HcT
Electro-Hydraulic
Control Solutions

Quick Start Guide

DVC Family

The information in this publication is intended as a guide only, and HCT take NO responsibility for usage and implementation in any user written application code structure.
HCT strongly suggests that the user attends one of the product training courses to ensure correct and full understanding of this information and to learn further optimized methods of control techniques.
Please contact HCT customer service to book one of the scheduled training dates or to discuss arranging a course specific to your company needs.

Thank you for using High Country Tek Inc. Products.

Manual Index:

1 Software Installation

1.1 System Requirements

Windows XP Professional
Windows Vista Business or Ultimate
40 megabytes of disk space to support a complete system install
PC with Serial Port - RS232 or USB port
For USB ports you need a USB to RS232 converter (i.e. Dongle)
DVC5/7/10 controller module
DVC10 serial cable

1.2 Installation

When you install the DVC development software following the steps outlined here you will have one Programming Tool and one Program Loader Monitor that supports the DVC5, DVC7 and the DVC10.

To install the DVC development software, close all program applications. Insert the DVC Software CD in the CD-ROM drive, and wait for the installer program to execute. The installer will guide you through the installation procedure.

1.3 Software Overview

The DVC software package includes two applications named "Programming Tool" and "Program Loader Monitor". These applications provide the means to configure, design, create, load, and monitor the user application. The Programming tool is used to configure the inputs and outputs and program the control logic for an application. The Program Loader Monitor is used to download the user application program files to the DVC5/7/10. It also performs real time monitoring of your system inputs/outputs. Also, you can modify some input/output configuration settings (i.e. like analog ramp rates). Another feature of the Program Loader Monitor is it allows you to see and change EEmem variable settings. For example, this is a convenient way to have the user interact with the system to change a maximum pressure setting or input a customer job number as the system is running.

2 Hardware Wiring Installation

Design your system by first determining the inputs and outputs required. Refer to our hardware wiring layouts located in both the HCT Programmer's User's guide section 10 and the website, www.highcountrytek.com.

3 DVC Expansion Modules

The DVC5, DVC7 and DVC10 are programmable controllers. Each has a fixed number of Inputs and Outputs for standalone operation. If your system requires more inputs or outputs than the DVC5/7/10 provide you have two expansion options. Additional DVC5, DVC7 and DVC10s can be configured each as master controllers for a portion of your application. Each DVC5, DVC7 and DVC10 will have their own application program and can communicate with each other using an application specific CAN Bus message we will describe. In addition the DVC10 can talk to other DVC expansion modules over the CAN Bus to expand your specific input and output capabilities. It is important to note that if the system requires multiple modules of the same model, specific module names and Mac ID# must be unique. For instance, if the system uses (2) DVC50 modules, one could be named dvc50_1 with a Mac ID of 50. The other could be named dvc50_2 with a Mac ID of 51.

DVC10 Expansion Modules:

DVC21	40 Sinking and Sourcing Digital Inputs
DVC22	40 Sinking Digital Inputs
DVC41	12 High Side Outputs for Bang-Bang Valves or LEDs
DVC50	3 Output Groups, 8 Digital, 3 Analog and 3 Universal I/Os similar to the DVC10
DVC61	4x20 character screen display with 12 display variables and 5 single pole double throw digital inputs
DVC62	RS232 connected handheld 4X20 character display with a 20 button keypad for data entry.
DVC65	2x20 Character Screen Output
DVC70	Memory module for Data Logging and Tracing
DVC80	J1939 CAN Bus Support
DVC5/7/10 to 5/7/10	The ability multiple DVC5, DVC7 and DVC10s to talk to each other
D206	RS232 connected Touch Screen Color Graphical Display

Refer to our expansion module section located in both the HCT Programmer's User's guide and the website, www.highcountrytek.com.

4 Basic List of Actions to Create your First Application

The Intella programming environment allows the programmer to be creative in their programming style. Here are some suggestions as how a program may be structured.

4.1 Do's and Don'ts

Always Do the Following

- Fully read any supplied information / data sheets **BEFORE** starting and visit our website at http://www.highcountrytek.com
- Make sure the driver supply voltage is sufficient for the coils on the valve being driven! (24V supply is OK for both 24V & 12V coils while 12V supply is only good for 12V coils).
  - **NOTE:** Coil resistance changes with heat, and even modules with 'Current Feedback' for auto-corrections, may NOT have sufficient supply voltage to maintain maximum or correct valve operation.
  -
- Check that the power supply voltage is CORRECT, 'ELECTRICALLY CLEAN' AND STABLE.
- Totally 'Isolate' the modules from all other equipment and disconnect all power supplies BEFORE any form or welding or battery charging takes place or damage may occur.
- Unless specified, ensure that each valve coil has 'dedicated' wires to the driver connector for correct operation.
- Use 'Screened / Shielded' cable where recommended in the application data sheets.
- Check ALL cables to and from this unit to ensure correct, firm connections with NO loose wires and NO short or open circuits.
- Ensure that any unused wires / terminals are isolated safely and not shorted together.
- Fit correctly rated 'In-Line' fuses on all Power supply lines to protect the system and controller unit as recommended in the HCT detailed literature.
- Operate the modules WITHIN the specified operating temperature range for best, longest and most reliable performance.
- Contact HCT for additional information if you are still unsure of connecting and usage of the driver electronics chosen for your application.
- Keep notes on final parameters, connections and modifications to allow drawings to be updated and replacement parts to be quickly adjusted / set and installed if ever needed.

Never Do the Following

- Attempt to use a module if you are unsure of I/O connections or expected application operation.
- Attempt to use a module in areas or systems where AC or DC inductors HAVE NOT been fully suppressed (i.e. relay coils, motor starters etc.)
- Use a controller in areas of 'Radio Frequency' (RF) without adequate screening / shielding.
- Attempt to use a power supply that is not rated for the correct required output current and voltage under full load conditions as damage may occur to the driver card / module.
- Allow wires TO or FROM the controller to be short circuited to each other or shorted to chassis / cabinet etc
- Disconnect or connect wires to a module unless it is fully isolated from the power supply (i.e. turned OFF).
- Operate electronics in environments not specified in the modules literature (i.e. high humidity, Intrinsically safe or hazardous, explosive gasses etc)
- Allow unauthorized or unqualified personnel to connect, set-up, use, modify or repair any driver or controller.

Wherever Possible Do the Following

- Keep High Voltage AC cables **separate** from Low Voltage DC signal and power supply cables.
- Use the largest practical wire size (gage) with the biggest cross sectional area (mm^2) possible to minimize losses over cable length.
- Use a 'Switch-Mode Power supply, their wide input voltage range and built in filters on input and outputs make them ideal for today's systems.
- Mount the modules separately to motor starters, contactors and relay switches to minimize electrical noise pickup and other noise (RF) related problems.

4.2 Create machine function flowchart (a.k.a. Sequence of Operations)

This idea forces the development team to understand what the function of the machine will be before the program is written. Using flowchart notation, the development team documents each function of the machine. In the case of a hydraulic log splitter, maybe the first function would be to assure the engine is running, providing pressure for the pump. It is advised that the first flowchart of a project contain very simple steps in the functionality. Later revisions can combine simple functions into more complex functions. Insert as much information in this flowchart as possible. For instance, if there is a pressure relief valve in the machine, set at 1500psi, list that in the flowchart. Once the flowchart is finished, everyone should agree on the machine functionality. This step will help to minimize changes to the program at a later date. This step could take considerable time, but will make the program much simpler to write. Refer to chapter 5 for an example.

4.3 Separate different functions into different logic bubbles.

It is not incorrect to place all of the program logic in the always code, but it can slow down the program and make it more difficult to troubleshoot. Time critical logic should be contained in the 'always' code. Different functions should be contained in individual bubbles. For example, a pump control's logic can be contained in a bubble by itself. The logic to control the track drive of a bulldozer can be contained in a bubble by itself. These functions may not be as critical as something such as looking for a high pressure situation.

4.4 Use meaningful variable names.

The limit on variable size is 32 characters. A variable that isn't an input or output device is called an internal variable. An internal variable that is for Valve1 minimum current setting for the program loader monitor could be labeled Valve1MinCurA_PLM_IV. This naming convention could be used throughout the program. An EEmem variable for a valve's minimum current setting could be named Valve1MinCurB_eem. Below are some other examples.

Valve1ctl_AI2 – analog input#2. Variable names in capitol letters designate an abbreviation.
Cross_valve_UI2 – universal input #2.
Output_shaft_RPM_PI1 – rpm pulse input#1.
Pump#2_milling_pwm_OG1 – output group #1, pwm controlled motor.
OK_to_begin_led_OG3 – output group #3, discrete(bang bang) output to control LED.
Engine_rpm_lo_J1939 – The low byte of the engine rpm from the j1939 network.
Engine_temp_J1939 – The engine temperature from the j1939 network.
Network_status_eem - EEmem variable.
Mast_into_position1_IV – internal variable.
EEmem_spare_128 – eemem variable that doesn't have a specific name because unused. Makes programmer aware of spare EEmem addresses.

4.5 Declare all variables in one location

Have one area of one bubble that is executed once to declare all variable names. This area would include unsigned integers (Uint), EEmemory variables (EEmem), Timers (timer), constants (const) and others. Declare all 128 EEmem variables. Variables that aren't used could be name EEmem_spare_89, or such. An exception would be a variable that is declared private; this would have to be declared in the bubble used. Declaring all the variables in one location could make it easier to add another variable at a later time, and provides for a cleaner, more structured program.

4.6 Comment important information into the program

Information such as Programmer's name, date of program creation, revision history, and any description of something that is difficult to understand is appreciated by anyone offering assistance in troubleshooting the program. Explaining a complex math equation can be beneficial at a later date to refresh the programmer's memory of why a function was built the way it was and to assist in troubleshooting. Comment as much or little as necessary. Comments do not contribute to the compiled program (.pgm) file size. Remember to add a colon (‘) before starting comments. The Intella software interprets the (‘) that the information to the right is not to be compiled.

```
‘Programmer’s Name: Paul Programmer
‘Program creation date: 5/28/2007
‘Revision history: 5/28/07 – initial release
‘Program description: This program will…
‘        The math function for the tank level takes into consideration the ….
```

The "Programming Tool" is used to create your application for the DVC5/7/10. The "Program Loader Monitor" is used to load the user application into the DVC5/7/10 module and monitor the inputs and outputs in real-time mode as your application executes. Both programs are located in the Windows Start Menu under the c:\Program Files\HCT Products. In order to create a user application, the following steps generally should be followed:

a) Architect your system. In the design process compile a total list of digital inputs, pulse inputs, PWM outputs etc. A total I/O count is needed to assure the correct amount of DVC inputs / outputs are specified in external modules, if needed. Also, remember to allow room for expansion with spare I/O.
b) Define your system components and their interactions. HCT does have experience with a wide variety of sensor manufactures. Request a sensor recommendation sheet of sensors known to work with the HCT system.
c) Start the Intella Programming Tool. Create a new project by saving the file, i.e. hydraulic log splitter. Select your Project type to be a DVC5, DVC5/J1939, DVC7 or DVC10 project.
d) Configure the DVC5/7/10 and DVC expansion module's inputs and outputs that drive or sense your components. With the mouse on the screen, right click to choose the expansion module. Insert the expansion module, set the modules unique name and Mac ID#. Assign names to the real world I/O, refer to section 1.16 for further guidance on I/O names.
e) Create a flow diagram of machine function (aka Sequence of Operations) refer to chapter 4.1 and chapter 5 for additional information. Once the Sequence of Operations (SoO) is complete, this needs to be converted into a program. The ‘always’ code as well as bubble logic sequences will be used during this step. Refer to ‘DVC Family System and Programming User Guide’ section 5 for programming examples. Refer to Appendix A for compiler keywords. Program the control logic as to how the outputs respond to the inputs.
f) Compile the program. Refer ‘DVC Family System and Programming User Guide’ to section 3.1.
g) Load the compiled program files into the DVC5/7/10. Refer to ‘DVC Family System and Programming User Guide’ section 3.5.
h) Run your system and debug your application.

Graphical windows allow you to specify the system components you will need and the electrical characteristics of your inputs and outputs. Text screens are provided to enter the program control logic (using a subset of the Basic language) for both your time critical and normal operations.

5 Flowchart (Sequence of Operations) Example

This example will demonstrate how to create a flowchart and Sequence of Operations. This example may or may not work; it is for educational purposes only.

5.1 Define functionality

Programming Team:

Paul – Programmer, Electrical Engineer
Mark – Program Manager
Amanda – Application Engineer
Todd – Hydraulics Engineer

Date: 6/4/08

Program Description:

Using the HCT DVC family, control steering and propulsion functions of a new skidsteer model
Using a j1939 equipped diesel engine.

Functional Description:

The operator has control over one joystick. Critical functions are the engine rpm > 1000rpm and high side pressure is below 2000psi, then the system looks for input from the joystick as indicated below.
Joystick has 5 positions
Joystick forward – skidsteer moves forward motion
Joystick back – skidsteer moves in reverse motion
Joystick right – skidsteer turns right
Joystick left – skidsteer turns left
Joystick center – skidsteer stops in present position

I/O needed

Inputs	Joystick forward	digital input 1, active high
	Joystick back	digital input 2, active high
	Joystick right	digital input 3, active high
	Joystick left	digital input 4, active high
	Joystick center	digital input 5, active high
	Left wheels, high side press. Trans	0-3000psi transducer, 0-5v, 0-100% input
	Right wheels, high side press. Trans	0-3000psi transducer, 0-5v, 0-100% input
	Engine load	j1939 signal
	Engine rpm	j1939 signal

Outputs

Left wheels pump forward control pwm	Min cur 0.2a, Max cur 0.9a
Left wheels pump reverse control pwm	Min cur 0.2a, Max cur 0.9a
Right wheels pump forward control pwm	Min cur 0.2a, Max cur 0.9a
Right wheels pump reverse control pwm	Min cur 0.2a, Max cur 0.9a
Turning left indicator, bang bang out	LED indicator along with standard relay
Turning right indicator, bang bang out	LED indicator along with standard relay

5.2 Define functionality steps

Step #1 Assure engine rpm is above 1000rpm
Assure high pressure, left wheel system < 2000psi
Assure high pressure, right wheel system < 2000psi

Step #2 Holding position, wait for operator input for direction.
if joystick is in center position. Stop motion
Then
Left wheels pump, 0% displacement, neither forward/reverse enable
Right wheels pump, 0% displacement, neither forward/reverse enable

Step #2a if joystick is in forward position. Forward
Then
Left wheels pump, ramp 50% displacement, forward enable
Right wheels pump, ramp 50% displacement, forward enable

Step #2b if joystick is in back position. Reverse
Then
Left wheels pump, ramp 50% displacement, reverse enable
Right wheels pump, ramp 50% displacement, reverse enable

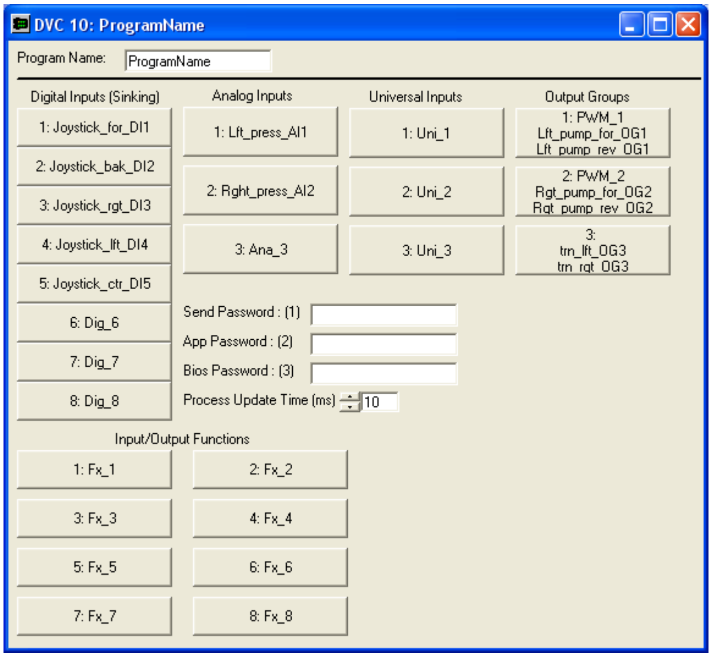
Step #2c if joystick is in right position. Turn right
Then
Left wheels pump, ramp 50% displacement, forward enable
Right wheels pump, ramp 50% displacement, reverse enable

Step #2d if joystick is in left position. Turn left
Then
Left wheels pump, ramp 50% displacement, reverse enable
Right wheels pump, ramp 50% displacement, forward enable

6 I/O List to Hardware Definitions

6.1 Declare I/O

When the flow charts are complete from chapter 5, above, start the Intella software and define the project, then declare all I/O.



DVC 10: ProgramName

Program Name: ProgramName

Digital Inputs (Sinking)	Analog Inputs	Universal Inputs	Output Groups
1: Joystick_for_DI1	1: Lft_press_AI1	1: Uni_1	1: PWM_1 Lft_pump_for_OG1 Lft_pump_rev_OG1
2: Joystick_bak_DI2	2: Rght_press_AI2	2: Uni_2	2: PWM_2 Rgt_pump_for_OG2 Rgt_pump_rev_OG2
3: Joystick_rgt_DI3	3: Ana_3	3: Uni_3	3: trn_lft_OG3 trn_rgt_OG3
4: Joystick_lft_DI4			
5: Joystick_ctr_DI5			
6: Dig_6			
7: Dig_7			
8: Dig_8			

Send Password : (1)

App Password : (2)

Bios Password : (3)

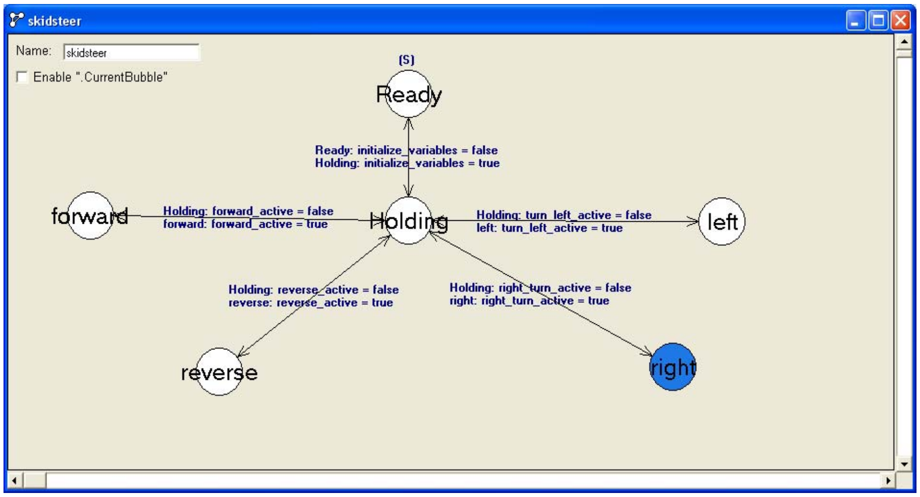
Process Update Time (ms) 10

Input/Output Functions

1: Fx_1	2: Fx_2
3: Fx_3	4: Fx_4
5: Fx_5	6: Fx_6
7: Fx_7	8: Fx_8

6.2 Create bubble framework

Now, create the framework for the different bubbles



skidsteer

Name: skidsteer

□ Enable ".CurrentBubble"

(S)
Ready
Ready: initialize_variables = false
Holding: initialize_variables = true
forward
Holding: forward_active = false
forward: forward_active = true
Holding
Holding: turn_left_active = false
left: turn_left_active = true
left
Holding: reverse_active = false
reverse: reverse_active = true
Holding: right_turn_active = false
right: right_turn_active = true
reverse
right

Notice the bubbles are directly related to the flowchart steps, now the programmer needs to write code for the individual bubbles. This is a very simple example, but it gives a demonstration of how the flowchart can be created and then converted into the bubble logic.

7 EE Memory Variable Use

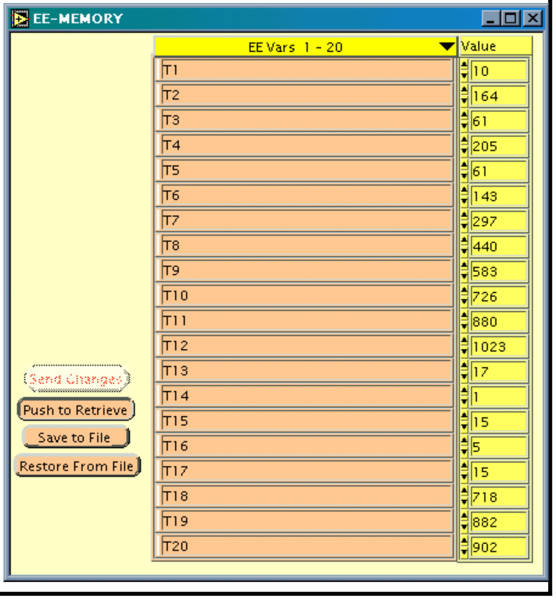
Electronically erasable memory (EE Memory) is memory that is maintained (non volatile) when there is no power to the DVC5/7/10. The DVC5/7/10 has 128 EE memory locations. EEmemory locations can be used to interface to the compiled, running DVC program. For instance, if during troubleshooting, the user wanted to change between different virtual display screens, the programmer may create a EEmem variable 'virtual_screen'. By programming the DVC, the contents of the variable 'virtual_screen' could be monitored to determine which virtual display is active. EE memory locations are all unsigned 16-bit values that can store any number from 0 to 65535. EE memory names can be 32 characters in length. The actual EE memory is not used while the program is running. A mirror copy in DVC5/7/10's program memory is used to prevent over usage of the EE memory. There is a special command to save the mirror copy to the EE memory and another one to copy the EE memory to the program memory location. The EE memory has an approximate 1 million writes guarantee. If a new value where to be stored every minute, the DVC5/7/10 is guaranteed to run for 1.9 Years.
To save all of the EE memory, execute this line of code in your application: EECommand = EEWrite. In between writes you will need to reset the EEcommand to 0. Also, note that the EE memory will only be written to if one or more of the EE memory variables has changed. So if you insert the EEcommand = EEwrite in your always bubble or some other frequently executed logic bubble you are very unlikely to exceed the 1 million writes limit of the memory chips. A typical sequence of code to update EE memory would look like the following:

```
Dim eememory_update_timer as timer
EEcommand = 0
If (eememory_update_timer = 0s) then
   eememory_update_timer = 2s    ‘ timer set to > 10ms needed to insure at least one EEcommand = 0
                                 ‘ executed between EEcommand = EEwrite commands
   EEcommand = EEwrite
End if
```

To declare an EE memory variable use this line of code: Dim VarName as EEmem
To save all of the EE memory use this command: EECommand = EEWrite

To restore actual EEmemory values to program memory use this line of code: EECommand = EERead
EERead (is rarely used but) would be used if you had changed an EE memory variable in your application but had not saved it to permanent memory and wished to reset the variable to the permanent EE memory value.

Note: When the DVC5/7/10 powers up the program memory copy of EE memory is automatically initialized to the values stored in permanent EE memory. Therefore, you do not need to start your program with an EEread command.



EE Vars 1 - 20	Value
T1	10
T2	164
T3	61
T4	205
T5	61
T6	143
T7	297
T8	440
T9	583
T10	726
T11	880
T12	1023
T13	17
T14	1
T15	15
T16	5
T17	15
T18	718
T19	882
T20	902

7.1 EE memory when online (PLM)

Double click on the EE memory Status icon on the main Program Loader Monitor window to activate this screen. EE memory is non-volatile (i.e. it retains the value even when power is turned off). Memory variable names and values are set using the Programming Tool or as the program executes. The selector at the top of the screen will allow you to page through the variables. You can change the variables integer values using the up and down arrows next to the variable's name. The new values can be sent to the DVC5/7/10 EE memory by selecting the Send Changes button. Additionally, the Save to File button allows the user to save the EE variables and their respective values to a .DAT file on the PC. The Restore from File button allows the user to recall previously saved EE variable values from a .DAT file. The Push to Retrieve button initiates the Restore operation after the Restore From File button is selected.

8 HCT Terminology and Definitions

Always bubble – Time critical logic needs to be contained here because this bubble will be executed based on the process update time(default 10ms).
Analog Inputs – The state of the input is measured from 0-5vdc. Under the universal Input configuration the input ranges are (-1 to +1 volt), (0 to 5volts), (0 to 10volts), or (0 to 22mA).
Bang Bang Valve – Discrete function on/off type valve.
Compile – This function (Ctrl-M) will create a .pgm file that will be loaded into the DVC system. Any syntax errors in the program will be flagged in this step.
Counter Mode Input – Input counts high pulses depending input range, counts would be reset through program.
Current Regulation - provides for automatic hardware and BIOS based adjustments to maintain the coil current at the application code's PWM% setting.
Digital Inputs – The state of the input is either a logical 0 (off) or a logical 1 (on).
Dual Coil High-Side Output- Gives programmer access to (2) PWM high side outputs.
DVC – Digital Valve Controller.
EEMemory - Electronically erasable memory (EE Memory) is memory that is maintained (non volatile) when there is no power to the DVC5/7/10. EEmemory locations can be used to interface to the compiled, running DVC program.
Enable Current Ramps - The output current will be ramped up or down based on the ramp times to the Low-Side Name setpoint.
Enable Process PI - provides a facility where the application program sets a desired setpoint value and then continually determines a feedback variable
Input/Output Funcions – Patented feature of HCT, allows programmer to simplify math functions by custom configuring an output based on a programmable input. Could be used to give a non-linear output of a pump.
High-Side Only Output – Gives programmer access to (2) bang bang high side outputs.
Output Group – The outputs of the DVC module can be configured into 4 groups, Dual Coil High Side, Single Coil High-Side, Single Coil Low-Side, High-Side Only.
Password (App) – Level 2 Password to limit access to certain screens in the Loader Monitor.
Password (Bios) – Level 3 Password to limit access to certain screens in the Loader Monitor.
Password (Send) – Level 1 Password to limit access to certain screens in the Loader Monitor.
Process Update Time (ms) – Sets the process time from 1 to 20ms. Default is 10ms.
Program Loader Monitor(PLM) – software used to download programs into DVC modules. Also used to monitor program 'online'.
Programming Tool – Intella software used to author and modify DVC code.
PWM Duty Cycle Control - The Low-Side Name allows direct PWM control.
PWM Frequency – Controls the frequency output.
PWM (Pulse Width Modulation) – Proportional control 0-100%.
RPM Pulse Input – Input will count pulses, set pulses time out and pulses per rev. Could be used to count teeth on a gear to determine speed of a driveshaft.
Single Coil Low-Side Output- Gives programmer access to (1) PWM low side output and (2) bang bang high side valves.
Single Coil High-Side Output- Gives programmer access to (1) PWM high side output and (1) bang bang high side valves.
System Heartbeat – Controller emits a pulse at a regular interval. Sometimes used to acknowledge a module is still online and communicating.
Universal Inputs – These inputs can be configured for three types of inputs, Analog Input, RPM Pulse Input, counter mode or digital input.

9 Safety is Everyone’s Responsibility

Safe work practices need to be observed in building the hardware connections, mounting the units to the machinery, and programming the controllers.

9.1 Safety in building the hardware connections

Safety should be at the forefront of the development team’s thoughts. Many times during development, technicians and engineers will fabricate test fixtures, care must be taken not to short circuit power supplies and output devices. Please adhere to all federal, state, and local laws regarding safety.

9.2 Safety in mounting the DVC units

Mounting the DVC units on mobile applications have their own safety guidelines. Care must be taken to locate a safe place on the machine free from excessive heat, and moving parts that may sever the controller’s wiring harness or physically harm the controllers. Please adhere to all federal, state, and local laws regarding safety.

9.3 Safety in programming the controllers

Safety to personnel and machinery must be observed when programming moveable functions. Test program changes to minimize safety hazards. If possible don’t test a machine at full function, only test program modification area in a controlled environment. Make sure to let other personnel in the area know that a change is being tested and possible negative outcomes. Please adhere to all federal, state, and local laws regarding safety.

High Country Tek Inc.
208 Gold Flat Court
Nevada City, CA, 95959.

Customer Service
Phone: 1 530 265 3236

www.highcountrytek.com