32000*	Time taken for the vehicle to travel from 0 rpm to the programmed Capture Distance 3 (see Program » Vehicle menu, page 52) during its most recent such trip. For accurate distance measurements, the Speed to RPM parameter must be set correctly (page 52).

Monitor Menu: VEHICLE, cont'd		
VARIABLE	DISPLAY RANGE	DESCRIPTION
Braking Distance Captured *Braking_Distance_Captured*	0–1000000.0 *0–400000000*	Distance traveled by the vehicle starting with vehicle braking (initiated by throttle reversal, brake pot, or VCL_Brake) and ending when Motor_RPM = 0. Units are meters or feet, depending on the setting of the Metric Units parameter (page 52). For accurate distance measurements, the Speed to RPM parameter must be set correctly (page 52).
Distance Since Stop *Distance_Since_Stop*	0–1000000.0 *0–400000000*	Distance traveled by the vehicle starting from a stop. In effect, the vehicle is used as a tape measure. (In other words, if you travel 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 (page 52). Units are meters or feet, depending on the setting of the Metric Units parameter (page 52).
Distance Fine *Distance_Fine_Long*	-214748364.8–214748364.7 *-2147483648–2147483647*	Position measurement. Net distance in both the forward and reverse directions. (In other words, if you travel 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. Units are decimeters or inches, depending on the setting of the Metric Units parameter (page 52). For accurate distance measurements, the Speed to RPM parameter must be set correctly (page 52).

Monitor Menu: CAN STATUS

VARIABLE	DISPLAY RANGE	DESCRIPTION
CAN NMT State *CAN_NMT_State*	0–127 *0–127*	Controller CAN NMT state: 0=initialization, 4=stopped, 5=operational, 127=pre-operational.
PDO1 MOSI Byte Map*	$0 - 2^{32}$	Mapping objects for PDO1 MOSI's eight bytes.
PDO1 MISO Byte Map*	$0 - 2^{32}$	Mapping objects for PDO1 MISO's eight bytes.
PDO2 MOSI Byte Map*	$0 - 2^{32}$	Mapping objects for PDO2 MOSI's eight bytes.
PDO2 MISO Byte Map*	$0 - 2^{32}$	Mapping objects for PDO2 MISO's eight bytes.

* Each of these byte maps is a submenu containing 8 variables, one for each byte. Each variable is 32 bits. For example, the PDO1 MOSI Byte Map menu looks like this:

PDO1 MOSI Byte Map

1	$0 - 2^{32}$	Mapping object for byte 1 of PDO1 MOSI.
2	$0 - 2^{32}$	Mapping object for byte 2 of PDO1 MOSI.
3	$0 - 2^{32}$	Mapping object for byte 3 of PDO1 MOSI.
4	$0 - 2^{32}$	Mapping object for byte 4 of PDO1 MOSI.
5	$0 - 2^{32}$	Mapping object for byte 5 of PDO1 MOSI.
6	$0 - 2^{32}$	Mapping object for byte 6 of PDO1 MOSI.
7	$0 - 2^{32}$	Mapping object for byte 7 of PDO1 MOSI.
8	$0 - 2^{32}$	Mapping object for byte 8 of PDO1 MOSI.

4b CONTROLLER INFORMATION MENU

This menu provides ID and version numbers for your controller hardware and software.

CONTROLLER INFORMATION MENU

VARIABLE	DISPLAY RANGE	DESCRIPTION
Model Number *Model_Number*	0–4294967295 *0–4294967295*	Model number. For example, if you have a 1236 controller with the model number 1236-4501, the Model Number variable will have a value of 12364501.
Serial Number *Serial Number*	0–4294967295 *0–4294967295*	Serial number. For example, if the serial number printed on your controller is 05045L.11493, the Serial Number variable will have the value of 11493.
Mfg Date Code *Manuf_Date*	0–32767 *0–32767*	Controller date of manufacture, with the last three digits being reserved for the day. For example, if the serial number printed on your controller is 05045L.11493, the Mfg Date Code variable will have the value of 5045 (45th day of 2005).
Hardware Version *Hardware_Ver*	0–32.767 *0–32767*	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 *OS_Ver*	0–32767 *0–32767*	Version number of the operating system software that is loaded into the controller. This variable specifies the major version number of the controller's operating system.
Build Number *Build_Number*	0–32767 *0–32767*	Build number of the operating system software that is loaded into the controller. This variable specifies the minor version number of the controller's operating system.
SM Version *SM_Ver*	0–327.67 *0–32767*	Version number of the Start Manager software that is loaded into the controller.
Param Blk Version *Param_Blk_Ver*	0–327.67 *0–32767*	Version number of the parameter block that is loaded into the controller.
VCL App Version *VCL_App_Ver*	0–327.67 *0–32767*	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.

5 INITIAL SETUP

The 1236/38 controller 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 your application. The first step is to contact Curtis.

CONTACT CURTIS
Before you can use your controller, Curtis must set its application and motor parameter defaults to specifically match your application. You will need to let us know exactly what motor you will be using. We will then contact the motor manufacturer to obtain the information necessary to set these defaults. You should contact us as soon as you know what motor you will be using, as some time is required for us to obtain the necessary information and tailor your controller.

Once Curtis has set your controller's defaults, you can start conducting the setup procedures. First, **jack the vehicle drive wheels up off the ground** so that they spin freely. Doublecheck all wiring to ensure it is consistent with the wiring guidelines presented in Section 2. Make sure all connections are tight.

Turn on the controller and plug in the 1311 programmer.

① **Motor encoder** *(see page 46)*

Set the Encoder Steps parameter to the correct setting for your motor's position encoder.

② **Motor temperature sensor** *(see page 47)*

Set the Sensor Type parameter to the predefined type (1–5) that corresponds to your motor temperature sensor. Typically, the motor temperature sensor will be a thermistor that should be connected from Analog 2 (pin 8) to ground (pin 7) as described on page 19. When connected like this, the Current Source must be set to On.

If your motor temperature sensor is not in the list of predefined types, you will have to set Sensor Type = 0 (for user-defined sensor) and then set the five sensor voltage-temp parameter pairs to match the thermistor used.

To check whether the parameter settings and the motor thermistor connections yield the correct motor temperature, read the Temperature value displayed in the 1311's Monitor»Motor menu (page 59). If the 1311 does not display the correct motor temperature, contact your Curtis applications engineer for help. If the correct motor temperature is not displayed, or if there is no motor temperature sensor, this setup procedure can continue only if the Sensor Enable is set to Off.

If the 1311 displays the correct motor temperature, continue with the procedure and set up the Sensor Enable, Temperature Hot, and Temperature Max parameters.

③ **Current limits** *(see page 37)*

The Drive, Regen, and Brake 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 three current limit parameters to your desired values.

④ **Battery** *(see page 49)*

Set the Nominal Voltage parameter to match the nominal battery pack voltage of your system.

⑤ **Main contactor** *(see page 43)*

Set up the parameters in the Main Contactor Menu.

⑥ **Throttle** *(see pages 11–15 and 40–41)*

Before the throttle can be set up the interlock must be verified as Off, by reading the Interlock value displayed in the Monitor»Inputs menu (page 56). If the 1311 indicates the interlock is On, review how you set the Interlock Type parameter (Main Contactor Menu) and turn the interlock off. Verify that the 1311 displays that the interlock is now Off. Contact your Curtis applications engineer to resolve any issues about the interlock before continuing with the setup procedure.

Once you have verified the interlock is off, you can set up the throttle input. The Throttle Type parameter must be set to match the type of throttle (1–5) and wiring that you are using, as described on pages 11–15. Adjust the Forward Deadband, Forward Max, Reverse Deadband and Reverse Max parameters to match the range of your throttle. The Throttle Pot value displayed on the Monitor»Inputs menu (page 56) 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 other parameters in the Throttle Menu as required by the application.

You will be able to verify that your throttle settings are correct by checking the Mapped Throttle value displayed in the Monitor»Inputs menu (page 56) 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 forward throttle max and should be = -100% through the range considered reverse throttle max. Contact your Curtis applications engineer to resolve any issues about the throttle setup before continuing with the setup procedure.

⑦ Brake *(see page 42)*

If the brake function is not used by your application, set the Brake Enable parameter = Off and Brake Type = 5.

Before the brake can be set up the interlock must be verified as Off, by reading the Interlock value displayed in the Monitor»Inputs menu (page 56). If the 1311 indicates the interlock is On, review how you set the Interlock Type parameter (Main Contactor Menu) and turn the interlock off. Verify that the 1311 Monitor Menu displays that the interlock is now Off. Contact your Curtis applications engineer to resolve any issues about the interlock before continuing with the setup procedure.

Once you have verified that the interlock is off, you can set up the brake input. The Brake Type parameter must be set to match the type of brake throttle (1–5) and wiring that you are using, as described on pages 11–14. Adjust the Brake Deadband and Brake Max parameters to match the range of your brake pot. The Brake Pot value displayed in the Monitor»Inputs menu (page 56) 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 and enter in these values for the deadband and max settings. Set up the other parameters in the Brake Menu as required by the application.

You will be able to verify that your brake settings are correct by checking the Mapped Brake value displayed in the Monitor»Inputs menu (page 56) 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 max. Contact your Curtis applications engineer to resolve any issues about the brake setup before continuing with the setup procedure.

⑧ Faults *(see Section 7)*

Turn the KSI input Off and then On (to clear any parameter change faults) and use the 1311 to check for faults in the controller. All faults must be cleared before continuing with the setup procedure. Use Section 7 for help in trouble-shooting. Contact your Curtis applications engineer to resolve any issues about the faults before continuing with the setup procedure.

⑨ Setting encoder direction and direction of rotation *(see page 46)*

With the vehicle drive wheels still jacked up, no faults present in the controller, the interlock Off (as verified in the Monitor»Inputs menu, page 56), and both the throttle and brake in neutral (Mapped Throttle = 0% and Mapped Brake = 0% in the Monitor»Inputs menu), the encoder direction can be checked. Use the Monitor»Motor menu (page 59) 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 you get a positive Motor RPM when you rotate the motor in the forward direction, and a negative Motor RPM when you rotate the motor in the reverse direction, the Swap Encoder Direction parameter is correct and should not be changed. If you are getting negative Motor RPM when rotating the motor forward, the Swap Encoder Direction parameter must be changed. Cycle KSI power and repeat the procedure until you are satisfied that the Swap Encoder Direction setting is correctly set. 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 your Curtis applications engineer to resolve any issues ahout encoder direction or emergency reverse before continuing with the setup procedure.

Now that you have the encoder direction set correctly, you can test to see which direction the motor will spin due to how the three phase connections (**U**, **V**, and **W**) to the motor are connected. First the following parameter settings need to be verified:

Control Mode Select = 2 (page 26)
Torque Control Mode/ Max Speed = 1000 RPM (page 32)
Torque Control Mode/ Ki = 0 (page 32)

Next, cycle KSI input Off and then On (to clear any parameter change faults) and use the 1311 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 the 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, but it may turn in the wrong direction. Observe the direction of rotation of the motor and if it is turning in the wrong direction return the throttle to neutral, and change the setting of the Swap Two Phases parameter. Cycle power, turn on interlock, and turn on direction. Apply throttle and verify that the direction of rotation of the motor matches the direction input. If the motor does not respond properly you should contact your Curtis applications engineer to resolve any issues about encoder direction or emergency reverse before continuing with the setup procedure.

CAUTION

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

Once the motor is responding properly, lower the vehicle to put the drive wheels on the ground.

at Speed incrementally until the Motor RPM is able to obtain the Max Speed.

d. If fast torque oscillations are felt at maximum speed you should decrease the Kp value. The higher the Kp value, the faster the motor control will respond to exceeding Max Speed. If Kp is set too high, fast torque oscillations can be felt at maximum speed.

e. Ki (the integrator term) forces the motor to stay below the set Max Speed. On many applications, 0% is an acceptable value for Ki. To determine whether Ki is set acceptably for your application, find a hill and drive the vehicle down the hill at maximum speed. If you notice a surging feel as the motor exceeds the Max Speed and then gradually brings the speed back down to the set Max Speed, Ki is set too low. Setting Ki too high will likely cause motor oscillations even on flat ground. These oscillations are similar to those caused by Kp, but usually slower (lower frequency).

f. Set the Response Menu parameters as described on page 33.

g. Set the Restraint Menu parameters as described on page 36.

These tuning steps are for drive motors. If your controller is being used with a pump motor, set the parameter Pump Enable = On (page 31), and then set the Proportional Driver Menu parameters (page 44).

6

VEHICLE CONTROL LANGUAGE (VCL)

Curtis 1236/38 AC induction controllers are the first motor controllers in the industry with built-in programmable logic controllers with application-specific functions. VCL (Vehicle Control Language) software provides a way to implement unique and complex vehicle control functions on 1236/38 controllers.

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 two important manuals on your PC—the VCL Programmer's Guide and the VCL Common Functions Manual. These two manuals, which are in PDF format, include more detailed information about VCL than is in this 1236/38 manual.

This chapter of the 1236/38 manual summarizes VCL and also describes aspects and functions of VCL that are unique to the 1236/38 product. For a more complete understanding of the functions and capabilities of VCL, see the 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 1311 parameter Forward Offset.

- Functions are followed by parentheses; for example:
  Reset_Controller() is a function
  Reset_Voltage is a variable.

- The prefix "@" specifies a location (or source or address) rather than a value; for example:
  "@setpoint" is the location inside the processor where the setpoint variable is stored
  "setpoint" is the value of the setpoint 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 on 1236/38 controllers. In addition, the 1236/38 controllers have these functions developed specifically for them:

ENABLE_PRECHARGE()........... p. 90
DISABLE_PRECHARGE().......... p. 91
SET_DIGOUT()........................... p. 92
CLEAR_DIGOUT()...................... p. 92
ENABLE_EMER_REV().............. p. 93
DISABLE_EMER_REV()............. p. 93
SET_INTERLOCK()..................... p. 94
CLEAR_INTERLOCK()............... p. 94
SETUP_POT_FAULTS() .... p. 96

These functions, which are not included in the VCL Common Functions Manual, are described at the end of this chapter.

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 stored at power-down and recalled by the operating system when the NVM_NVUser_Restore function is used. Thus, they can then be recalled at the next power-on cycle, which restores their previous values. See the section on non-volatile memory access in the VCL Common Functions manual for more information.

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 1311 parameters. These 1311 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_UserBit variables. These variables are typically written to EEPROM through the 1311 programmer interface (i.e., when a 1311 user changes a parameter setting using the 1311). They can be used in the VCL code, but changing a P_User (or P_UserBit) 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 1311 parameters only.

TYPE	QUANTITY	RANGE
RAM	120 variables	User1 – User120
NVUser EEPROM	15 variables	NVUser1 – NVUser15
Block EEPROM	38 blocks (15 variables each)	NVM3 – NVM40
Parameters EEPROM	100 variables and 20 variables of 8 bits each (160 bits)	P_User1 – P_User100 P_UserBit1 – P_UserBit20

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:

FUNCTION	FUNCTION FULL NAME	INSTANCES	SERVICE RATE
ABS	Absolute Value	2	4 ms
ADC	Analog to Digital Converter Input	2	1 ms
CAN	CAN Communications	12	4 ms
CPY	Copy	8	4 ms
DLY	Delay	16	1 ms
FLT	Filter	4	1 ms
LIM	Limit	4	4 ms
MAP	Map	4	4 ms
MTD	Multiply then Divide	4	4 ms
NVM	Non-Volatile Memory	38	2 ms
PID	Proportional Integral Derivative	2	1 ms
POT	Potentiometer Input	2	4 ms
PWM	Pulse Width Modulated Output	6	4 ms
RMP	Ramp	4	1 ms
SCL	Scaling	4	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 15 associated bit-variables.

I/O CONTROL WITH VCL

Digital Inputs

The 1236/38 controller has 15 digital inputs. Eight are switch inputs (Sw_1 through Sw_8), shown on the standard wiring diagram (Figure 3, page 10). The other seven 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 (i.e., open coil check).

To address a digital input in a VCL program, use the desired input label (Sw_1 through Sw_15). You must 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_15 = OFF)
    {
    ;put code here to run when switch 15 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 your VCL code. The debounce time can be varied from 0 to 32 milliseconds in 4ms 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.

Digital Outputs

The 1236/38 controller has two types of digital outputs: driver and digital. <u>Driver outputs</u> 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. <u>Digital outputs</u> are low current NPN transistor drivers that are only On or Off. They work well for driving LEDs or for interfacing to another digital device.

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 “high power outputs” on page 18 and “digital outputs” on page 17.

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) is different from Drivers 1–4. It can be controlled in two ways: with the PID software, 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. See page 69 for more information on interfacing the proportional driver.

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

The 1236/38 controller has two potentiometer inputs, which are typically used for throttle and brake. Many features (mapping, acceleration rates, etc.) are built in as 1311 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 throttle pot (or brake pot) information is to set the 1311 parameter Throttle Type (or Brake Type) to an appropriate value of 1–4 as shown on pages 11–14. 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 1311 parameter Throttle Type (or Brake Type) should be set to a value of 5 (page 17). Setting the 1311 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 13 (page 78).

When the Throttle Type setting is = 5, the Throttle_Pot_Output is a VCL variable that the OS will update with the current value of the throttle pot input. Similarly, when the Brake Type setting is = 5, the Brake_Pot_Output is a VCL variable that the OS will update with the current value of the brake pot input. However, the value of the Throttle_Pot_Output (or Brake_Pot_Output) will remain clamped to = 0 until the VCL function Setup_Pot() is executed.

Typically the Setup_Pot() function is executed at the beginning of a VCL program to define the potentiometer input connection 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 on page 15 for a Throttle Type 2. TWO_WIRE potentiometer connections are the same as the 2-wire potentiometer connections shown on page 14 for a Throttle Type 1. ONE_WIRE potentiometer connections are the same as the Voltage Source or Current Source connections shown on page 15 for a 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 Thottle_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).

The 0–100% position of the potentiometer is represented by a value from 0–32767 in VCL. Once set up (through the VCL Setup_Pot() function) the potentiometer value is automatically and continuously loaded into the variable Throttle_Pot_Output or Brake_Pot_Output. 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 setting = 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, a Brake Type = 5 means that 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 set the high and low threshold 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). See page 95 for more detail on this function.

Analog Inputs

The 1236/38 controller has 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 = 0–32767.

```
User2 = Analog2_Input
```

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

Analog Output

The 1236/38 controller has 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 Put_PWM() and Automate_PWM() used by these other drivers. The scaling is 0–10V = 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 CONTROLLER'S 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 1236/38 controller are both sophisticated and flexible. Before applying VCL to modify these chains, it is important to fully understand the ramifications of these changes. The motor command diagram is presented in Figure 13.

Fig. 13 *Motor command diagram.*

Throttle Processing

The top of Figure 13 shows the throttle processing section. The throttle signal chain flows left to right starting with the physical throttle pot. The voltage on the throttle wiper input (pin 16) is input into the controller and has the VCL variable name Throttle_Pot_Raw which is displayed in the 1311 Monitor Menu. 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 (see page 40) 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 (see page 40) 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 is = 0.

The signal then passes through a selector switch. If the Throttle_Type parameter is set to 5 (Throttle Type = VCL input, see page 40), 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 output of the multiplying and summing nodes is a VCL variable called Mapped_Throttle, which is displayed in the 1311 Monitor Menu. Checking the value of Mapped_Throttle using the 1311 is a very good way to see if your Throttle Menu parameters are set correctly. A VCL program can control the throttle by changing the variables VCL_Throttle (only if Throttle Type = 5), Throttle_Multiplier, and Throttle_Offset. The effect of these variables can be observed on the Mapped_Throttle 1311 Monitor Menu.

Mapped_Throttle next passes through a limiter that is active only if a pump motor is being operated (Pump Enable = On, see page 31). When active, this function limits the throttle signal to only positive (forward) motion, which is appropriate for controlling a hydraulic pump motor.

The throttle signal continues to a selector switch that will set the throttle signal = 0% if any of the following conditions is present: Interlock_State = Off (see page 56), a fault has set Throttle request = 0% (see the Troubleshooting Chart, Table 5), or if Main_State ≠ 5 or 10 (see page 60).

After this selector switch the throttle signal is a VCL variable called Throttle_Command, which is displayed in the 1311 Monitor Menu. Throttle_Command is the final value of the throttle signal chain that is input to the Control Mode Processing block; see Figure 14. Checking the value of Throttle_Command using the 1311 is a very good way to see the final throttle signal. If Throttle_Command is non-zero, the motor controller will output signals to the motor to make it spin.

For investigating why a motor is <u>not</u> spinning, it is useful to use the 1311 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 13) 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	ACCESS	DESCRIPTION
Throttle_Pot_Raw	Read Only	Voltage measurement at pin 16, scaled for the proper wiring
Throttle_Pot_Output	Read Only	Throttle pot input value after being scaled for the proper wiring. For use in VCL program when Throttle Type = 5.
Mapped_Throttle	Read Only	Throttle signal value after mapping
Throttle_Command	Read Only	Command resulting from throttle processing
VCL_Throttle	Read/Write	VCL-accessible throttle command
Throttle_Multiplier	Read/Write	Multiplies or divides the throttle signal
Throttle_Offset	Read/Write	Provides a +/- offset to the throttle signal

Brake Processing

Brake processing is optional as it can be turned Off (by setting Brake_Pedal_Enable = Off, see page 42). If turned On, brake processing can be done with or without VCL. Any non-zero brake command will then override the throttle signal and the motor controller will brake to a stop as determined by the parameters Brake Current Limit (page 37) and Brake Taper Speed (page 36).

The lower part of Figure 13 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 input into the controller and has the VCL variable name Brake_Pot_Raw which is displayed in the 1311 Monitor 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 (page 42) and the brake potentiometer input (Brake_Pot_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 (page 42).

The signal then passes through a selector switch. If the Brake_Type parameter is set to 5 (Brake Type = VCL input, see page 42), 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 position or 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 is a VCL variable called Mapped_Brake, which is displayed in the 1311 Monitor Menu. Checking the value of Mapped_Brake using the 1311 is a very good way to see if your 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 effect of this variable can also be observed on the Mapped_Brake 1311 Monitor Menu.

The brake signal then goes through a second selector switch that will set the brake signal = 0% if the Brake Pedal Enable parameter (page 42) is set Off. If set On then the brake signal will pass through. The brake signal after this second selector switch is a VCL variable called Brake_Command, which is displayed in the 1311 Monitor Menu.

Brake_Command is the final value of the brake signal chain that is input to the Control Mode Processing block; see Figure 14. Checking the value of Brake_Command using the 1311 is a very good way to see the final brake signal. If Brake_Command is non-zero, the throttle signal will be set to 0% (see Figure 13) and the motor controller will use the parameters Brake Current Limit and Brake Taper Speed to bring the motor to a stop.

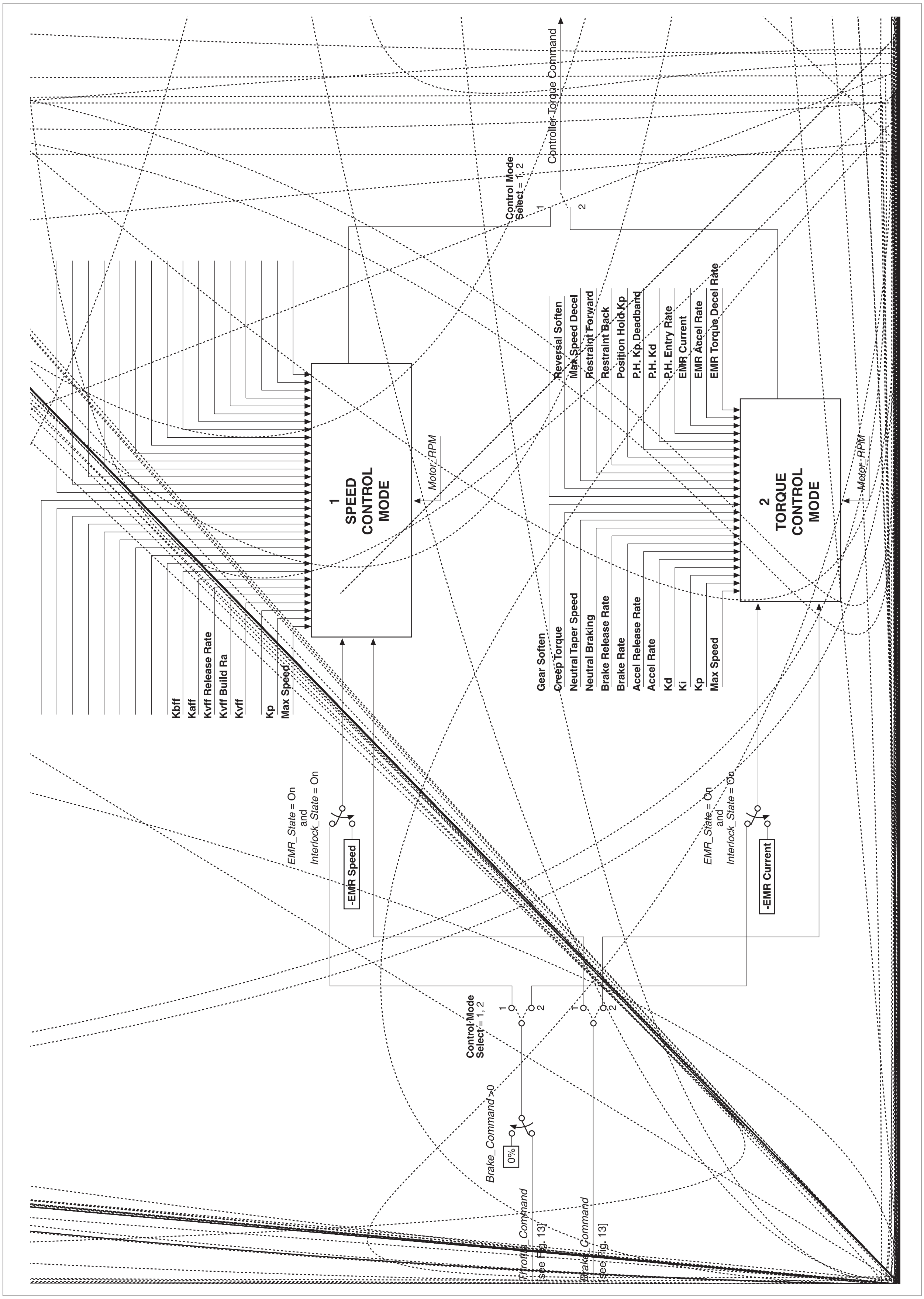


Fig. 14 *Control Mode Mapping.*

The following brake processing variables are accessible by VCL:

VCL VARIABLE	ACCESS	DESCRIPTION
Brake_Pot_Raw	Read Only	Voltage measurement at pin 17
Brake_Pot_Output	Read Only	Brake pot input value after being scaled for the proper wiring
Mapped_Brake	Read Only	Brake pot value after mapping
VCL_Brake	Read/Write	VCL-accessible brake command
Brake_Command	Read Only	Command resulting from brake processing

Control Mode Mapping and Motor Control Processing

Figure 14 begins with the Throttle_Command and Brake_Command inputs and a switch that will zero the Throttle_Command if Brake_Command is any value but 0%. The signal chains are then directed to Speed Control, Torque Control, or Pump Control based on Control Mode Select. Note that the emergency reverse function is handled differently depending on the control mode selected. In speed control mode, EMR Speed sets the throttle signal while in torque control the throttle signal is set by EMR Current. In pump control mode there is no emergency reverse.

The control mode function uses algorithms to convert the incoming throttle and brake signals and the motor RPM input into a Controller Torque Command.

VCL can modify the 1311 control mode parameters in RAM by using the VCL variable name for the 1311 parameter. For example,

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

will change the RAM value of the torque control mode's Brake Rate; the new value will be used in determining the Controller Torque Command. However, the value of the stored EE value of this parameter 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. VCL cannot write to the EE memory. The 1311 parameter settings in EE memory can be changed by using the 1311 to change the values in the program menus.

The selected control mode calculates the desired Controller Torque Command, which is passed to the Motor Control block (see Figure 13). The Motor Control block uses its mathematical model of the specific AC induction motor used to generate the high efficiency three-phase outputs that are output to the AC motor via the cables connected to the **U**, **V**, and **W** terminals.

INTERFACING THE PROPORTIONAL CURRENT DRIVER

VCL code can directly interface the 1236/38's proportional current driver (PD); see Figure 15. VCL can change the working parameters of the PD and can provide the command.

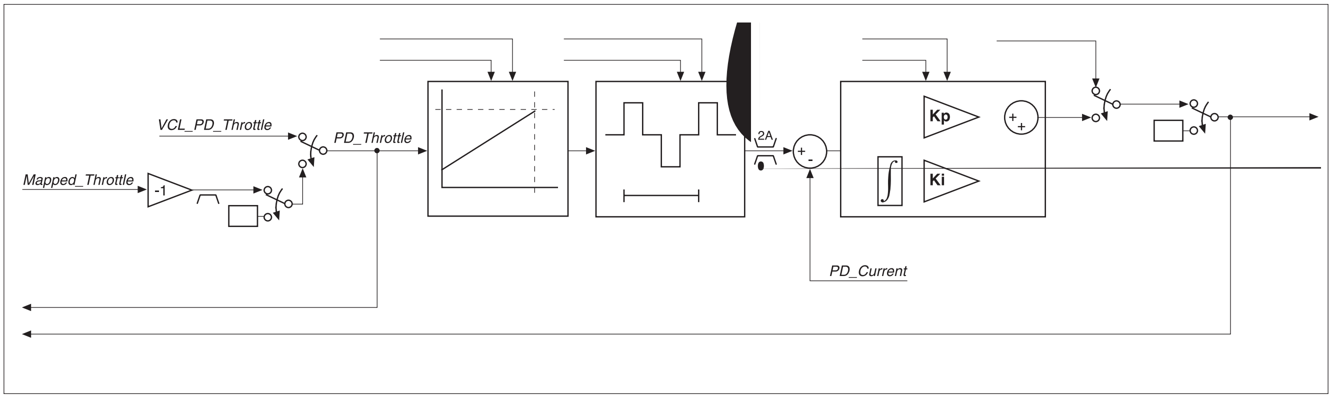


Fig. 15 *Proportional driver processing.*

USING THE FAULT HANDLER IN VCL

The operating system of the controller 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, fault text is displayed on the 1311 System Faults and Fault History Menus. These operating system faults are covered in Section 7. Additionally, the operating system also makes the status of the operating system faults available for use in VCL programs in the form of five variables called Status1, Status2, Status3, Status4, and Status5. 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 Status1–5 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.

Here are the bit locations of each of the operating system faults in the Status1–5 variables:

```
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 = Brake Wiper Low (Code 44)
*      Bit6 = Brake Wiper High (Code 43)
*      Bit7 = EEPROM Failure (Code 46)

Status2
*      Bit0 = HPD/Seqencing Fault (Code 47)
*      Bit1 = Severe Undervoltage (Code 17)
*      Bit2 = Severe Overvoltage (Code 18)
*      Bit3 = Undervoltage Cutback (Code 23)
*      Bit4 = Overvoltage Cutback (Code 24)
*      Bit5 = Controller UnderTemp (Code 21)
*      Bit6 = Controller OverTemp Cutback (Code 22)
*      Bit7 = Controller Severe UnderTemp (Code 15)

Status3
*      Bit0 = Controller Severe OverTemp (Code 16)
*      Bit1 = Coil1 Driver Open/Short (Code 31)
*      Bit2 = Coil2 Driver Open/Short (Code 32)
*      Bit3 = Coil3 Driver Open/Short (Code 33)
*      Bit4 = Coil4 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 Runtime 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 Detect (Code 73)
```

The 1236/38 operating system also provides the capability to create OEM-defined custom faults using VCL. Just like the system faults, the VCL faults result in flashed codes on the controller Status LEDs and fault text is displayed on the 1311 System Faults and Fault History Menus. Optionally, the VCL can assign fault actions to occur automatically when the associated fault is set. There are 16 VCL faults available and they are stored in the VCL variables UserFault1 and UserFault2. The UserFault1,2 variables are Read/Write (R/W) and the 16 faults are stored in the lower byte of each variable like this:

```
UserFault1
*      Bit0  = VCLfault1 (Code 51)
*      Bit1  = VCLfault2 (Code 52)
*      Bit2  = VCLfault3 (Code 53)
*      Bit3  = VCLfault4 (Code 54)
*      Bit4  = VCLfault5     u   /54)
*        it4  = VC  fault2 (Code     /54)
*        it4  = VC  fault3 (Code     /54)
*        it4  = VC  h  ak 7                              UserFa     /
*             = (Cod/    /
*          /    = (Code              /
*          /    = (Code              /
*          /    = (Co               /
*          /    = (Co               /
*          /    = (Code        /
*          /    = (Code        /
*          /    = kha        o  /  T       t       e  p  T  mc k  T oO
```

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 1311 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 16 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. Here are the fault actions available in the User_Fault_Actions_xx variables:

```
           VARIABLE                              FAULT ACTION
User_Fault_Actions_xx
*      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 = Not Used
*      Bit12 = Not Used
*      Bit13 = Not Used
*      Bit14 = Not Used
*      Bit15 = Not Used
```

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 1311 System Faults and Fault History Menus.

To add the fault text on the 1311 System Faults and Fault History menus it is necessary to create a Fault Definition for the 1311. Creating fault definitions is a subject that is covered in detail in the VCL Programmer’s Guide (Chapter 5, Support for the 1311 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
       {
       UserError1.2 = 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 1311 System Faults menu (only while the fault is set) and this fault will be logged into the 1311 Fault History menu after being set. The text displayed in either of these 1311 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 1311 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. If you wish 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).

VCL FUNCTIONS SPECIFIC TO THE 1236/38 AC CONTROLLER

Function descriptions are provided here for the functions that are unique to the 1236/38. 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. 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 3 volts of KSI, or one second 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.

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 aborted.
1 – Precharge successfully aborted.

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).

The low power digital output is protected from excessive current (current over 15 mA); a fault will occur when this current has been exceeded, and the driver will be shut off (open). Running the function again will reactivate the driver, and will attempt to clear the fault.

Syntax **`Set_Digout(DigOut_ID)`**

Parameters

`DigOut_ID` is the low power digital I/O 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 low power digital I/O 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 1311 EMR Type 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; see page 53. To view the current emergency reverse state, see the 1311 menu Monitor » Inputs: Emer Rev.

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 1311 EMR Type 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 the 1311 menu Monitor » Inputs: Emer Rev.

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 1311 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 13. Additionally, if the main contactor is used (1311 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 1311 menu Monitor » Inputs: Interlock. To view the current main contactor state, see the 1311 menu 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 1311 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 13. Additionally, if the main contactor is used (1311 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 the 1311 menu Monitor » Inputs: Interlock. The view the current main contactor state, see the 1311 menu 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_FAULTS()

This function sets the upper and lower wiper fault voltages for a given pot input and sets the replacement wiper voltage value that will be used if there is a fault. The valid range for the function parameters is 0–6.25 V (0–400 counts). If this function is not run, the default thresholds depend on the 1311 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	0.1 V	5.5 V
2	none	5.5 V
3	0.1 V	5.5 V
4	0.1 V	5.5 V
5 (ONE_WIRE)	none	5.5 V
5 (TWO_WIRE)	0.1 V	5.5 V
5 (THREE_WIRE)	0.1 V	5.5 V

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: 1 V = 64 counts.

`High_Fault`
Specifies the upper threshold voltage limit.
Scaling: 1 V = 64 counts.

`Fault_Value`
The value that is used for the pot input when there is a fault (0–32767).
Scaling: ±32767 = ±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 at 0.3 volts (19/64) and the upper pot voltage at 5.0 volts (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%.

7

DIAGNOSTICS AND TROUBLESHOOTING

The 1236/38 controller detects a wide variety of faults or error conditions. Faults can be detected by the operating system or by the VCL code. This section describes the faults detected by the operating system.

Faults detected by VCL code (faults 51–67 in Table 5) 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 1311 programmer or (2) by observing the fault codes issued by the Status LEDs. See Table 4 for a summary of LED display formats.

The 1311 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 1311 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: Battery Undervoltage (code 23).

In the Fault menu of the 1311 programmer, the words **Undervoltage Cutback** will be displayed; the real-time battery voltage is displayed in the Monitor menu (“Keyswitch 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 5), which also lists possible fault causes and describes the conditions that set and clear each fault.

Summary of LED display formats

The two LEDs have four different display modes, indicating the type of information they are providing.

Table 4 TYPES OF LED DISPLAY

DISPLAY	STATUS
Neither LED illuminated	Controller is not powered on, has a dead battery, or is severely damaged.
Yellow LED flashing	Controller is operating normally.
Yellow and red LEDs both on solid	Controller is in Flash program mode.
Red LED on solid	Watchdog failure. Cycle KSI to restart.
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 5, 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 not, shut off KSI and remove the 35-pin connector. Check the connector for corrosion or damage, clean it if necessary, and re-insert it.

Table 5 TROUBLESHOOTING CHART

CODE	PROGRAMMER LCD DISPLAY / *EFFECT OF FAULT*	POSSIBLE CAUSE	SET/CLEAR CONDITIONS
12	Controller Overcurrent *ShutdownMainContactor; ShutdownMotor; ShutdownEMBrake.*	1. External short of phase U,V, or W motor connections. 2. Motor parameters are mis-tuned. 3. Controller defective.	*Set:* Phase current exceeded the current measurement limit. *Clear:* Cycle KSI.
13	Current Sensor Fault *ShutdownMainContactor; ShutdownMotor; ShutdownEMBrake.*	1. Leakage to vehicle frame from phase U, V, or W (short in motor stator). 2. Controller defective.	*Set:* Controller current sensors have invalid offset reading. *Clear:* Cycle KSI.
14	Precharge Failed *ShutdownMainContactor; ShutdownMotor; ShutdownEMBrake.*	1. External load on capacitor bank (B+ connection stud) that prevents the capacitor bank from charging. 2. See 1311 menu Monitor » Battery: Capacitor Voltage.	*Set:* Precharge failed to charge the capacitor bank to the KSI voltage. *Clear:* Cycle Interlock input or use VCL function *Precharge()*.
15	Controller Severe Undertemp *ShutdownMainContactor; ShutdownMotor; ShutdownEMBrake; ShutdownThrottle; FullBrake.*	1. Controller is operating in an extreme environment. 2. See 1311 menu Monitor » Controller: Temperature.	*Set:* Heatsink temperature below -40°C. *Clear:* Bring heatsink temperature above -40°C, and cycle interlock or KSI.
16	Controller Severe Overtemp *ShutdownMainContactor; ShutdownMotor; ShutdownEMBrake; ShutdownThrottle; FullBrake.*	1. Controller is operating in an extreme environment. 2. Excessive load on vehicle. 3. Improper mounting of controller. 4. See 1311 menu Monitor » Controller: Temperature.	*Set:* Heatsink temperature above +95°C. *Clear:* Bring heatsink temperature below +95°C, and cycle interlock or KSI.
17	Severe Undervoltage *Reduced drive torque.*	1. Battery Menu parameters are misadjusted. 2. Non-controller system drain on battery. 3. Battery resistance too high. 4. Battery disconnected while driving. 5. See 1311 menu Monitor » Battery: Capacitor Voltage. 6. Blown B+ fuse or main contactor did not close.	*Set:* Capacitor bank voltage dropped below the Severe Battery Undervoltage limit (see page 49) with FET bridge enabled. *Clear:* Bring capacitor voltage above Severe Battery Undervoltage limit.
18	Severe Overvoltage *ShutdownMainContactor; ShutdownMotor; ShutdownEMBrake; ShutdownThrottle; FullBrake.*	1. Battery Menu parameters are misadjusted. 2. Battery resistance too high for given regen current. 3. Battery disconnected while regen braking. 4. See 1311 menu Monitor » Battery: Capacitor Voltage.	*Set:* Capacitor bank voltage exceeded the Severe Battery Overvoltage limit (see page 49) with FET bridge enabled. *Clear:* Bring capacitor voltage below Severe Battery Overvoltage limit, and then cycle KSI.
21	Controller Undertemp Cutback *None, unless a fault action is programmed in VCL.*	1. Controller is performance-limited at this temperature. 2. Controller is operating in an extreme environment. 3. See 1311 menu Monitor » Controller: Temperature.	*Set:* Heatsink temperature dropped below -25°C. *Clear:* Bring heatsink temperature above -25°C.

Table 5 TROUBLESHOOTING CHART, continued

CODE	PROGRAMMER LCD DISPLAY *EFFECT OF FAULT*	POSSIBLE CAUSE	SET/CLEAR CONDITIONS
22	Controller Overtemp Cutback *Reduced drive and brake torque.*	1. Controller is performance-limited at this temperature. 2. Controller is operating in an extreme environment. 3. Excessive load on vehicle. 4. Improper mounting of controller. 5. See 1311 menu Monitor » Controller: Temperature.	*Set:* Heatsink temperature exceeded 85°C. *Clear:* Bring heatsink temperature below 85°C.
23	Undervoltage Cutback *Reduced drive torque.*	1. Normal operation. Fault shows that 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 1311 menu Monitor » Battery: Capacitor Voltage. 7. Blown B+ fuse or main contactor did not close.	*Set:* Capacitor bank voltage dropped below the Battery Undervoltage limit (see p. 49) with the FET bridge enabled. *Clear:* Bring capacitor voltage above the Undervoltage limit.
24	Overvoltage Cutback *Reduced brake torque.*	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 1311 menu Monitor » Battery: Capacitor Voltage.	*Set:* Capacitor bank voltage exceeded the Battery Overvoltage limit (see page 49) with the FET bridge enabled. *Clear:* Bring capacitor voltage below the Overvoltage limit.
25	+5V Supply Failure *None, unless a fault action is programmed in VCL.*	1. External load impedance on the +5V supply (pin 26) is too low. 2. See 1311 menu Monitor » outputs: 5 Volts and Ext Supply Current.	*Set:* +5V supply (pin 26) outside the +5V±10% range. *Clear:* Bring voltage within range.
26	Digital Out 6 Overcurrent *Digital Output 6 driver will not turn on.*	1. External load impedance on Digital Output 6 driver (pin 19) is too low.	*Set:* Digital Output 6 (pin 19) current exceeded 15 mA. *Clear:* Remedy the overcurrent cause and use the VCL function *Set_DigOut()* to turn the driver on again.
27	Digital Out 7 Overcurrent *Digital Output 7 driver will not turn on.*	1. External load impedance on Digital Output 7 driver (pin 20) is too low.	*Set:* Digital Output 7 (pin 20) current exceeded 15 mA. *Clear:* Remedy the overcurrent cause and use the VCL function *Set_DigOut()* to turn the driver on again.

Table 5 TROUBLESHOOTING CHART, continued

CODE	PROGRAMMER LCD DISPLAY *EFFECT OF FAULT*	POSSIBLE CAUSE	SET/CLEAR CONDITIONS
28	Motor Temp Hot Cutback *Reduced drive torque.*	1. Motor temperature is at or above the programmed Temperature Hot setting, and the requested current is being cut back. 2. Motor Temperature Control Menu parameters are mis-tuned. 3. See 1311 menus Monitor » Motor: Temperature and Monitor » Inputs: Analog2. 4. If the application doesn't use a motor thermistor, Temp Compensation and Temp Cutback should be programmed Off.	*Set:* Motor temperature is at or above the Temperature Hot parameter setting. *Clear:* Bring the motor temperature within range.
29	Motor Temp Sensor Fault *MaxSpeed reduced (LOS, Limited Operating Strategy) and motor temperature cutback is disabled.*	1. Motor thermistor is not connected properly. 2. If the application doesn't use a motor thermistor, Temp Compensation and Temp Cutback should be programmed Off. 3. See 1311 menus Monitor » Motor: Temperature and Monitor » Inputs: Analog2.	*Set:* Motor thermistor input (pin 8) is at the voltage rail (0 or 10V). *Clear:* Bring the motor thermistor input voltage within range.
31	Coil1 Driver Open/Short *ShutdownDriver1.*	1. Open or short on driver load. 2. Dirty connector pins. 3. Bad crimps or faulty wiring.	*Set:* Driver 1 (pin 6) is either open or shorted. *Clear:* Correct open or short, and cycle driver.
31	Main Open/Short *ShutdownDriver1; ShutdownMotor; ShutdownEMBrake.*	1. Open or short on driver load. 2. Dirty connector pins. 3. Bad crimps or faulty wiring.	*Set:* Main contactor driver (pin 6) is either open or shorted. *Clear:* Correct open or short, and cycle driver.
32	32 3	t	either open or shorted.

Table 5 TROUBLESHOOTING CHART, continued

CODE	PROGRAMMER LCD DISPLAY *EFFECT OF FAULT*	POSSIBLE CAUSE	SET/CLEAR CONDITIONS
37	Motor Open *ShutdownMainContactor; ShutdownMotor; ShutdownEMBrake.*	1. Motor phase is open. 2. Bad crimps or faulty wiring. 3. Bad crimps or faulty wiring.	*Set:* Motor phase U, V, or W deteected open. *Clear:* Cycle KSI.
38	Main Contactor Welded *ShutdownMainContactor; ShutdownMotor; ShutdownEMBrake.*	1. Main contactor tips are welded closed. 2. Motor phase U 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 stud).	*Set:* Just prior to the main contactor closing, the capacitor bank voltage (B+ connection stud) was loaded for a short time and the voltage did not discharge. *Clear:* Cycle KSI.
39	Main Contactor Did Not Close *ShutdownMainContactor; ShutdownMotor; ShutdownEMBrake.*	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 stud) that prevents capacitor bank from charging. 4. Blown B+ fuse.	*Set:* With the main contactor commanded closed, the capacitor bank voltage (B+ connection stud) did not charge to B+. *Clear:* Cycle KSI.
41	Throttle Wiper High *ShutdownThrottle.*	1. Throttle pot wiper voltage too high. 2. See 1311 menu Monitor » Inputs: Throttle Pot.	*Set:* Throttle pot wiper (pin 16) voltage is higher than the high fault threshold (can be changed with the VCL function Setup_Pot_Faults()). *Clear:* Bring throttle pot wiper voltage below the fault threshold.
42	Throttle Wiper Low *ShutdownThrottle.*	1. Throttle pot wiper voltage too low. 2. See 1311 menu Monitor » Inputs: Throttle Pot.	*Set:* Throttle pot wiper (pin 16) voltage is lower than the low fault threshold (can be changed with the VCL function Setup_Pot_Faults()). *Clear:* Bring throttle pot wiper voltage above the fault threshold.
43	Brake Wiper High *FullBrake.*	1. Brake pot wiper voltage too high. 2. See 1311 menu Monitor » Inputs: Brake Pot.	*Set:* Brake pot wiper (pin 17) voltage is higher than the high fault threshold (can be changed with the VCL function Setup_Pot_Faults()). *Clear:* Bring brake pot wiper voltage below the fault threshold.
44	Brake Wiper Low *FullBrake.*	1. Brake pot wiper voltage too low. 2. See 1311 menu Monitor » Inputs: Brake Pot.	*Set:* Brake pot wiper (pin 17) voltage is lower than the low fault threshold (can be changed with the VCL function Setup_Pot_Faults()). *Clear:* Bring brake pot wiper voltage above the fault threshold.
45	Pot Low Overcurrent *ShutdownThrottle; FullBrake.*	1. Combined pot resistance connected to pot low is too low. 2. See 1311 menu Monitor » Outputs: Pot Low.	*Set:* Pot low (pin 18) current exceeds 10mA. *Clear:* Clear pot low overcurrent condition and cycle KSI.

Table 5 TROUBLESHOOTING CHART, continued

CODE	PROGRAMMER LCD DISPLAY / *EFFECT OF FAULT*	POSSIBLE CAUSE	SET/CLEAR CONDITIONS
46	EEPROM Failure *ShutdownMainContactor; ShutdownMotor; ShutdownEMBrake; ShutdownThrottle; ShutdownInterlock; ShutdownDriver1; ShutdownDriver2; ShutdownDriver3; ShutdownDriver4; ShutdownPD; FullBrake.*	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 1311, or by loading new software into the controller.	*Set:* Controller operating system tried to write to EEPROM memory and failed. *Clear:* Download the correct software (OS) and matching parameter default settings into the controller and cycle KSI.
47	HPD/Sequencing Fault *ShutdownThrottle.*	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 1311 menu Monitor » Inputs.	*Set:* HPD (High Pedal Disable) or sequencing fault caused by incorrect sequence of KSI, interlock, direction, and throttle inputs. *Clear:* Reapply inputs in correct sequence.
49	Parameter Change Fault *ShutdownMainContactor; ShutdownMotor; ShutdownEMBrake.*	1. This is a safety fault caused by a change in certain 1311 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.	*Set:* Adjustment of a parameter setting that requires cycling of KSI. *Clear:* Cycle KSI.
51–67	OEM Faults *(See OEM documentation.)*	1. These faults can be defined by the OEM and are implemented in the application-specific VCL code. See OEM documentation.	*Set:* See OEM documentation. *Clear:* See OEM documentation.
68	VCL Runtime Error *ShutdownMainContactor; ShutdownMotor; ShutdownEMBrake; ShutdownThrottle; ShutdownInterlock; ShutdownDriver1; ShutdownDriver2; ShutdownDriver3; ShutdownDriver4; ShutdownPD; FullBrake.*	1. VCL code encountered a runtime VCL error. 2. See 1311 menu Monitor » 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.	*Set:* Runtime VCL code error condition. *Clear:* Edit VCL application software to fix this error condition; flash the new compiled software and matching parameter defaults; cycle KSI.
69	External Supply Out of Range *None, unless a fault action is programmed in VCL.*	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 mis-tuned. 3. See 1311 menu Monitor » Outputs: Ext Supply Current.	*Set:* 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 Ext Supply Max and Ext Supply Min parameter settings (page 45). *Clear:* Bring the external supply current within range.

Table 5 TROUBLESHOOTING CHART, continued

CODE	PROGRAMMER LCD DISPLAY / *EFFECT OF FAULT*	POSSIBLE CAUSE	SET/CLEAR CONDITIONS
71	OS General *ShutdownMainContactor; ShutdownMotor; ShutdownEMBrake; ShutdownThrottle; ShutdownInterlock; ShutdownDriver1; ShutdownDriver2; ShutdownDriver3; ShutdownDriver4; ShutdownPD; FullBrake.*	1. Internal controller fault.	*Set:* Internal controller fault detected. *Clear:* Cycle KSI.
72	PDO Timeout *ShutdownInterlock; CAN NMT State set to Pre-operational.*	1. Time between CAN PDO messages received exceeded the PDO Timeout Period.	*Set:* Time between CAN PDO messages received exceeded the PDO Timeout Period. *Clear:* Cycle KSI.
73	Stall Detect *Control Mode changed to LOS (Limited Operating Strategy).*	1. Stalled motor. 2. Motor encoder failure. 3. Bad crimps or faulty wiring. 4. Problems with power supply for the motor encoder. 5. See 1311 menu Monitor » Motor: Motor RPM.	*Set:* No motor encoder movement detected. *Clear:* Either cycle KSI, or detect valid motor encoder signals while operating in LOS mode and return Throttle Command = 0 and Motor RPM = 0.

8 MAINTENANCE

There are no user serviceable parts in the Curtis 1236/38 controller. **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 diagnostics 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.

CAUTION

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.

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 Section 2, page 6, for maximum tightening torque specifications for the battery and motor connections.

DIAGNOSTIC HISTORY

The 1311 programmer can be used to access the controller's diagnostic history file. The programmer will read out all the faults the controller has experienced since the last time the diagnostic 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 diagnostic history file. This allows the controller to accumulate a new file of faults. By checking the new diagnostic history file at a later date, you can readily determine whether the problem was indeed fixed.

APPENDIX A

THEORY OF OPERATION

The Curtis 1236/38 controller converts DC battery power to 3-phase AC power by precisely controlling the induction drive for high bandwidth, high efficiency, and low ripple torque generation. To realize this level of precise torque control of induction motor drives in electric vehicles, Curtis engineers carefully evaluated and incorporated the latest technology in microprocessors, power electronics, and motor control.

Invented by Nikola Tesla in 1888, the induction motor became a workhorse that contributed to the vast industrial growth in the twentieth century. Until recently relegated to non-dynamic applications where transient response wasn't a critical concern, induction motors are now the motor of choice in high performance control applications. This shift was facilitated by the enormous advancements in microprocessors and power silicon devices in the last twenty years, coupled with intense research and development.

The 3-phase induction motor has three sets of distributed windings in the stator winding slots. The standard induction motor has a rotor with aluminum bars short-circuited by cast aluminum end-rings. There are no brushes, commutators, or slip-rings, and—unlike DC and synchronous motors—there is no need for permanent magnets or a separate current supply for the rotor. The brushless construction of the induction motor and the rugged rotor provide high reliability, fault tolerance, low maintenance, and low cost.

Three-phase sinusoidal voltages, electrically displaced by 120°, are applied to the phase windings to create the stator magnetic field. The field rotates at the stator voltage frequency times the number of pole pairs. This rotating stator field induces currents in the conductive rotor bars by transformer action which, in turn, create a second rotor magnetic field. The rotor field reacts to the stator field to generate torque. The differential speed, or slip frequency, between the stator field and rotor speed is critical to the torque and speed control of an induction motor.

Motor Control Algorithms

Two main approaches are commonly used for induction motor control: scalar control and vector control.

Scalar control (e.g., volts/Hz) modulates only the magnitude and frequency of the applied voltage or current. Although scalar control has the advantage of being simpler than vector control, it has poor dynamic response and lower operation efficiency. The various methods used to improve performance require extensive characterization of the motor and loads.

Vector control (e.g., indirect rotor flux orientation, stator flux orientation, etc.) manipulates the magnitude, frequency, and phase of the control variables to provide better control. The mathematical model of an induction motor is complex. Using a series of reference frame transformations, vector control simplifies the model to enable precise control of torque and flux, similar to a SepEx motor controller.

Figure A-1 shows a typical diagram of indirect rotor flux orientation. The instantaneous 3-phase currents are transformed to the rotor flux reference frame, using rotor speed and slip frequency—which means that the motor currents are now observed from the viewpoint of rotating with the rotor flux. As a result of this transformation the currents, now in what is called the d/q reference frame, lose their sinusoidal nature and look like DC signals. In the d/q reference frame, q-axis current controls torque and d-axis current controls flux. If properly oriented, the torque and flux remain independent of each other, and the motor can achieve high efficiency and dynamic response.

Fig. A-1 *Diagram of Indirect Field Orientation (IFO) technique.*

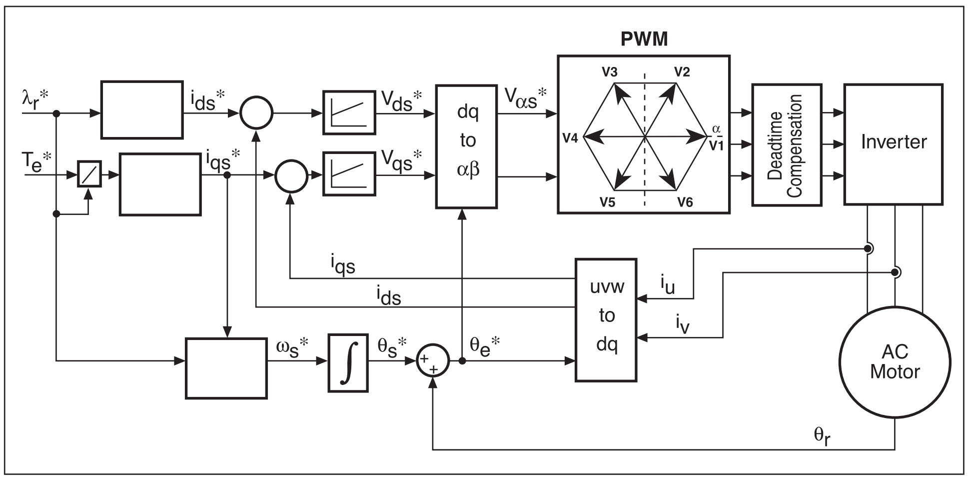
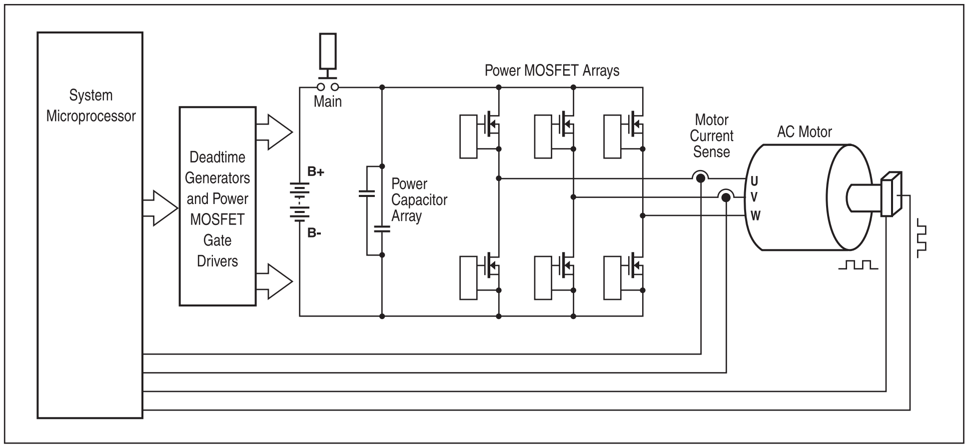


Fig. A-2 *Power section topology.*



Heavy copper busbars connect the IMS modules to the brass external motor connection studs. A bank of power capacitors keeps DC bus levels stable during high frequency MOSFET switching and also reduces EMI on the external B+ and B- cables.

Motor currents and motor speed and direction are the primary feedback signals used in the motor control algorithms. Accurate Hall sensors detect the motor currents; they do this by sensing the flux created by the motor currents on the U and V motor output busbars where they pass through the capacitor board on their way to the external motor connections. Motor speed and direction are simultaneously sensed by a quadrature-type speed encoder mounted on the motor shaft.

APPENDIX B

VEHICLE DESIGN CONSIDERATIONS
REGARDING ELECTROMAGNETIC COMPATIBILITY (EMC)
AND ELECTROSTATIC DISCHARGE (ESD)

ELECTROMAGNETIC COMPATIBILITY (EMC)

Electromagnetic compatibility (EMC) encompasses two areas: emissions and immunity. *Emissions* are radio frequency (RF) energy generated by a product. This energy has the potential to interfere with communications systems such as radio, television, cellular phones, dispatching, aircraft, etc. *Immunity* is the ability of a product to operate normally in the presence of RF energy.

EMC is ultimately a system design issue. Part of the EMC performance is designed into or inherent in each component; another part is designed into or inherent in end product characteristics such as shielding, wiring, and layout; and, finally, a portion is a function of the interactions between all these parts. The design techniques presented below can enhance EMC performance in products that use Curtis motor controllers.

Emissions

Signals with high frequency content can produce significant emissions if connected to a large enough radiating area (created by long wires spaced far apart). Contactor drivers and the motor drive output from Curtis controllers can contribute to RF emissions. Both types of output are pulse width modulated square waves with fast rise and fall times that are rich in harmonics. (Note: contactor drivers that are not modulated will not contribute to emissions.) The impact of these switching waveforms can be minimized by making the wires from the controller to the contactor or motor as short as possible and by placing the wires near each other (bundle contactor wires with Coil Return; bundle motor wires separately).

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

Immunity

Immunity to radiated electric fields can be improved either by reducing overall circuit sensitivity or by keeping undesired signals away from this circuitry. The controller circuitry itself cannot be made less sensitive, since it must accurately detect and process low level signals from sensors such as the throttle potentiometer. Thus immunity is generally achieved by preventing the external RF energy from coupling into sensitive circuitry. This RF energy can get into the controller circuitry via conducted paths and radiated paths.

Conducted paths are created by the wires connected to the controller. These wires act as antennas and the amount of RF energy coupled into them is generally proportional to their length. The RF voltages and currents induced in each wire are applied to the controller pin to which the wire is connected. Curtis controllers include bypass capacitors on the printed circuit board's throttle wires to reduce the impact of this RF energy on the internal circuitry. In some applications, additional filtering in the form of ferrite beads may also be required on various wires to achieve desired performance levels.

Radiated paths are created when the controller circuitry is immersed in an external field. This coupling can be reduced by placing the controller as far as possible from the noise source or by enclosing the controller in a metal box. Some Curtis controllers are enclosed by a heatsink that also provides shielding around the controller circuitry, while others are partially shielded or unshielded. In some applications, the vehicle designer will need to mount the controller within a shielded box on the end product. The box can be constructed of just about any metal, although steel and aluminum are most commonly used.

Most coated plastics do not provide good shielding because the coatings are not true metals, but rather a mixture of small metal particles in a non-conductive binder. These relatively isolated particles may appear to be good based on a dc resistance measurement but do not provide adequate electron mobility to yield good shielding effectiveness. Electroless plating of plastic will yield a true metal and can thus be effective as an RF shield, but it is usually more expensive than the coatings.

A contiguous metal enclosure without any holes or seams, known as a Faraday cage, provides the best shielding for the given material and frequency. When a hole or holes are added, RF currents flowing on the outside surface of the shield must take a longer path to get around the hole than if the surface was contiguous. As more "bending" is required of these currents, more energy is coupled to the inside surface, and thus the shielding effectiveness is reduced. The reduction in shielding is a function of the longest linear dimension of a hole rather than the area. This concept is often applied where ventilation is necessary, in which case many small holes are preferable to a few larger ones.

Applying this same concept to seams or joints between adjacent pieces or segments of a shielded enclosure, it is important to minimize the open length of these seams. Seam length is the distance between points where good ohmic contact is made. This contact can be provided by solder, welds, or pressure contact. If pressure contact is used, attention must be paid to the corrosion characteristics of the shield material and any corrosion-resistant processes applied to the base material. If the ohmic contact itself is not continuous, the shielding effectiveness can be maximized by making the joints between adjacent pieces overlapping rather than abutted.

The shielding effectiveness of an enclosure is further reduced when a wire passes through a hole in the enclosure; RF energy on the wire from an external field is re-radiated into the interior of the enclosure. This coupling mechanism can be reduced by filtering the wire where it passes through the shield boundary.

Given the safety considerations involved in connecting electrical components to the chassis or frame in battery powered vehicles, such filtering will usually consist of a series inductor (or ferrite bead) rather than a shunt capacitor. If a capacitor is used, it must have a voltage rating and leakage characteristics that will allow the end product to meet applicable safety regulations.

The B+ (and B-, if applicable) wires that supply power to control panels—such as an electronic throttle, or control wires such as keyswitch, direction, etc.—should be bundled with the other control wires to the panel so that all these wires are routed together. If the wires to the control panel are routed separately, a larger loop area is formed. Larger loop areas produce more efficient antennas which will result in decreased immunity performance.

Keep all low power I/O separate from the motor and battery leads. When this is not possible, cross them at right angles.

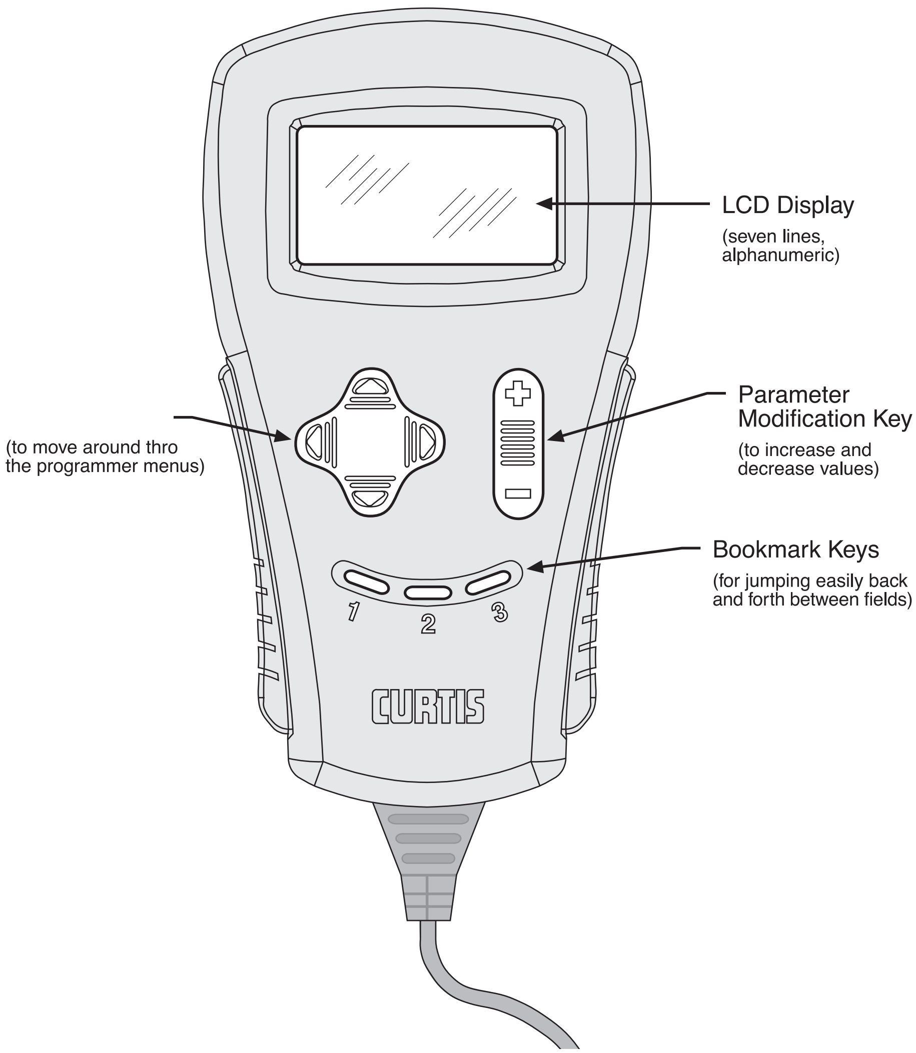
ELECTROSTATIC DISCHARGE (ESD)

Curtis PMC motor controllers contain ESD-sensitive components, and it is therefore necessary to protect them from ESD (electrostatic discharge) damage. Most of these control lines have protection for moderate ESD events, but must be protected from damage if higher levels exist in a particular application.

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

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

MOV, transorbs, or other devices can be placed between B- and offending wires, plates, and touch points if ESD shock cannot be otherwise avoided.


LCD Display
(seven lines, alphanumeric)
(to move around thro the programmer menus)
Parameter Modification Key
(to increase and decrease values)
Bookmark Keys
(for jumping easily back and forth between fields)
1
2
3
CURTIS

PROGRAMMER OPERATION

The 1311 programmer is easy to use, with self-explanatory functions. After plugging in the programmer, wait a few seconds for it to boot up and gather information from the controller.

For experimenting with settings, the programmer can be left plugged in while the vehicle is driven.

The bookmark keys can make parameter adjustment more convenient. For example, in setting the drive forward throttle parameters, you might set a bookmark at the first of these parameters [Program » Throttle » Forward Offset] and another at the raw throttle readout [Monitor » Inputs » Throttle Pot]; this way you can easily toggle between the readout and the parameters.

PROGRAMMER MENUS

There are six main menus, which in turn lead to nested submenus:

<u>Program</u> — provides access to the individual programmable parameters (see Section 3).

<u>Monitor</u> — presents real-time values during vehicle operation; these include all inputs and outputs, as well as the mapped throttle values and conditioned throttle requests (see Section 4a).

<u>Faults</u> — presents diagnostic information, and also a means to clear the fault history file (see Section 7).

<u>Functions</u> — provides access to the controller-cloning commands (see page 55) and to the “reset” command.

<u>Information</u> — displays data about the host controller: model and serial numbers, date of manufacture, hardware and software revisions, and itemization of other devices that may be associated with the controller’s operation.

<u>Programmer Setup</u> — displays data about the programmer: model and serial numbers, date of manufacture, and a list of the programmable parameters that can be accessed with this particular programmer.

APPENDIX D

SPECIFICATIONS

Table D-1 SPECIFICATIONS: 1236/38 CONTROLLER

Nominal input voltage	24–36V, 36–48V, 48–80V
PWM operating frequency	10 kHz
Maximum encoder frequency	15 kHz
Maximum controller output frequency	300 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 temp. range	-40°C to 50°C (-40°F to 122°F)
Internal heatsink operating temp. range	-40°C to 95°C (-40°F to 203°F)
Heatsink overtemperature cutoff	linear cutback starts at 85°C (185°F); complete cutoff at 95°C (203°F)
Heatsink undertemperature cutoff	complete cutoff at -40°C (-40°F)
Package environmental rating	IP65
Weight	*1236:* 4.08 kg (9.0 lbs) ; *1238:* 6.40 kg (14.1 lbs)
Dimensions (W×L×H)	*1236:* 165 × 232 × 102 mm (6.5" × 9.1" × 4.1") *1238:* 275 × 232 × 106 mm (10.8" × 9.1" × 4.2")
Regulatory compliance	EMC emissions: EN50081-2/08.93 EMC immunity: EN50082-2: 1995 Safety, uncontrolled runaway: EN1175 UL Recognized Component Meets UL583 dielectric test.

MODEL NUMBER	NOMINAL BATTERY VOLTAGE (volts)	CURRENT LIMIT (amps)	2 MIN RATING * (amps)	1 HOUR RATING * (amps)
1236-44XX	24–36	400	400	155
-45XX	24–36	500	500	180
-53XX	36–48	350	350	140
-63XX	48–80	300	300	100
1238-46XX	24–36	650	650	265
-54XX	36–48	450	450	210
-56XX	36–48	650	650	210
-65XX	48–80	550	550	155

Notes: All current ratings are rms values per motor phase. Internal algorithms automatically reduce maximum current limit when heatsink temperature is >85°C or battery voltage is outside the allowed limits. Heatsink temperature is measured internally near the power MOSFETs.

2-minute ratings are based on an initial controller heatsink temperature of 25°C and a maximum heatsink temperature of 85°C. No addtitional external heatsink is used for the 2-minute rating test.

1-hour ratings are based on an ambient temperature of 25°C with the controller mounted to a heatsink with a thermal resistance of 0.35°C/W for the 1236, or 0.25°C/W for the 1238, operating at a maximum baseplate temperature of 85°C. These thermal resistances are approximately equivalent to a 0.5m × 0.5m × 8mm thick vertical steel plate in free air with 6kph airflow on one side. Customer installations with less heatsinking may have lower 1-hour ratings. The 1-hour rating can be increased with a properly mounted controller that can reject the long-term heat generation.