DRIVVEN

Stand Alone Direct Injector Driver System
User's Manual

D000103 3 Channel
D000100 6 Channel
D000101 9 Channel
D000102 12 Channel

October 2010

Drivven, Inc. • 12001 Network Blvd • Ste 110 • San Antonio, Texas 78249 • USA
Phone : 210.248.9308
Web : www.drivven.com , E-mail : info@drivven.com

HIGH VOLTAGE: This device normally operates at voltages up to 150 volts. Extreme care should be taken to protect against shock. Even when the device is completely powered down, allow approximately three minutes for the internal high voltage to dissipate. Do not touch any of the module screw terminals or injector terminals while the device is powered.

Table of Contents

Table of Figures and Tables

Introduction

The Drivven CompactRIO (cRIO) DI Driver Module Kits are intended to be used within a full authority engine control system where the cRIO controller or PXI controller are performing the engine control algorithms. However, some customers have requested a stand-alone driver system which allows them to use another external controller to generate digital fuel control signals to interface to our DI Driver Modules. Drivven now offers a family of Stand-Alone Direct Injector (SADI) Driver Systems that enable this configuration.

Drivven offers four different base SADI Driver Systems which provide three, six, nine, and twelve driver channels for solenoid injectors. When used with piezo injectors, each system provides two, four, six, and eight driver channels, respectively. These base systems provide a digital input per channel for directly commanding each injector. Each base system includes the following hardware:

3 Channel SADI Driver System (SADI3)

- 1 cRIO-9012 RT Controller
- 1 cRIO-9111 4-Slot FPGA Chassis
- 1 Drivven DI Driver Module
- 1 cRIO-9411 Digital Input Module

6 Channel SADI Driver System (SADI6)

- 1 cRIO-9012 RT Controller
- 1 cRIO-9111 4-Slot FPGA Chassis
- 2 Drivven DI Driver Modules
- 1 cRIO-9411 Digital Input Module

9 Channel SADI Driver System (SADI9)

- 1 cRIO-9012 RT Controller
- 1 cRIO-9112 8-Slot FPGA Chassis
- 3 Drivven DI Driver Modules
- 2 cRIO-9411 Digital Input Modules

12 Channel SADI Driver System (SADI12)

- 1 cRIO-9012 RT Controller
- 1 cRIO-9112 8-Slot FPGA Chassis
- 4 Drivven DI Driver Modules
- 2 cRIO-9411 Digital Input Modules

Advanced Software Package

Drivven also offers an Advanced Software Package with each system which includes the following additional features:

Engine Position Tracking (EPT)- This feature takes cam and crank signals as inputs and tracks the crankshaft angle for two-stroke or four-stroke engine cycles. The crank and cam signals can be from an optical encoder and camshaft sensor, or from production crank and cam position sensors (VR sensors require additional hardware). The EPT feature allows the user to enter the start of injection angles and durations for injection events. Up to five pulses per cycle are possible.

Rail Pressure Control- With the addition of a Drivven Low-Side Driver and an NI-9215 common rail fuel pressure can be controlled. The software uses the NI-9215 to measure the rail pressure while using the Low-Side Driver to PWM the Inlet Metering Valve (IMV) and the High Pressure Valve (HPV) on the rail with a closed loop PID algorithm. The user may tune the PID gains to suit the requirements of their system.

Triggered Multi Pulse (TMP)- This features uses the rising or falling edge of the external digital command as a trigger for a multi-pulse injection sequence for up to five pulses per trigger. The user may define the delay and duration for each of five injection pulses. This is useful for adding multi-pulse injection capabilities to a system that only allows one output trigger per cycle.

Injector Calibration- This feature allows the user to command a programmable number of injections at a fixed duration and interval. This is helpful for measuring the mass of fuel injected per injection event.

In order to use the rail pressure control of the Advanced Add-on Package, a Low-Side Driver and NI-9215 must be purchased in addition to the advanced add-on software package. The different channel-count SADI Driver Systems with rail pressure control capabilities include the following:

3 Channel SADI Driver System (SADI3)

- 1 cRIO-9012 RT Controller
- 1 cRIO-9111 4-Slot FPGA Chassis
- 1 Drivven DI Driver Module
- 1 cRIO-9411 Digital Input Module
- 1 Low-Side Driver Module
- 1 NI-9215

6 Channel SADI Driver System (SADI6)

- 1 cRIO-9012 RT Controller
- 1 cRIO-9112 8-Slot FPGA Chassis
- 2 Drivven DI Driver Modules
- 1 cRIO-9411 Digital Input Module
- 1 Low-Side Driver Module
- 1 NI-9215

9 Channel SADI Driver System (SADI9)

- 1 cRIO-9012 RT Controller
- 1 cRIO-9112 8-Slot FPGA Chassis
- 3 Drivven DI Driver Modules
- 2 cRIO-9411 Digital Input Modules
- 1 Low-Side Driver Module
- 1 NI-9215

12 Channel SADI Driver System (SADI12)

- 1 cRIO-9012 RT Controller
- 1 cRIO-9112 8-Slot FPGA Chassis
- 4 Drivven DI Driver Modules
- 2 cRIO-9411 Digital Input Modules
- 1 Low-Side Driver Module
- 1 NI-9215

Each SADI Driver System is delivered with an installed startup-executable application which begins executing about 20 seconds after the cRIO controller is powered up. Drivven's CalVIEW is used as a calibration user interface to the system. The calibration settings are stored on the RT controller and are loaded immediately upon system boot. The user interface is accessed by running the CalVIEW console software on a Windows PC, communicating with the RT controller over Ethernet. LabVIEW development tools are NOT required, but the free LabVIEW Run-time Engine is required to be installed on the Windows PC to support CalVIEW. Each SADI Driver System is provided with a CalVIEW Console license. The CalVIEW user interface is not required for normal driver operation. The system will run according to the previously saved calibration. No other products from National Instruments or Drivven are required for this system to operate.

If the user sends Drivven an injector ahead of time, Drivven will calibrate the current profile so that the user can run right out of the box, for no additional charge. It should also be noted that similar systems can also be custom configured for port fuel injectors, using Drivven PFI Driver modules.

CalVIEW and the Direct Injector Driver modules have separate user manuals for their individual use. The information in this SADI Driver System manual may overlap in some areas, but please refer to the other product's documentation for product-specific information. The hardware system configurations available are shown below in Figure 1 and Figure 2.

System Configurations

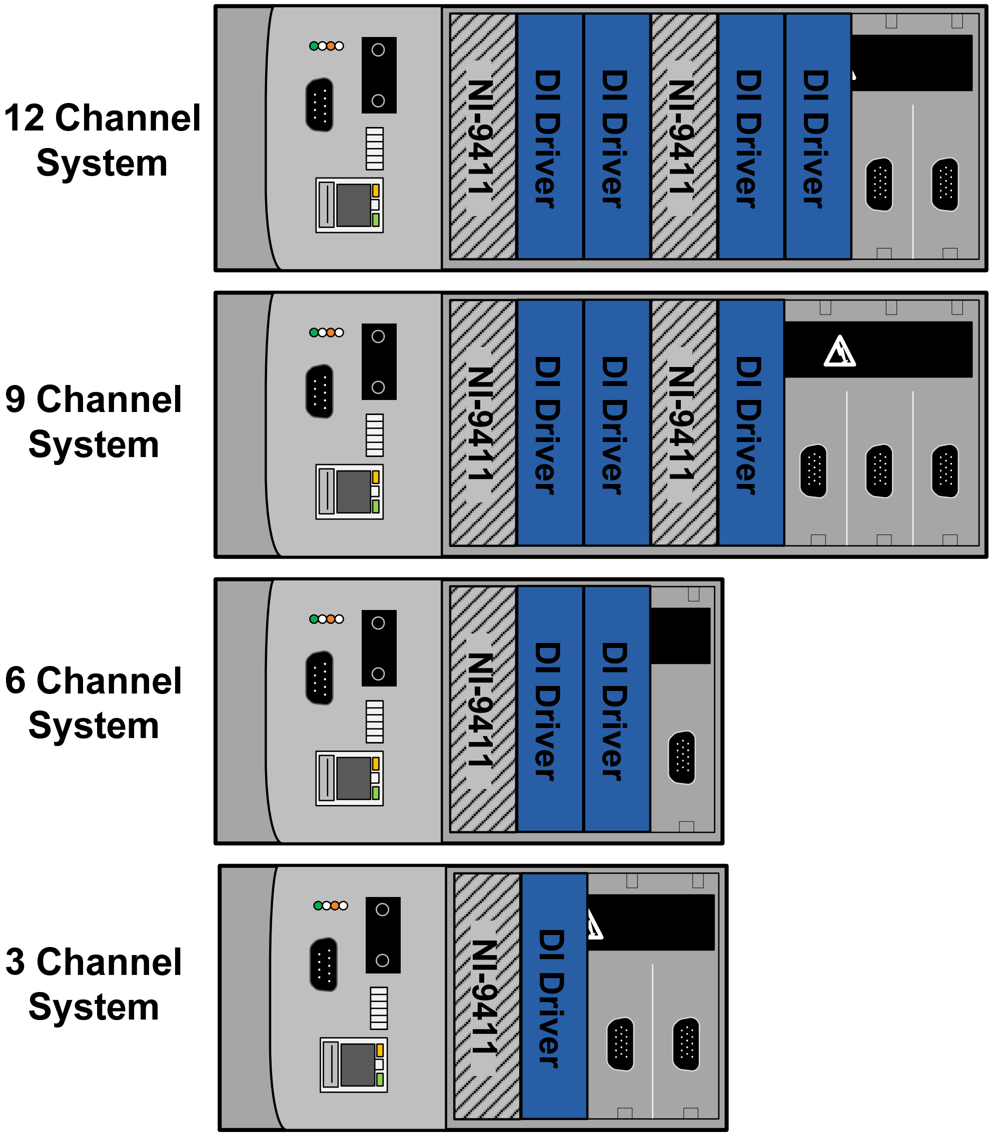


Figure 1. Base SADI-Driver System Configurations

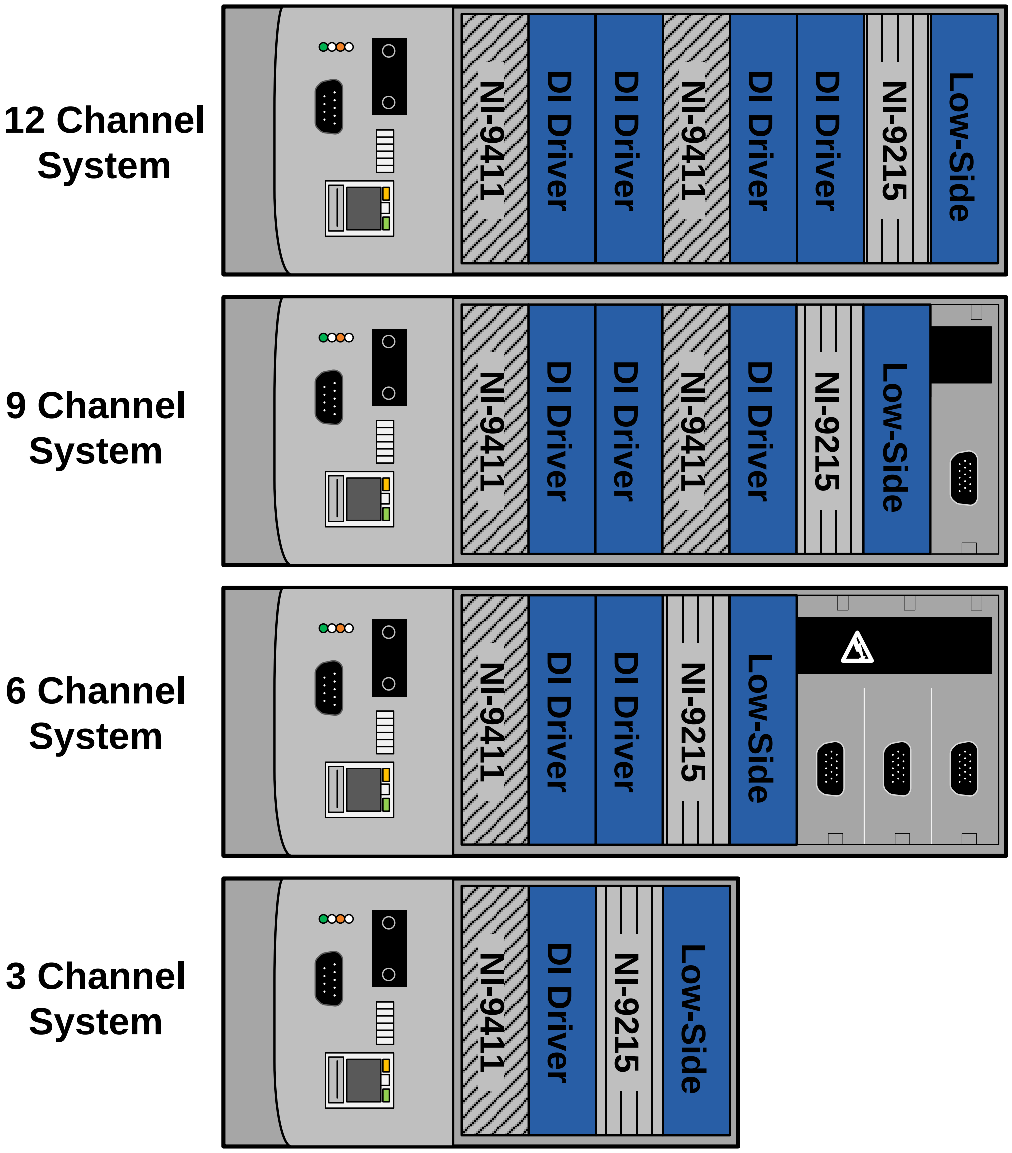
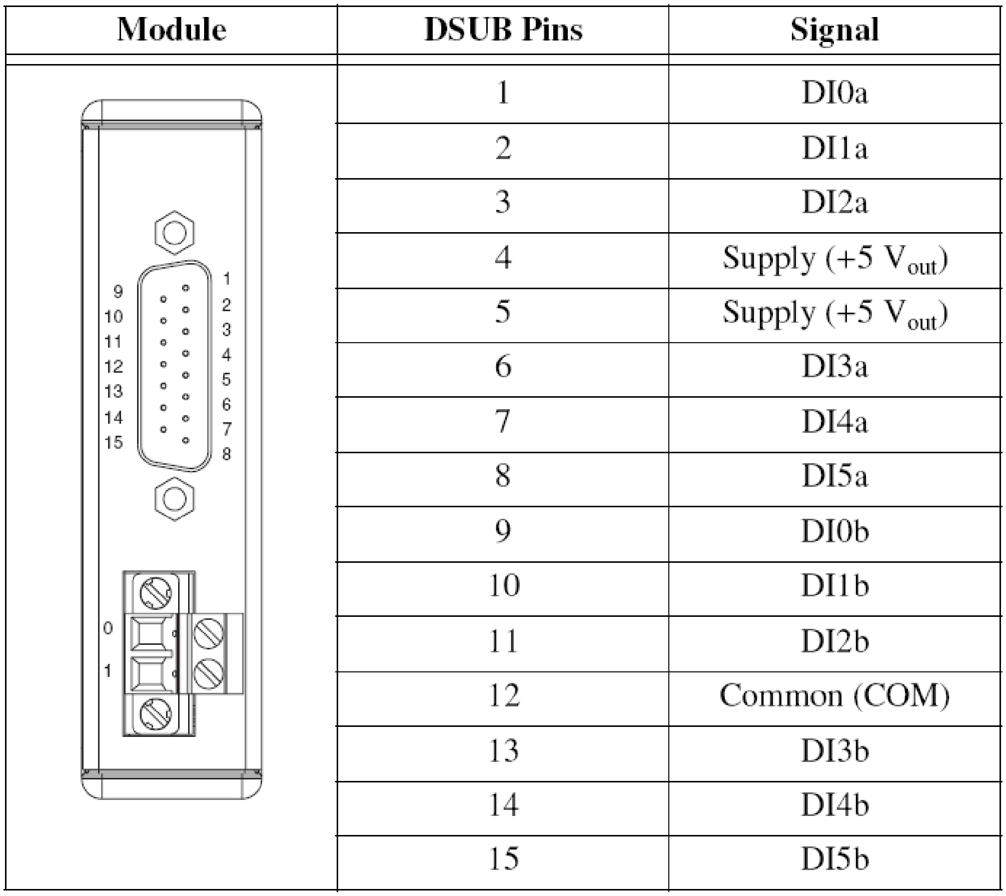


Figure 2. Advanced SADI-Driver System Configurations with Rail Pressure Control

NI-9411 Digital Input Module Pinout

The pinout for the connectors of the NI-9411 module is shown below in Tables 1 and 2.

Table 1. NI-9411 DB-15 Connector Pinout



9
10
11
12
13
14
15
1
2
3
4
5
6
7
8
0
1

Module	DSUB Pins	Signal
	1	DI0a
	2	DI1a
	3	DI2a
	4	Supply (+5 V_{out})
	5	Supply (+5 V_{out})
	6	DI3a
	7	DI4a
	8	DI5a
	9	DI0b
	10	DI1b
	11	DI2b
	12	Common (COM)
	13	DI3b
	14	DI4b
	15	DI5b

Table 2. NI-9411 Power Screw Terminal Pinout

Screw Terminal Pins	Signal
0	BATT
1	GND

DI Driver Module Pinout

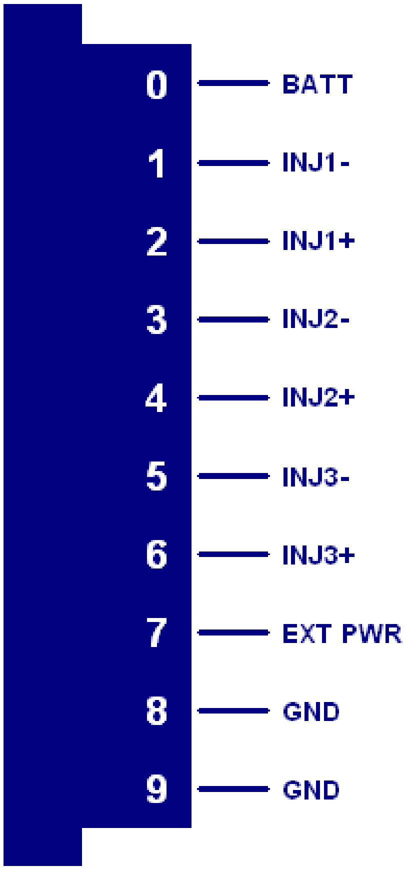
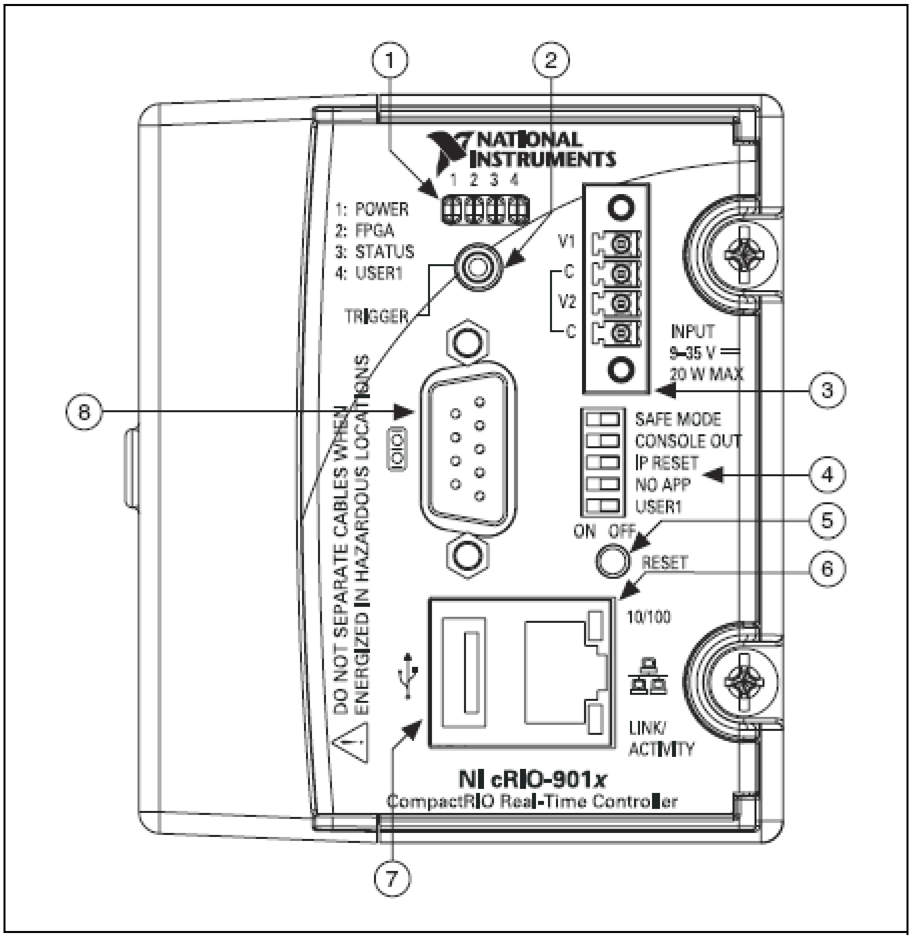


Figure 3. DI Driver Module Screw Terminal Pinout



1	LEDs	5	Reset Button
2	SMB Connector	6	RJ-45 Ethernet Port
3	Power Connector	7	USB Port
4	DIP Switches	8	RS-232 Serial Port

Figure 4. cRIO-9012 Controller Pinout

Powering the System

Warning: The external battery supply input terminals on the Drivven DI Driver Module are not reverse voltage polarity protected. Connecting power to the module in reverse polarity will damage the module. This event is not covered by the warranty. Please refer to the DrivvenReverseBatteryNotice.pdf document (available on the website) for a recommended solution for protecting a system from reverse battery polarity.

The SADI Driver System requires power from a battery or switching power supply ranging from 9V to 32V. If the source is ever expected to drop briefly below 9V, such as during engine cranking, then a separate power source should be supplied to the cRIO-9012 controller to prevent controller reset. If using a separate power source for the cRIO-9012, it is not necessary to connect the grounds between the two power supplies. But it is okay if they are connected. The power source for the NI-9411, Drivven DI Driver and Low-Side Driver modules should not fall below 7V. The NI-9411 does not need to be connected to the power supply unless the 5V outputs are going to be used.

Refer to the component Pinouts above for proper screw terminal connections for power and ground. Powers and grounds should be connected in a star fashion and not daisy-chained. A daisy-chained power configuration may cause bad fluctuations in power to each module during operation. Drivven recommends using soldered or well-crimped ferrules on all screw terminal connections as shown below in Figure 5.

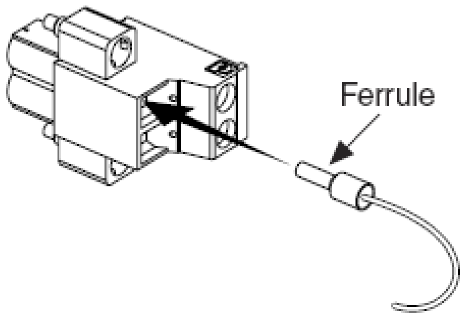


Figure 5. Use of Ferrules on Screw Terminal Connections

Connecting Injectors

Each direct injector will have two leads. The polarity of the injector leads typically does not matter for solenoid injectors, but does matter for piezo injectors. Each DI Driver module has three pairs of screw terminals for three injectors. The screw terminals are labeled INJX– and INJX+.

When connecting more than one injector to a SADI there are a few steps that should be taken to optimize the system. Individual DI driver modules can only drive one injector at a time. Therefore, if any injection events overlap or require simultaneous operation, then they must be operated from different modules. For SADI Driver Systems that have more than one DI driver, care should be taken to optimize the load on the DI Drivers' power supplies. Optimizing the power supplies is obtained by sequencing injection events in a way that prevents two sequential injection events (on different injectors) from happening on the same DI driver. As an example, for a SADI 12 running a V10 with solenoid injectors and firing order of 1-10-9-4-3-6-5-8-7-2 the injectors should be connected as shown in Table 3.

Table 3. Example Routing of NI-9411 Digital Inputs to DI Driver Channels

NI-9411 Digital Input Channel	DI Driver Channel	Cylinder Injector Number
NI-9411-1 Channel 0	DI Driver-1 Channel 1	1
NI-9411-1 Channel 1	DI Driver-1 Channel 2	3
NI-9411-1 Channel 2	DI Driver-1 Channel 3	7
NI-9411-1 Channel 3	DI Driver-2 Channel 1	10
NI-9411-1 Channel 4	DI Driver-2 Channel 2	6
NI-9411-1 Channel 5	DI Driver-2 Channel 3	2
NI-9411-2 Channel 0	DI Driver-3 Channel 1	9
NI-9411-2 Channel 1	DI Driver-3 Channel 2	5
NI-9411-2 Channel 2	DI Driver-3 Channel 3	Not Used
NI-9411-2 Channel 3	DI Driver-4 Channel 1	4
NI-9411-2 Channel 4	DI Driver-4 Channel 2	8
NI-9411-2 Channel 5	DI Driver-4 Channel 3	Not Used

Some solenoid injectors with very low inductance require additional inductance added in series with the injectors in order to obtain proper current control (such as Delphi DFI 1.X injectors). All piezo actuated injectors require series inductance as well. Drivven can provide such inductors to the customer when shipping the system if the customer informs Drivven of the type of injector to be used. These inductors have leads attached which can be soldered in series with the injector leads. It does not matter which lead of the injector with which they are placed in series. Figure 6 below shows the inductor and part number which Drivven supplies, if needed.

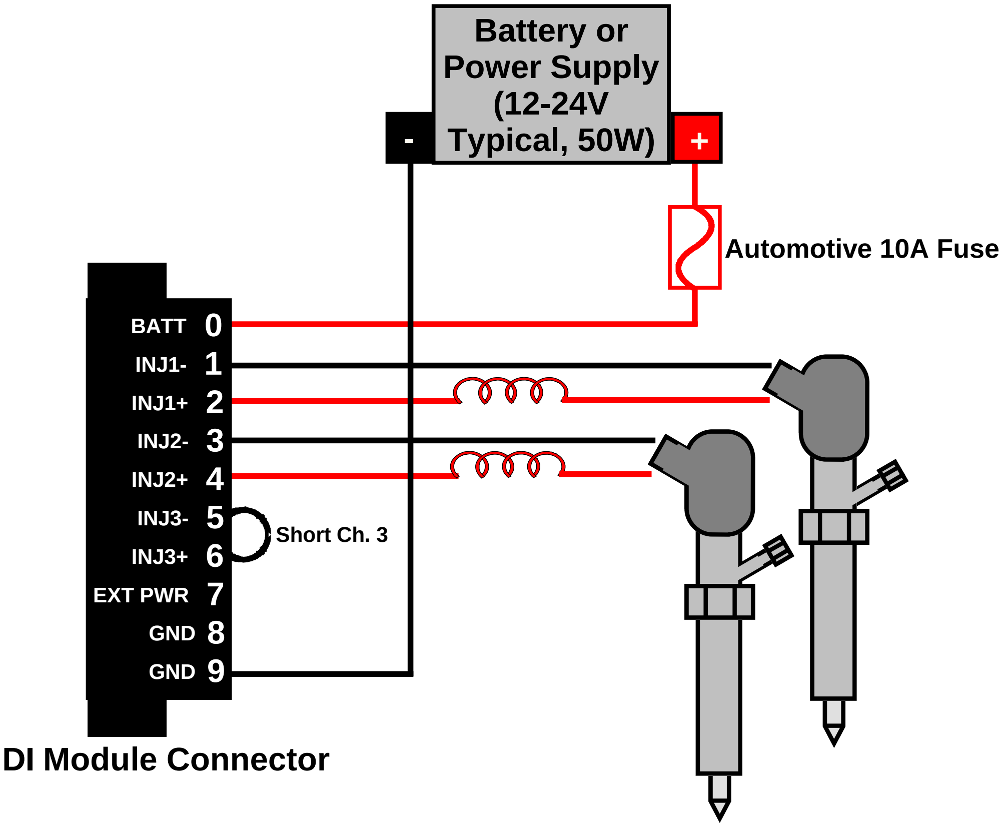


Figure 6. Connecting Piezo Injectors to the DI Driver Module

In-series inductance and shorting channel 3 is mandatory for piezo mode operation only. If you are interested in purchasing the same inductors that we use, the part number is PCR0830-250M-H. These inductors can be ordered directly from Coilws.com. More information on connecting injectors to the DI module can be found in the DI driver manual available at http://drivven.com/Documentation.htm .

Connecting Digital Commands

Each NI-9411 digital input module has six digital inputs. Each SADI Driver System always contains an NI-9411 module in slot one. Each NI-9411 module commands DI Driver modules in the next two slots. The first three input channels of an NI-9411 module command the DI Driver module in the next adjacent slot. The last three input channels of the NI-9411 module command the DI Driver in the second adjacent slot. Refer to Table 3for the routing of digital input channels to DI Driver channels. The table applies to 3, 6, 9 and 12 channel systems. For systems operating piezo injectors, the third channel of each DI Driver module is not used and must be shorted across INJ3+ and INJ3- terminals.

Table 4. Routing of NI-9411 Digital Inputs to DI Driver Channels

NI-9411 Digital Input Channel	**DI Driver Channel**	**Channel Reference Number**
NI-9411-1 Channel 0	DI Driver-1 Channel 1	1
NI-9411-1 Channel 1	DI Driver-1 Channel 2	2
NI-9411-1 Channel 2	DI Driver-1 Channel 3	3
NI-9411-1 Channel 3	DI Driver-2 Channel 1	4
NI-9411-1 Channel 4	DI Driver-2 Channel 2	5
NI-9411-1 Channel 5	DI Driver-2 Channel 3	6
NI-9411-2 Channel 0	DI Driver-3 Channel 1	7
NI-9411-2 Channel 1	DI Driver-3 Channel 2	8
NI-9411-2 Channel 2	DI Driver-3 Channel 3	9
NI-9411-2 Channel 3	DI Driver-4 Channel 1	10
NI-9411-2 Channel 4	DI Driver-4 Channel 2	11
NI-9411-2 Channel 5	DI Driver-4 Channel 3	12

The COM pin (12) of each NI-9411 module should be connected to the external commanding system ground. The “b” pins of the NI-9411 digital input channels must be left disconnected (floating) unless used as a differential input channel (with RS485 compatible or complimentary outputs).

It is good practice to use external 1K pull-up or pull-down resistors on each digital input command line in case the commands are unintentionally disconnected. For active-high commands, the inputs should be pulled low, with the resistor connected between the input and ground. For active-low commands, the inputs should be pulled high, with the resistor connected between the input and 5V or the main power source. If 5V is used, then the 5V output pins (4, 5) of the NI-9411 module may be used for this purpose.

The maximum allowed injection pulse is internally limited to 5 msec, therefore if any channels are commanded continuously, a 5 msec injection may occur on an injector. As long as the input remained active, no other injection events would be allowed because only one injector is allowed to be commanded at a time. If any injection commands to the same DI Driver module overlap, then all injection events for that module will be turned off for the portion where overlap occurs.

When using another engine control system to command the SADI Driver System, it is likely that the port fuel injector or ignition driver outputs of the engine controller will be used. Typically, port fuel injector or ignition driver outputs are low-side driver circuits. This means that one side of the port fuel injector solenoid or ignition coil primary is connected to battery while the other side (low-side) is connected to the engine controller and programmatically switched to ground during fuel injection pulses or ignition dwell, respectively. In order to utilize these low-side driver outputs to command the SADI Driver System, a pull-up resistor must be connected between a constant voltage source (such as 5V or battery) and each of the digital input “a” pins of the NI-9411 module (as described above). Then, the port fuel injector or ignition output pin of the engine controller must be connected to the “a” pin of the digital input channel to be commanded. The pull-up resistors act as a low-current load to the fuel injector or ignition outputs of the engine controller.

This physical arrangement provides for an active-low injection command because the engine controller’s low-side drivers will pull the digital inputs of the NI-9411 module low when injection events are desired. This configuration requires the **DIX_TriggerPolarity** Boolean on the Module Control tab to be set to “ACTIVE LOW” (see Using the Calibration Interface on page 23)

Since most engine controllers have diagnostic features to detect fault conditions with the fuel injector or ignition loads, and pull-up resistors are not inductive loads, the engine controller may report open-circuit faults. Those should be ignored. Figure 7 below illustrates this configuration.

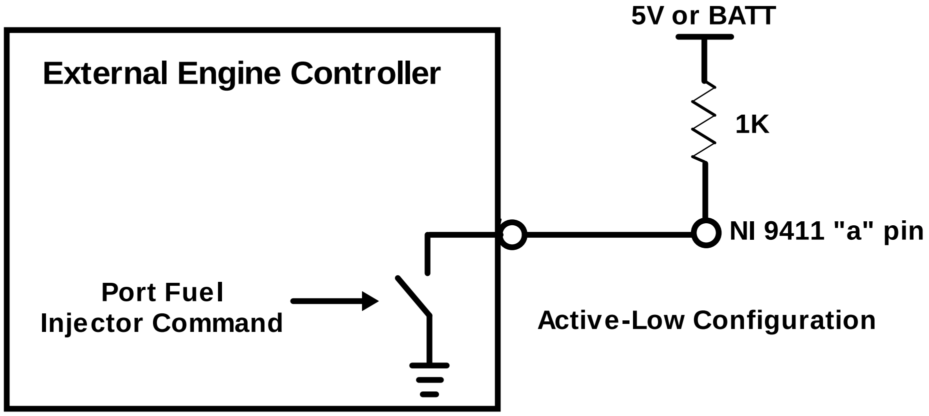


Figure 7. Connecting an Engine Controller Port Fuel Injector Output to the SADI Driver System

If general purpose TTL commands are used from an external controller, then the TTL commands may be directly connected to the “a” pins of the NI-9411 module. Trigger polarity should be set according to the controller output polarity. Pull-up or pull-down resistors should be used on each “a” pin for further robustness in case commands are accidentally disconnected.

DB-15 Connector Parts

Table 4 below is a list of the connector parts included with the SADI Driver System to connect external digital commands to the NI-9411 modules. The crimper tool and pin positioner are available from major electronics distributors (Allied Electronics, Mouser, Digi-Key, etc.) but are not included with the SADI Driver System.

Table 5. Connector Parts List

Description	Mfr.'s Part #
AMP HDP-20 Series 109 15P Receptacle Housing	205163-1
AMP HDP-20 Series 109 Crimp Socket Contact	205090-1
Norcomp D-Sub Connector Hood, 15P 45 Degree	971-015-020R121
AMP D-Sub Insert/Extract Tool	91067-2
Paladin Tools Crimper, 26-20 AWG	PA-1460

Table 6. NI-9411 Module Pigtail Description

Pin Description	Pin #	Wire Color
DI0a	1	Brown
DI0b*	9	Brown
DI1a	2	Orange
DI1b*	10	Orange
DI2a	3	Yellow
DI2b*	11	Yellow
Supply (+5 V Out)	4	Red
DI3a	6	Green
DI3b*	13	Green
DI4a	7	Blue
DI4b*	14	Blue
DI5a	8	Purple
DI5b*	15	Purple
Common (COM)	12	Black

*The "b" pins of the NI-9411 should not be connected unless you are using complimentary digital inputs.

Connections for Rail Pressure Control

This section explains how to connect the Low-Side Driver and NI-9215 with an advanced add-on software package.

Table 7. Low-Side Driver and NI-9215 Connections for Rail Pressure Control

Pin Description	Pin #	Connected To
LS1	1	IMV
LS2	2	IMV
LS3	3	HPV
LS4	4	HPV
AI0a	0	Rail Pressure Sensor +
AI0b	1	Rail Pressure Sensor -

LS#= Low-Side Driver channel #
AI#=NI-9215 channel #
IMV=Inlet Metering Valve, typically found on the fuel pump
HPV=High Pressure Valve, typically found on the common rail, not always present

Figure 8 below is a graphical representation of the connections between the modules and the sensor and actuators required for rail pressure control. For more information about connecting to the Low-Side Driver refer to its manual.

Battery voltage needs to be measured in order for the Low-Side Driver to be able to control the IMV and HPV correctly. Battery voltage is measured by the DI driver so the power supply for the DI driver and the high side of IMV and HPV needs to be the same power supply. The battery voltage measurement is used to limit the maximum duty cycle of the IMV and HPV so that the current does not exceed the capabilities of the channels they are connected to. The voltage of the power supply for the SADI is not read through the NI-9215 because it is limited to reading ±10volts.

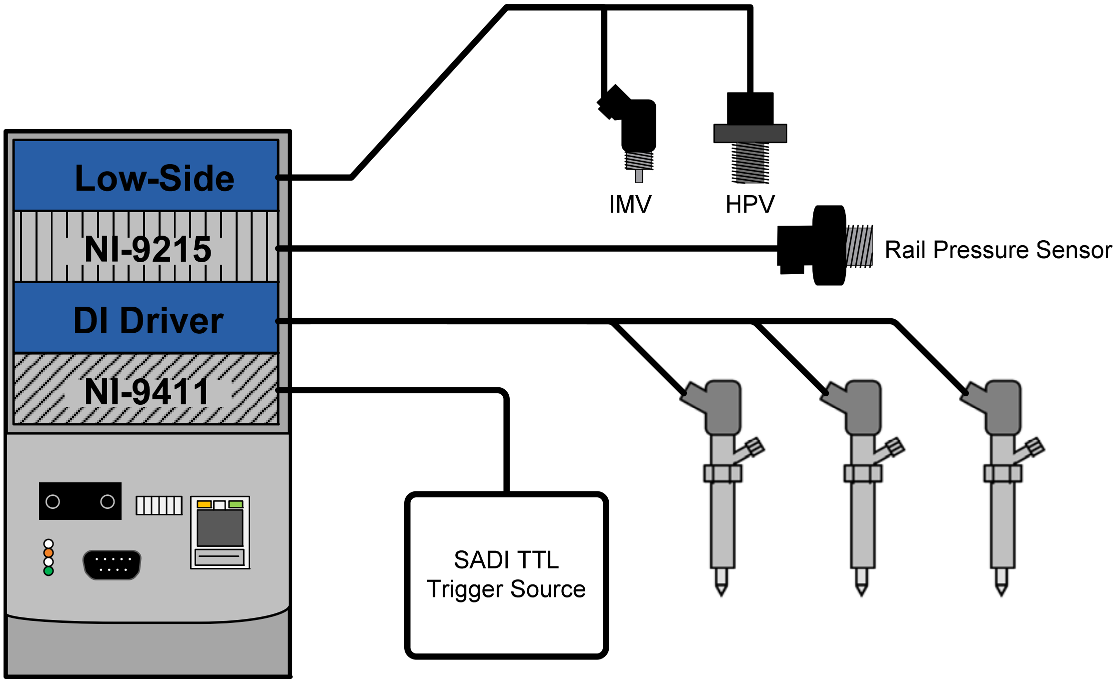


Figure 8. Advanced SADI Driver Connected to Control Rail Pressure

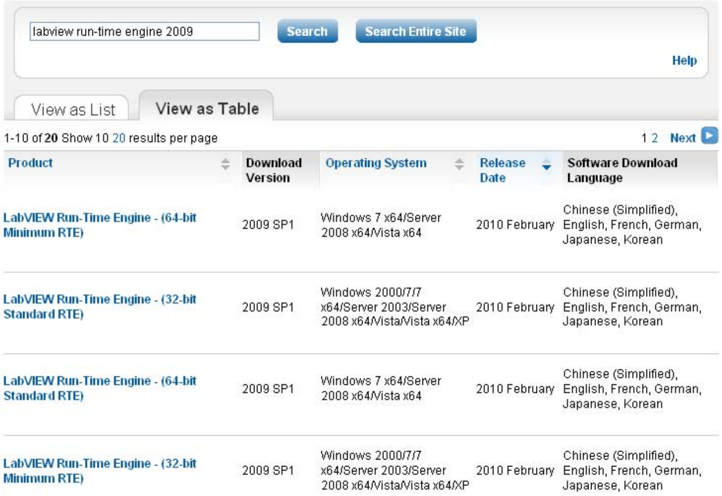
Using the SADI Driver System

The SADI Driver System is delivered with an installed executable which is marked to begin execution when the system boots. The power and boot sequence takes approximately 20 seconds. When the SADI Driver System application is running, the User LED on the cRIO-9012 controller will begin blinking at a rate of approximately 4 Hz. If the application detects a critical fault, such as a short circuit, the system will prevent any further injection events and shutdown the internal boost power supplies. A non-critical fault, such as an open circuit will clear itself when the condition is removed. Any present fault condition is signaled by a steady ON User LED on the cRIO-9012 controller. Critical fault conditions can be cleared by cycling power to the entire system or by connecting to the calibration interface and manually clearing the fault.

The SADI Driver System is delivered to customers with a calibration such that the individual DI Driver modules are not enabled when the system powers up. Therefore, the customer must connect to the calibration interface to enable the modules, make injection current profile modifications, if necessary, and save the calibration before the system may be used stand-alone in the field. After calibrations are saved, those calibrations will be loaded and used each time the system powers up. For example, if the user wants DI Driver modules 1 and 2 to be enabled upon power up, but modules 3 and 4 to be disabled, then those settings must be saved within the calibration interface. If the customer provided Drivven with an injector for current profile calibration beforehand, then those calibrations will already be saved when the customer receives the system.

Connecting to the System Calibration Interface

The SADI Driver System cRIO-9012 controller is delivered to customers with an IP address of 192.168.0.131, unless otherwise specified by the customer. The controller can also be configured to operate in DHCP mode. The first thing that must be done in order to connect to the calibration interface is install the National Instruments Standard Run-time Engine 2009 sp1 or later (as required by the version of CalVIEW being used). Go to the National Instruments website support page at ni.com/support and select the “Drivers and Updates” from the left column. Enter a search term of “Run-time Engine 2009 sp1” and then select the appropriate Standard (not minimum) Run-time Engine version required by CalVIEW. This is shown below in Figure 9, properly configured to search for the LabVIEW Run-time Engine. Be sure to install any patches if necessary.



1-10 of **20** Show 10 20 results per page 1 2 Next

Product	Download Version	Operating System	Release Date	Software Download Language
LabVIEW Run-Time Engine - (64-bit Minimum RTE)	2009 SP1	Windows 7 x64/Server 2008 x64/Vista x64	2010 February	Chinese (Simplified), English, French, German, Japanese, Korean
LabVIEW Run-Time Engine - (32-bit Standard RTE)	2009 SP1	Windows 2000/7/7 x64/Server 2003/Server 2008 x64/Vista/Vista x64/XP	2010 February	Chinese (Simplified), English, French, German, Japanese, Korean
LabVIEW Run-Time Engine - (64-bit Standard RTE)	2009 SP1	Windows 7 x64/Server 2008 x64/Vista x64	2010 February	Chinese (Simplified), English, French, German, Japanese, Korean
LabVIEW Run-Time Engine - (32-bit Minimum RTE)	2009 SP1	Windows 2000/7/7 x64/Server 2003/Server 2008 x64/Vista/Vista x64/XP	2010 February	Chinese (Simplified), English, French, German, Japanese, Korean

Figure 9. Searching for the LabVIEW Run-time Engine

Go to Drivven's website and download and install the latest version of CalVIEW at: http://www.drivven.com/visitor_download/Software/CalVIEW2.90.zip. Please follow the instructions in the CalVIEW user manual for properly activating CalVIEW on your computer.

Connect the cRIO-9012 controller to an Ethernet network using the RJ-45 Ethernet port on the controller. Use a Category 5 (CAT-5) Ethernet cable to connect the controller to an Ethernet hub, or directly to a computer. When connecting the cRIO controller directly to your host computer, without any hubs or switches in between, your computer must be on the same IP subnet as the cRIO controller in order for communication to take place. For example, if the cRIO controller IP address is 192.168.0.131 then your host computer must have an IP address of 192.168.0.X, where "X" is anything between 1 and 255, but not the same as the cRIO controller. This can be done by configuring your computer TCP/IP settings manually and setting a fixed IP address. The netmask should be set to 255.255.255.0. The DNS and Gateway IP addresses should be left blank.

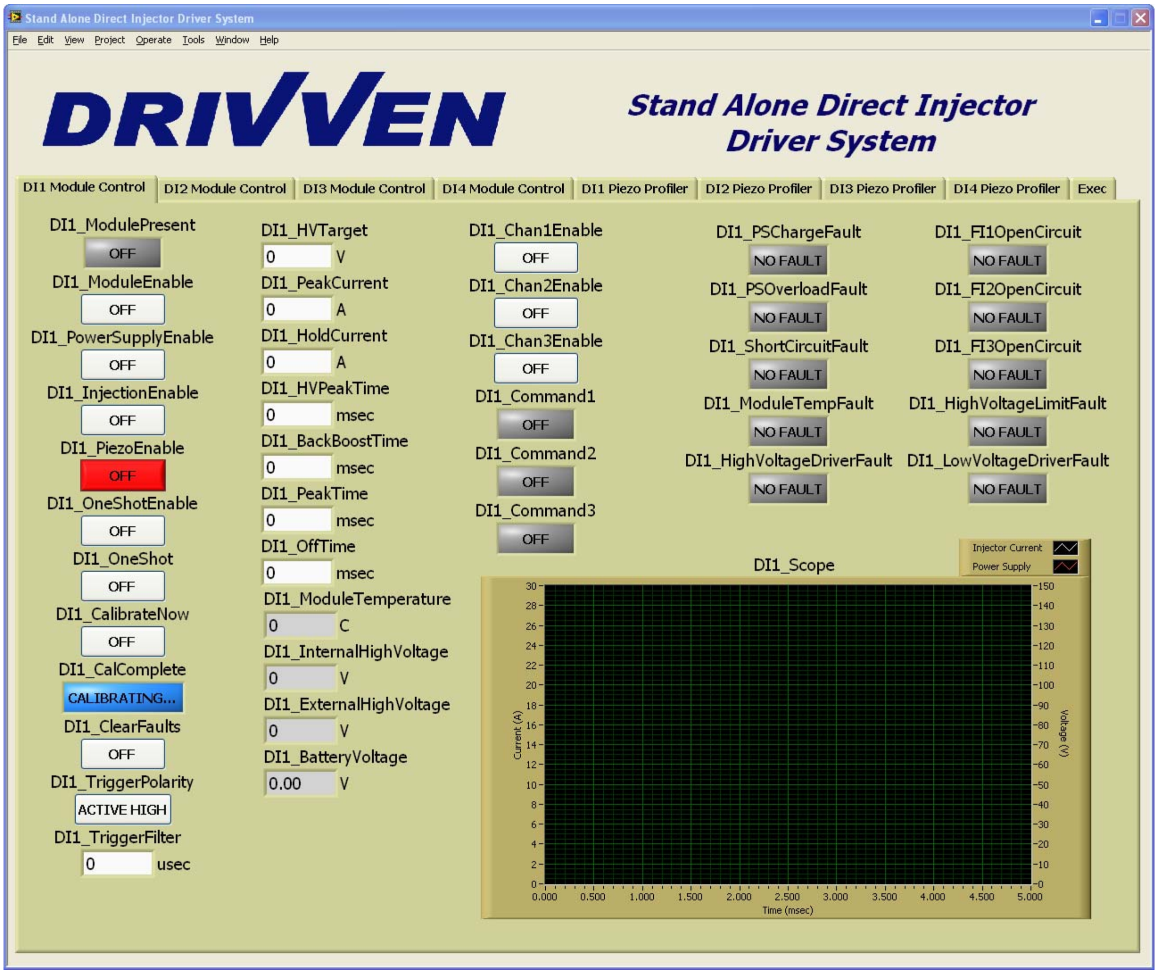
It is possible to put the cRIO controller on a network and configure the cRIO controller for DHCP mode so that it will acquire TCP/IP settings from your network router. In order to reconfigure the cRIO controller TCP/IP settings, please refer to the following section titled "Changing the IP Address."

The cRIO-9012 controller (target) must be powered and the user LED on the front must be blinking in order for a connection to be made with CalVIEW.

Open CalVIEW on the host computer, right click inside the target item list and select "Set New Target Address." If the cRIO-9012 controller and your computer are on the same subnet (default = 192.168.0.X) then CalVIEW will locate the SADI Driver System on the network automatically. The pop-up CalVIEW dialog box should also locate a host VI and a CalVIEW paring on the cRIO controller. Press the button to tell CalVIEW to download these files and press OK. CalVIEW will take a few seconds to transfer these files from the cRIO controller to your computer and use them to make the proper user interface connection. Then press the green "RUN" button at the top of the CalVIEW console to start the user interface.

The user interface will look similar to the panel shown in Figure 10. Figure 10 shows a calibration interface for a 12 channel SADI Driver System. Systems with fewer channels will have fewer tabs.

Note: The IP address of the host computer must be on the same subnet as the cRIO-9012 controller in order to connect to the calibration interface.



DI1_ModulePresent OFF
DI1_ModuleEnable OFF
DI1_PowerSupplyEnable OFF
DI1_InjectionEnable OFF
DI1_PiezoEnable OFF
DI1_OneShotEnable OFF
DI1_OneShot OFF
DI1_CalibrateNow OFF
DI1_CalComplete CALIBRATING...
DI1_ClearFaults OFF
DI1_TriggerPolarity ACTIVE HIGH
DI1_TriggerFilter 0 usec

DI1_HVTarget 0 V
DI1_PeakCurrent 0 A
DI1_HoldCurrent 0 A
DI1_HVPeakTime 0 msec
DI1_BackBoostTime 0 msec
DI1_PeakTime 0 msec
DI1_OffTime 0 msec
DI1_ModuleTemperature 0 C
DI1_InternalHighVoltage 0 V
DI1_ExternalHighVoltage 0 V
DI1_BatteryVoltage 0.00 V

DI1_Chan1Enable OFF
DI1_Chan2Enable OFF
DI1_Chan3Enable OFF
DI1_Command1 OFF
DI1_Command2 OFF
DI1_Command3 OFF

DI1_PSChargeFault NO FAULT
DI1_PSOverloadFault NO FAULT
DI1_ShortCircuitFault NO FAULT
DI1_ModuleTempFault NO FAULT
DI1_HighVoltageDriverFault NO FAULT

DI1_FI1OpenCircuit NO FAULT
DI1_FI2OpenCircuit NO FAULT
DI1_FI3OpenCircuit NO FAULT
DI1_HighVoltageLimitFault NO FAULT
DI1_LowVoltageDriverFault NO FAULT

Figure 10. Calibration Interface Screen Shot

The user may press the square red stop button on the CalVIEW console to stop the host user interface. This will not stop the cRIO-9012 target application. If the user desires to also stop the target application, then the red “Stop Target” button should be pressed.

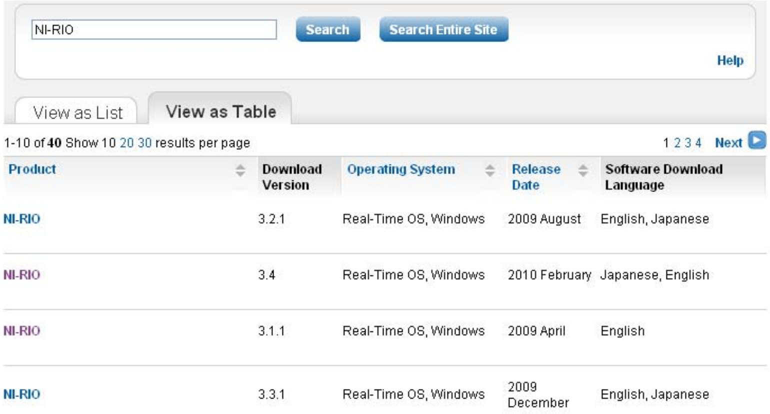
Other than being manually stopped (“Stop Target”) or powered down by the user, the CalVIEW interface is always available to the user, even during normal system operation in the field. Viewing the calibration interface does not limit the performance of the system in any way. The CalVIEW interface may be started and stopped any number of times while the target application is running.

When the user makes changes to the parameters through the user interface host VI, they may be saved to the target by pressing the purple disk (Save Calibration as Target Default) button on the CalVIEW console.

There are several other features of CalVIEW that are not discussed here, such as saving a CalVIEW settings file for quickly starting the host user interface. Those features are discussed in detail within the CalVIEW user manual which is installed to your Windows Start menu under Programs->Drivven->CalVIEW.

Changing the IP Address

If the customer needs to change the TCP/IP settings for the cRIO-9012 controller, then an additional software component from National Instruments must be installed. It is the NI-RIO driver. The properly configured software search from the National Instruments website is shown below in Figure 11. Select and install the latest version available.



1-10 of **40** Show 10 20 30 results per page 1 2 3 4 Next

Product	Download Version	Operating System	Release Date	Software Download Language
NI-RIO	3.2.1	Real-Time OS, Windows	2009 August	English, Japanese
NI-RIO	3.4	Real-Time OS, Windows	2010 February	Japanese, English
NI-RIO	3.1.1	Real-Time OS, Windows	2009 April	English
NI-RIO	3.3.1	Real-Time OS, Windows	2009 December	English, Japanese

Figure 11. Searching for the NI-RIO Driver

When installing NI-RIO, accept all of the default components to install. After installing NI-RIO, you will be able to run the National Instruments program called Measurement and Automation Explorer, or MAX.

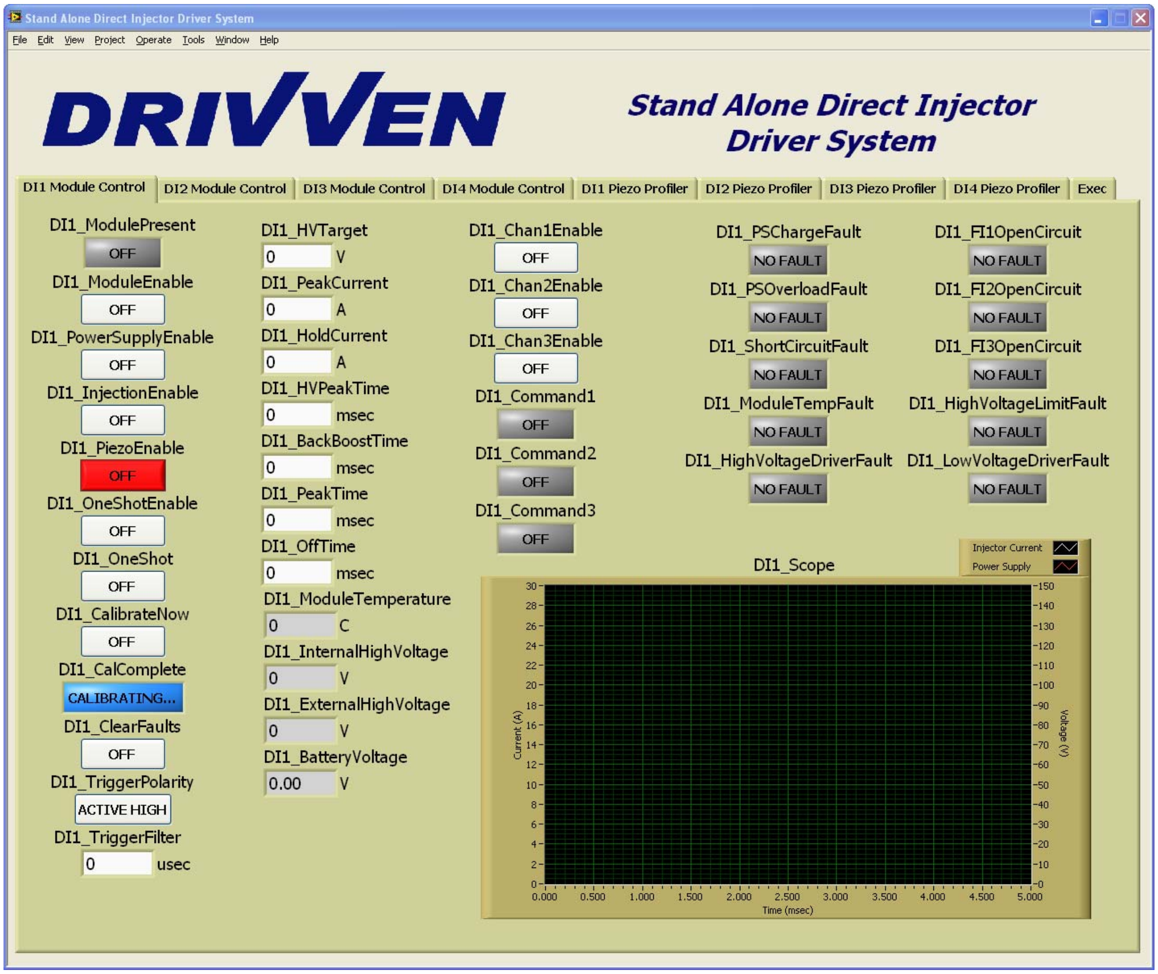
There is a series of dip switches on the front panel of the cRIO-9012 controller. Turn the dip switch labeled IP RESET to ON (to the left). Using a small Phillips screwdriver, press the recessed RESET button to reset the controller. This will cause the controller to reboot and the IP address to be cleared. Wait for about 10 seconds and set the IP RESET dip switch back to OFF. Assuming that the cRIO-9012 controller is already on your network, or directly connected to your computer using a Ethernet cable, you can use MAX to navigate to the controller settings window.

Within MAX, you can navigate to the “Remote Systems” and find your cRIO-9012 controller having the special IP address of 0.0.0.0. Click on this system to view the system identification and IP settings in the window to the right. Proceed to make any necessary changes and apply them.

Note that the IP RESET procedure is only necessary if the preconfigured IP address from Drivven (192.168.0.131) is not compatible with your network or your computer’s subnet. If it is compatible, then you should be able to find the system within MAX without resetting the IP address. Then you can proceed to make the necessary changes and apply them. Keep in mind that if you are connecting to the cRIO system point-to-point, directly with your computer, then be sure to leave the Gateway and DNS fields empty. Otherwise, CalVIEW will have difficulty locating the system. If the cRIO is placed on an existing network, be sure that the Gateway and DNS fields are entered correctly, or CalVIEW will still have difficulty. It may be best to leave the DNS field empty at all times, unless required for a specific reason.

Note: The IP address of the host computer must be on the same subnet as the cRIO-9012 controller in order to find the system within MAX (except for the case where the cRIO-9012 controller IP address is reset to 0.0.0.0).

Using the Calibration Interface



DI1_ModulePresent: OFF
DI1_ModuleEnable: OFF
DI1_PowerSupplyEnable: OFF
DI1_InjectionEnable: OFF
DI1_PiezoEnable: OFF
DI1_OneShotEnable: OFF
DI1_OneShot: OFF
DI1_CalibrateNow: OFF
DI1_CalComplete: CALIBRATING...
DI1_ClearFaults: OFF
DI1_TriggerPolarity: ACTIVE HIGH
DI1_TriggerFilter: 0 usec

DI1_HVTarget: 0 V
DI1_PeakCurrent: 0 A
DI1_HoldCurrent: 0 A
DI1_HVPeakTime: 0 msec
DI1_BackBoostTime: 0 msec
DI1_PeakTime: 0 msec
DI1_OffTime: 0 msec
DI1_ModuleTemperature: 0 C
DI1_InternalHighVoltage: 0 V
DI1_ExternalHighVoltage: 0 V
DI1_BatteryVoltage: 0.00 V

DI1_Chan1Enable: OFF
DI1_Chan2Enable: OFF
DI1_Chan3Enable: OFF
DI1_Command1: OFF
DI1_Command2: OFF
DI1_Command3: OFF

DI1_PSChargeFault: NO FAULT
DI1_PSOverloadFault: NO FAULT
DI1_ShortCircuitFault: NO FAULT
DI1_ModuleTempFault: NO FAULT
DI1_HighVoltageDriverFault: NO FAULT

DI1_FI1OpenCircuit: NO FAULT
DI1_FI2OpenCircuit: NO FAULT
DI1_FI3OpenCircuit: NO FAULT
DI1_HighVoltageLimitFault: NO FAULT
DI1_LowVoltageDriverFault: NO FAULT

Figure 12. Calibration Interface Screen Shot

DIX Module Control Tab (see Figure 1)

DIX_ModuleEnable (Boolean): If a DI driver module is inserted in the proper slot, externally powered, and DIX_ModuleEnable is TRUE, then the controller begins communicating with the module and allows the module to operate. When the module is properly recognized, then the DIX_ModulePresent Boolean will be set to TRUE. Also, after a few seconds, the DIX_CalComplete Boolean will be set to TRUE.

DIX_PowerSupplyEnable (Boolean): When TRUE, the module internal boost power supply is enabled and will maintain the working voltage specified by DIX_HVTarget. When FALSE, the module internal boost power supply is disabled.

DIX_InjectionEnable (Boolean): When TRUE, the module injection control circuitry is enabled. When FALSE, the module injection control circuitry is disabled. This parameter does not generate any fuel pulses. It only enables the driver circuitry to operate when fuel commands are generated.

DIX_CalibrateNow (Boolean): When this button is pressed, the DI Driver module will be calibrated with HVTarget, PeakCurrent, HoldCurrent, HVPeakTime, BackBoostTime, PeakTime, OffTime and various piezo settings. The DIX_CalComplete output Boolean will be FALSE during calibration and will be set TRUE when calibration is complete. The calibration procedure is automatically performed when modules are enabled and also when the system is powered up. Whenever a DI Driver module parameter listed above is changed, then the CalibrateNow button must be pressed in order to see its effect on injection current profiles.

DIX_ClearFaults (Boolean): When this one-shot is pressed, all critical faults of the DI Driver module will be cleared.

DIX_HVTarget (V): The working voltage set point of the internal boost power supply.

DIX_PeakCurrent (A): This is the current level which will be targeted by the injector driver circuit during the HVPeakTime (msec) period. However, the high voltage portion of the current profile is primarily based on HVPeakTime (msec). PeakCurrent (A) is only a backup during the high voltage phase since current is not sensed fast enough to be effective. This value should be set to the peak current level obtained from the OEM injector operation.

DIX_Peak2Current (A): This is the current level which will be targeted by the injector driver circuit from the end of HVPeakTime (msec) to the end of PeakTime (msec), using battery voltage. This value should be set to the current level which best implements a similar current profile to the OEM operation. If this intermediate current level is not necessary, then set the value equal to the hold current level. If your DI Driver module contains firmware previous to version E1, then the driver will use the PeakCurrent value for the remainder of PeakTime.

DIX_HoldCurrent (A): The target current level which is driven through the injector solenoid after the expiration of PeakTime, and until the end of the injection pulse.

DIX_HVPeakTime (msec): The time period at the beginning of the injection pulse during which the high voltage supply is used to drive current through the injector solenoid. HVPeakTime may range from 0msec to 0.4msec. This parameter should be adjusted in small increments/decrements (0.005 msec) to achieve the initial peak current level using high voltage.

DIX_BackBoostTime (msec): The time period at the end of the injection pulse for which the back-emf of the injector solenoid is directed to the internal boost power supply. BackBoostTime may range from 0msec to 1.6msec.

DIX_OneShotEnable (Boolean): When TRUE, the OneShot mode is enabled and the OneShot Boolean may be pressed to generate a 1 msec one-shot pulse to channel 1 of the module.

DIX_OneShot (Boolean): When pressed, and OneShotEnable is TRUE, the DI Driver module will deliver a 1 msec one-shot pulse to channel 1. This Boolean returns to the FALSE state automatically.

DIX_PeakTime (msec): Determines the length of time that the driver circuit will use peak current as the current control threshold. PeakTime may range from 0 msec to 0.6 msec.

DIX_TriggerFilter (usec): Determines the amount of digital input command signal noise rejection. For example, a value of 1 would reject a low or high signal glitch of up to 1 usec width. Please note that the filter also causes a signal delay of the same amount of time. This value can range from 0 to 1638 usec.

CriticalFault (Boolean): Indicates that a critical fault occurred with any of the DI Driver modules. This Boolean will sometimes indicate TRUE while the modules are being calibrated. This is normal behavior and can be ignored.

DIX_ChanXEnable (Boolean): When ON, channel X is enabled and is operated according to TriggerPolarity and the respective external command.

DIX_PiezoEnable (Boolean): When ON, the DI Driver module will operate in piezo mode according to the settings on the DIX Piezo Profiler tab. This button is highlighted red when OFF to provide warning that solenoid injectors are being controlled. This is because piezo injectors can be damaged when controlled in solenoid mode.

DIX_CommandX (Boolean): These Boolean indicators show the high/low level of the external command.

DIX_XXXXFault (Boolean): These Boolean indicators show various fault conditions with the DI Driver module. See the DI Driver Module Kit manual for details.

DIX_ModuleTemperature (C): Indicates the internal DI Driver module temperature. The shutdown fault threshold is 80 C.

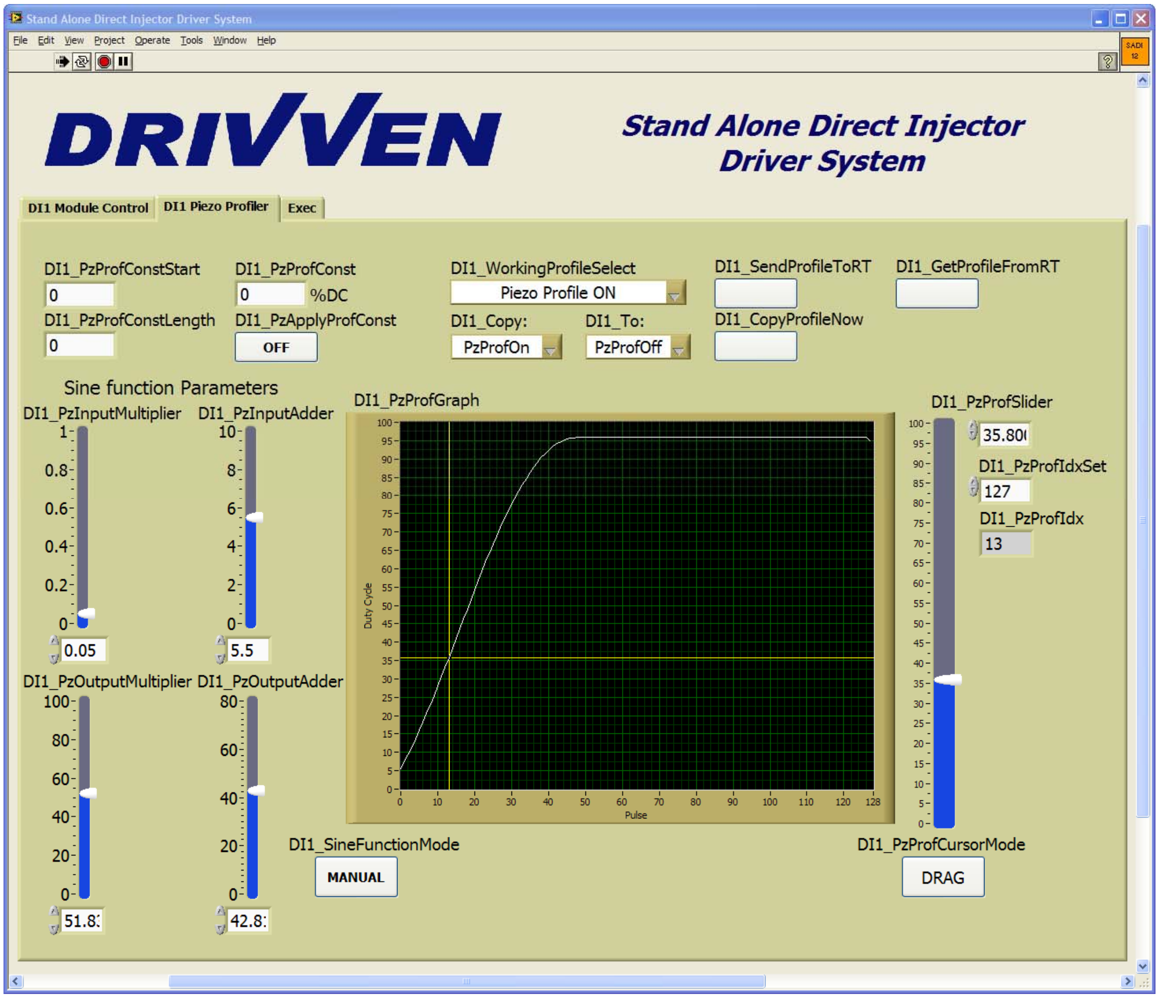
DIX_InternalHighVoltage (V): Indicates the internal boost power supply voltage.

DIX_ExternalHighVoltage (V): Indicates the external high voltage supply level. If the external high voltage supply pin is connected to a level greater than 6V, then the internal boost power supply will automatically shut down and rely on external power. Using an external high voltage supply is typically not necessary.

DIX_BatteryVoltage (V): Indicates the external battery voltage supply level. This should be in the range of 9-32V. The internal boost power supply will operate more efficiently and cooler if the battery voltage is in the range of 24-32V. The load and power consumption of the module and internal power supply are dependent on the solenoid load and rate of injection events.

DIX_Scope (A, V): The scope updates at the interval of approximately 2 seconds to show the latest current and voltage profile stored by the DI Driver module.

DIX Piezo Profiler Tab



DI1_PzProfConstStart: 0

DI1_PzProfConstLength: 0

DI1_PzProfConst: 0 %DC

DI1_PzApplyProfConst: OFF

DI1_WorkingProfileSelect: Piezo Profile ON

DI1_Copy: PzProfOn

DI1_To: PzProfOff

DI1_PzInputMultiplier: 0.05

DI1_PzInputAdder: 5.5

DI1_PzOutputMultiplier: 51.8:

DI1_PzOutputAdder: 42.8:

DI1_PzProfSlider: 35.80(

DI1_PzProfIdxSet: 127

DI1_PzProfIdx: 13

DI1_SineFunctionMode: MANUAL

DI1_PzProfCursorMode: DRAG

Figure 13. User Interface of DI Piezo Profiler

Please refer to the DI Driver User Manual for further information on calibrating for piezo injectors. Each “Pulse” referred to in this section is equal to 2 usec.

DIX_PzProfConstStart (Pulse): Sets the start location (0-128) for the constant duty cycle value to be applied to the piezo profile.

DIX_PzProfConstLength (Pulses): Sets the length for the constant duty cycle value to be applied to the piezo profile.

DIX_PzProfConst (%DC): Sets the duty cycle constant to be applied to the piezo profile.

DIX_PzApplyProfConst (Boolean): Applies the constant duty cycle value to the piezo profile.

DIX_SineFunction Mode (Boolean): When set to FUNCTION, then a sine wave is applied to the piezo profile. When set to MANUAL, then the piezo profile may be adjusted manually at individual points.

DIX_PzInputMultiplier (SGL): Scales the input to the sine function, effectively adjusting the period of the sine wave.

DIX_PzInputAdder (SGL): Offsets the input to the sine function, effectively shifting the sine wave horizontally.

DIX_PzOutputMultiplier (SGL): Scales the output of the sine function, effectively adjusting the amplitude of the sine wave.

DIX_PzOutputAdder (SGL): Offsets the output of the sine function, effectively shifting the sine wave vertically.

DIX_PzProfCursorMode (Boolean): When in DRAG mode, allows the user to drag the cursor to profile locations for setting new manual duty cycle values. When in CONTROL mode, allows the user to set the manual profile position using DIX_PzProfIdx.

DIX_PzProfIdx (SGL): When in CONTROL mode, sets the manual profile position for editing.

DIX_PzProfIdxSet (I32): Sets the manual duty cycle for the profile location specified by DIX_PzProfIdx.

DIX_WorkingProfileSelect (U8): Selects the local ON or OFF profile to work with.

DIX_Copy: (U8): Selects the local ON or OFF profile to copy to the other selected profile in DIX_To:. This copy procedure is only happening on the local computer, not on the target.

DIX_To: (U8): Selects the local ON or OFF profile to be copied from the other selected profile in DIX_Copy:. This copy procedure is only happening on the local computer, not on the target.

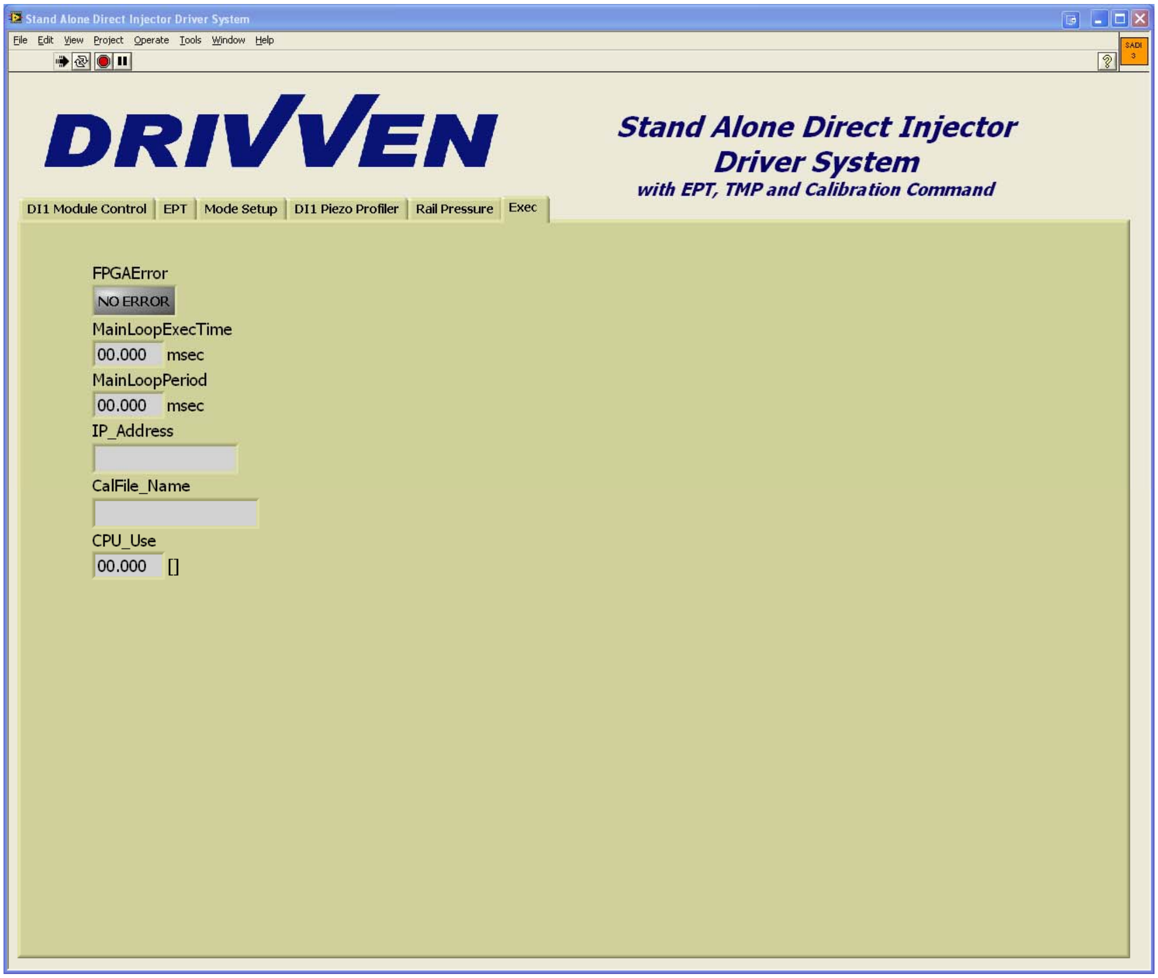
DIX_CopyProfileNow (Boolean): Copies the local profile selected in DIX_Copy: to the profile selected in DIX_To:. This copy procedure is only happening on the local computer, not on the target.

DIX_SendProfileToRT (Boolean): Sends the selected working profile to the cRIO RT target. However, it is not used by the DI Driver module until the CalibrateNow button is pressed. Also, this profile is not saved to the cRIO target calibration file (for use at power up) until the CalVIEW Console Save Default Calibration button is pressed. For typical experimentation with piezo profiles, the user will press the DIX_SendProfileToRT button, followed by the CalibrateNow button. When an acceptable profile is achieved, the CalVIEW Console Save Default Calibration button should be pressed.

DIX_GetProfileFromRT (Boolean): Retrieves the default profile from the cRIO target for experimenting with the profile within the user interface. This button should be pressed when first starting the user interface via CalVIEW to bring the profile into local memory.

DIX_PzProfGraph (SGL): Displays the local memory piezo profile selected in DIX_WorkingProfileSelect.

Execution Tab


Stand Alone Direct Injector Driver System
File Edit View Project Operate Tools Window Help
DRIVVEN
Stand Alone Direct Injector
Driver System
with EPT, TMP and Calibration Command
DI1 Module Control
EPT
Mode Setup
DI1 Piezo Profiler
Rail Pressure
Exec

FPGAError
NO ERROR
MainLoopExecTime
00.000 msec
MainLoopPeriod
00.000 msec
IP_Address
CalFile_Name
CPU_Use
00.000 []

Figure 14. User Interface of Execution Tab

FPGAError (Boolean): Indicates a fault with loading the FPGA portion of the application to the cRIO FPGA chassis. This error should never happen and is only an internal development diagnostic.

MainLoopExecTime (msec): Indicates the amount of time required to execute the main control loop on the cRIO target.

MainLoopPeriod (msec): Indicates the actual period measured for the main control loop on the cRIO target.

Advanced Add-On Package Description

SADI Driver systems can be customized to meet the needs of different projects. The most commonly requested advanced features have been combined into an Advanced Add-on Package. Contact Drivven for a quote for further customization.

The Advanced Add-on Package includes Engine Position Tracking (EPT), rail pressure control, Triggered Multi-Pulse (TMP) and injector calibration mode. It should be noted that the Advanced Add-On package allows for longer injection delays than the base SADI Driver's direct command mode. In order to utilize the rail pressure control software included with the Advanced Add-on Package, two additional cRIO modules have to be purchased. The required modules for rail pressure control are the Drivven Low-Side Driver and the NI-9215. The assigned module slots of the SADI Driver System with an Advanced Add-on Package capable of controlling rail pressure can be seen in Figure 2.

Advanced Software Package

Drivven also offers an Advanced Software Package with each system which includes the following additional features:

Engine Position Tracking (EPT)- This feature takes cam and crank signals as inputs and tracks the crankshaft angle for two-stroke or four-stroke engine cycles. The crank and cam signals can be from an optical encoder/camshaft, or from production crank and cam position sensors (VR sensors require additional hardware). The EPT feature allows the user to enter start of injection angles and durations for injection events. Up to five pulses per cycle are possible.

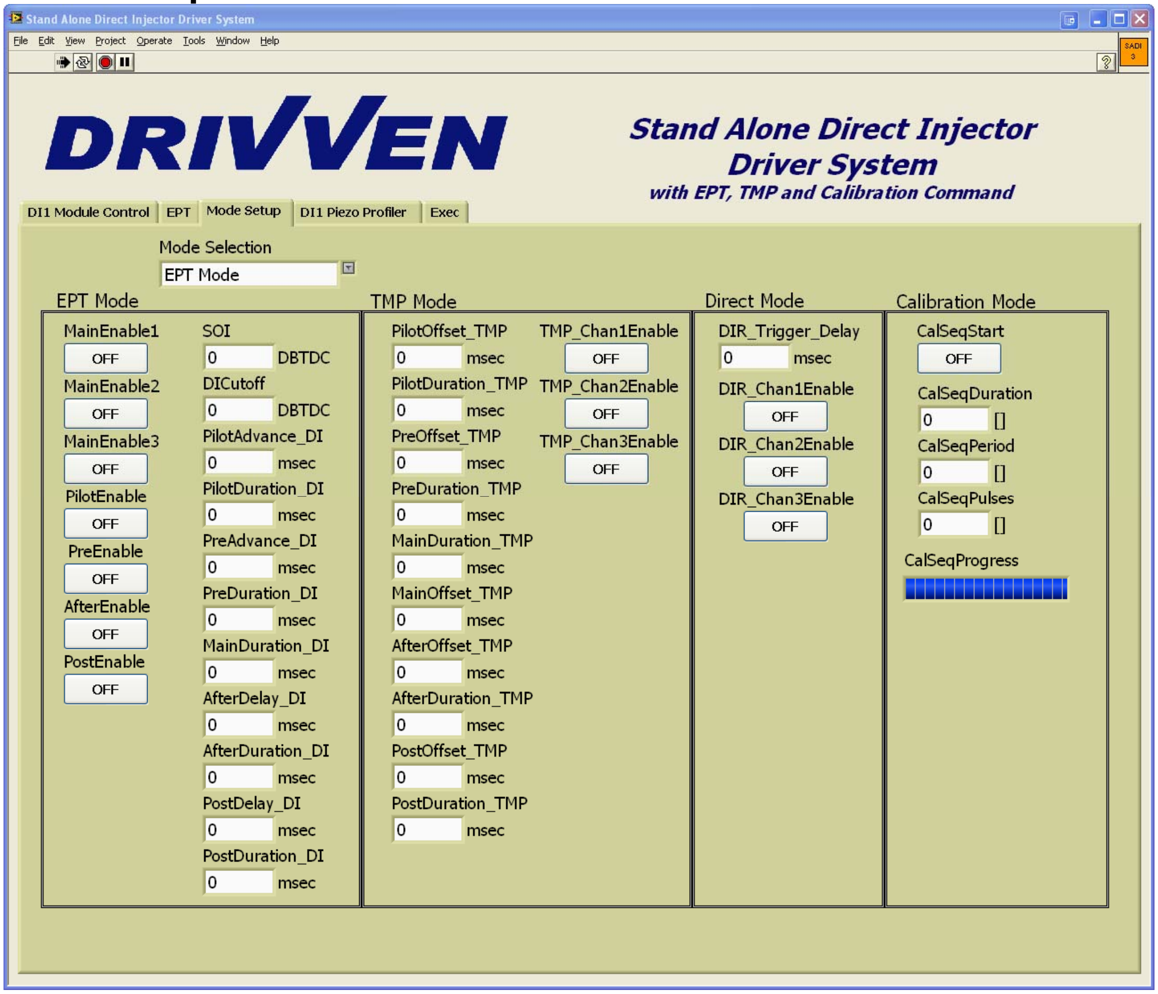
Rail Pressure Control- With the addition of a Drivven Low-Side Driver and an NI-9215 common rail fuel pressure can be controlled. The software uses the NI-9215 to measure the rail pressure while using the Low-Side Driver to PWM the Inlet Metering Valve (IMV) and the High Pressure Valve (HPV) on the rail with a closed loop PID algorithm. Users are allowed to select between different rail pressure control strategies to suit the needs of their system. The recommended setup though is to control the IMV with a PID loop and fix the HPV at a constant duty cycle. The user needs to tune the PID gains to suit the requirements of their system.

Triggered Multi-Pulse (TMP)- This features uses the rising or falling edge of the external digital command as a trigger for a multi-pulse injection sequence for up to five pulses per trigger. The user may define the delay and duration for each of five injection pulses. This is useful for adding multi-pulse injection capabilities to a system that only allows one output trigger per cycle.

Injector Calibration- This feature allows the user to command a programmable number of injections at a fixed duration and interval. This is helpful for measuring the mass of fuel injected per injection event.

In order to use the rail pressure control of the Advanced Add-on Package, a Low-Side Driver and NI-9215 must be purchased in addition to the advanced add-on software package. The different channel-count SADI Driver Systems with rail pressure control capabilities include the following:

Mode Setup Tab


Stand Alone Direct Injector Driver System
File Edit View Project Operate Tools Window Help
DRIVVEN
Stand Alone Direct Injector Driver System
with EPT, TMP and Calibration Command
DI1 Module Control
EPT
Mode Setup
DI1 Piezo Profiler
Exec

Mode Selection: EPT Mode

EPT Mode

- MainEnable1: OFF
- MainEnable2: OFF
- MainEnable3: OFF
- PilotEnable: OFF
- PreEnable: OFF
- AfterEnable: OFF
- PostEnable: OFF
- SOI: 0 DBTDC
- DICutoff: 0 DBTDC
- PilotAdvance_DI: 0 msec
- PilotDuration_DI: 0 msec
- PreAdvance_DI: 0 msec
- PreDuration_DI: 0 msec
- MainDuration_DI: 0 msec
- AfterDelay_DI: 0 msec
- AfterDuration_DI: 0 msec
- PostDelay_DI: 0 msec
- PostDuration_DI: 0 msec

TMP Mode

- PilotOffset_TMP: 0 msec
- PilotDuration_TMP: 0 msec
- PreOffset_TMP: 0 msec
- PreDuration_TMP: 0 msec
- MainDuration_TMP: 0 msec
- MainOffset_TMP: 0 msec
- AfterOffset_TMP: 0 msec
- AfterDuration_TMP: 0 msec
- PostOffset_TMP: 0 msec
- PostDuration_TMP: 0 msec
- TMP_Chan1Enable: OFF
- TMP_Chan2Enable: OFF
- TMP_Chan3Enable: OFF

Direct Mode

- DIR_Trigger_Delay: 0 msec
- DIR_Chan1Enable: OFF
- DIR_Chan2Enable: OFF
- DIR_Chan3Enable: OFF

Calibration Mode

- CalSeqStart: OFF
- CalSeqDuration: 0 []
- CalSeqPeriod: 0 []
- CalSeqPulses: 0 []
- CalSeqProgress

Figure 15. Mode Setup Tab User Interface

The Mode Setup Tab is broken into four different sections of controls and indicators. Each section is independent of one another and only active when the Mode Selection combo box control is set to that section.

EPT Mode

In this mode, engine position tracking is used and the start of the main injection pulse is delivered at a user-specified crank angle degree. The TDCs of the channel's cylinder and the inputs for the EPT are set in the EPT tab. Figure 16 is a diagram explaining the variables that define the injection pulse-train when in EPT mode.

MainEnableX(Boolean): When TRUE, all five fuel pulses are potentially enabled for channel X, depending on the other four enabled Booleans. When FALSE (default), all five fuel pulses are disabled.

PilotEnable(Boolean): When TRUE and MainEnableX is TRUE, the pilot pulse is enabled for all channels. When FALSE (default), the pilot pulse is disabled.

PreEnable(Boolean): When TRUE and MainEnableX is TRUE, the pre pulse is enabled for all channels. When FALSE (default), the pre pulse is disabled.

AfterEnable(Boolean): When TRUE and MainEnableX is TRUE, the after pulse is enabled for all channels. When FALSE (default), the after pulse is disabled.

PostEnable(Boolean): When TRUE and MainEnableX is TRUE, the post pulse is enabled for all channels. When FALSE (default), the post pulse is disabled.

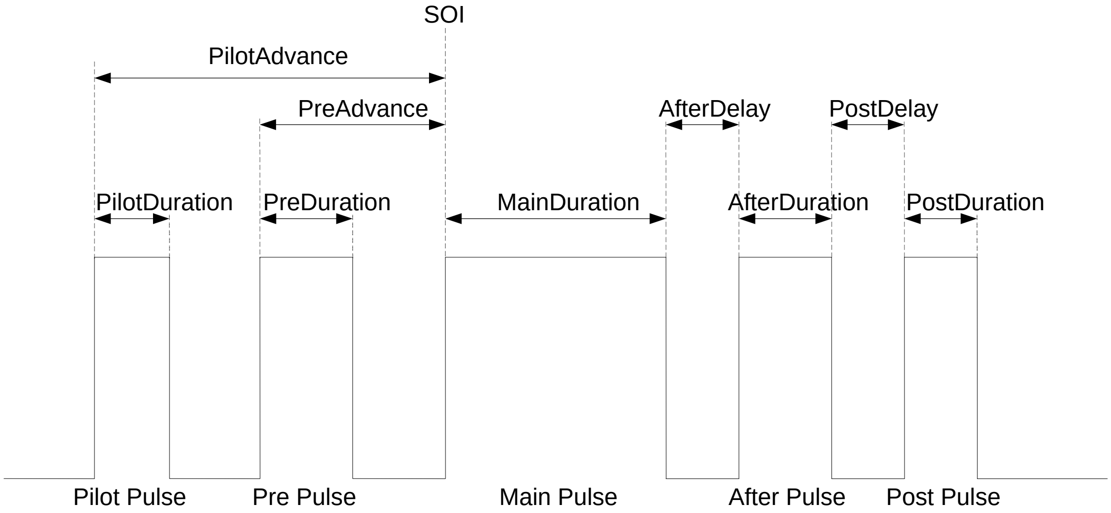


Figure 16. Fuel Pulse Timing Diagram for EPT mode

Start Of Injection (SOI) (CA-DBTDC): The main fuel pulse is generated with a leading edge coinciding with SOI. The length of the pulse will be according to MainDuration_DI. The units of SOI are crank angle degrees before TDC. The timing of the pilot and pre pulses is referenced to SOI. The timing of the after pulse is referenced to the end of the main pulse.

DICutoff(CA-DBTDC): The channels within a single DI Driver module are multiplexed. Therefore a crank-angle position must be set which determines when the software switches operation to the next channel. DICutoff is used for this purpose. One DICutoff value is applied to each channel TDC with units of Degrees Before TDC (DBTDC). Even though some channels may not be used or enabled, each channel's TDC must be in positional order as channel number increases. Below are some example TDC values for each channel.

Example 1 - Correct (positional order is maintained):
TDC1 = 0
TDC2 = 240
TDC3 = 480

Example 2 - Correct (positional order is maintained):
TDC1 = 240
TDC2 = 480
TDC3 = 0

Example 3 - Incorrect (positional order is not maintained):
TDC1 = 0
TDC2 = 480
TDC3 = 240

PilotAdvance_DI(msec): Determines the start time of the pilot pulse with respect to the start of the main pulse. This is a time advance, not a position advance. PilotAdvance is entered as a time with units of milliseconds. On the RT controller this value is internally limited to about 5 msec.

PilotDuration_DI(msec): Determines the length of the pilot fuel pulse. PilotDuration is entered as a time with units of milliseconds. PilotDuration is limited to a value of 5 msec in order to protect the DI driver module.

PreAdvance_DI(msec): Determines the start time of the pre pulse with respect to the start of the main pulse. This is a time advance, not a position advance. PreAdvance is entered as a time with units of milliseconds. PreAdvance is limited to a value of about 5 msec.

PreDuration_DI(msec): Determines the length of the pre fuel pulse. PreDuration is entered as a time with units of milliseconds. PreDuration is is limited to a value of 5 msec in order to protect the DI driver module.

MainDuration_DI(msec): Determines the length of the main fuel pulse. MainDuration is entered as a time with units of milliseconds. MainDuration is limited to a value of 5 msec in order to protect the DI driver module.

AfterDelay_DI(msec): Determines the start time of the after pulse with respect to the end of the main pulse. This is a time delay, not a position delay. AfterDelay is entered as a time with units of milliseconds. AfterDelay is limited to a value of about 5 msec.

AfterDuration_DI(msec): Determines the length of the after fuel pulse. AfterDuration is entered as a time with units of milliseconds. AfterDuration is limited to a value of 5 msec in order to protect the DI driver module.

PostDelay_DI(msec): Determines the start time of the post pulse with respect to the end of the after pulse. This is a time delay, not a position delay. PostDelay is entered as a time with units of milliseconds. PostDelay is limited to a value of about 5 msec.

PostDuration_DI(msec): Determines the length of the post fuel pulse. PostDuration is entered as a time with units of milliseconds. PostDuration is limited to a value of 5 msec in order to protect the DI driver module.

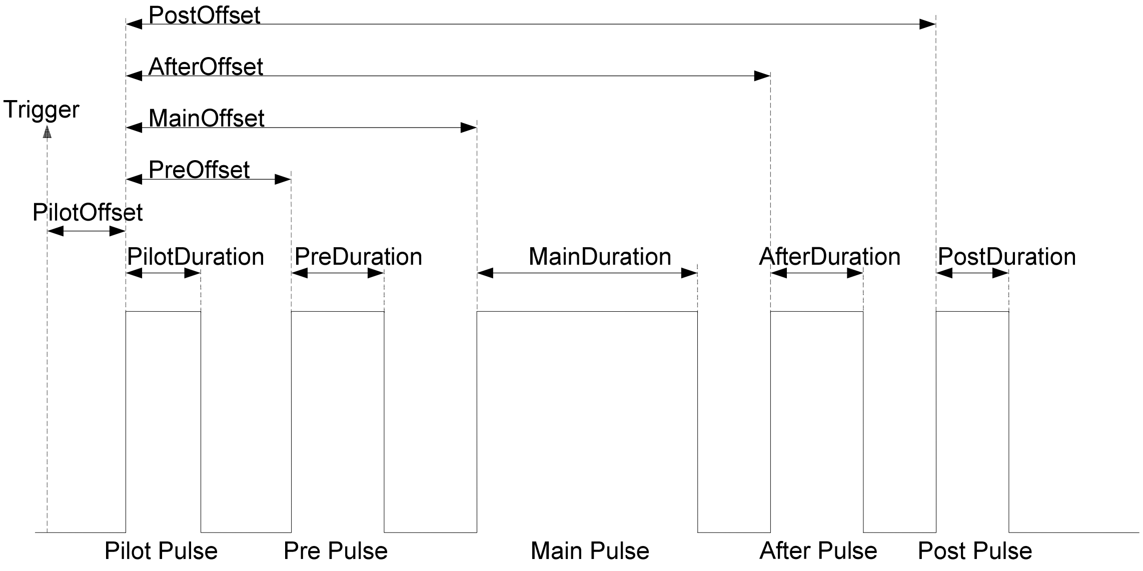


Figure 17. Fuel Pulse Timing Diagram for TMP mode

TMP Mode

PilotOffset_TMP(msec): Determines the initial delay between the incoming trigger and pilot pulse. PilotOffset_TMP is entered as a time with units of milliseconds. On the RT controller all of the TMP offset values are limited to 10 seconds for safety considerations.

PilotDuration_TMP(msec): Determines the length of the pilot fuel pulse. PilotDuration is entered as a time with units of milliseconds. PilotDuration is limited to a value of 5 msec.

PreOffset_TMP(msec): Sets the delay between the start of the pilot pulse and the pre-pulse pulse. On the RT controller all of the TMP offset values are limited to 10 seconds for safety considerations.

PreDuration_TMP(msec): Determines the length of the pre fuel pulse. PreDuration is entered as a time with units of milliseconds. PreDuration is limited to a value of 5 msec.

MainOffset_TMP(msec): Sets the delay between the start of the pilot pulse and the main pulse. On the RT controller all of the TMP offset values are limited to 10 seconds for safety considerations.

MainDuration_TMP(msec): Determines the length of the main fuel pulse. MainDuration is entered as a time with units of milliseconds. MainDuration is limited to a value of 5 msec.

AfterOffset_TMP(msec): Sets the delay between the start of the pilot pulse and the after pulse. On the RT controller all of the TMP offset values are limited to 10 seconds for safety considerations.

AfterDuration_TMP(msec): Determines the length of the after fuel pulse. AfterDuration is entered as a time with units of milliseconds. AfterDuration is limited to a value of 5 msec.

PostOffset_TMP(msec): Sets the delay between the start of the pilot pulse and the post pulse. On the RT controller all of the TMP offset values are limited to 10 seconds for safety considerations.

PostDuration_TMP(msec): Determines the length of the post fuel pulse. PostDuration is entered as a time with units of milliseconds. PostDuration is limited to a value of 5 msec.

TMP_ChanXEnable(msec): Enables channel X of the DI driver(s) to output the TMP pulse train when TMP mode has been selected.

Direct Mode

DIR_Trigger_Delay(msec): Delay in milliseconds between the incoming trigger for any channel and the firing of the enabled channels. The duration of the injection event is defined by the pulse width of the incoming trigger (limited to 5 msec). The maximum amount of delay has been set to 10 seconds for safety considerations.

DIR_ChanXEnable(Boolean): When True, enables the output channel X to be triggered by digital pulses coming on that channel’s designated input channel (see Table 3).

Calibration Mode

CalSeqStart(Boolean): When true, starts firing the number of pulses specified by CalSeqPulses.

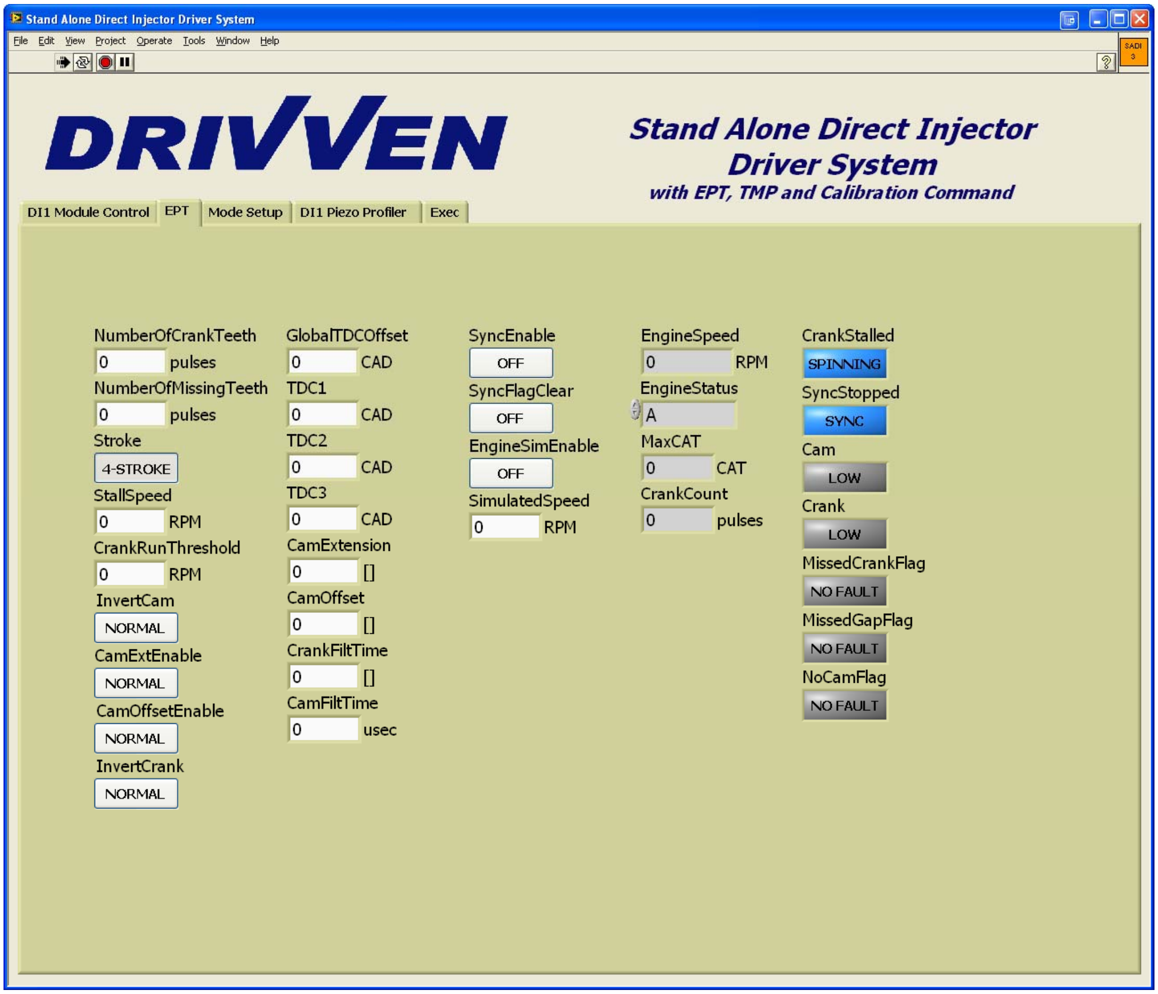
CalSeqDuration(msec): Pulse width of each pulse during a calibration sequence.

CalSeqPeriod(msec): Time period between the rising edge of injection pulses in a calibration sequence.

CalSeqPulses(-): Number of injection pulses in a calibration sequence.

CalSeqProgress(%): Progress through the calibration sequence.

EPT Tab


Stand Alone Direct Injector Driver System
File Edit View Project Operate Tools Window Help
DRIVVEN
Stand Alone Direct Injector Driver System
with EPT, TMP and Calibration Command
DI1 Module Control
EPT
Mode Setup
DI1 Piezo Profiler
Exec

NumberOfCrankTeeth 0 pulses
NumberOfMissingTeeth 0 pulses
Stroke 4-STROKE
StallSpeed 0 RPM
CrankRunThreshold 0 RPM
InvertCam NORMAL
CamExtEnable NORMAL
CamOffsetEnable NORMAL
InvertCrank NORMAL

GlobalTDCOffset 0 CAD
TDC1 0 CAD
TDC2 0 CAD
TDC3 0 CAD
CamExtension 0 []
CamOffset 0 []
CrankFiltTime 0 []
CamFiltTime 0 usec

SyncEnable OFF
SyncFlagClear OFF
EngineSimEnable OFF
SimulatedSpeed 0 RPM

EngineSpeed 0 RPM
EngineStatus A
MaxCAT 0 CAT
CrankCount 0 pulses

CrankStalled SPINNING
SyncStopped SYNC
Cam LOW
Crank LOW
MissedCrankFlag NO FAULT
MissedGapFlag NO FAULT
NoCamFlag NO FAULT

Figure 18. EPT Tab User Interface

The controls and indicators on this tab are not active unless the “Mode Selection” control is set to EPT mode.

Common Variables between N-M, N+1 and ENC

NumberOfCrankTeeth(-): Specifies the number of teeth on the crankshaft trigger wheel as if there were no missing or extra crank teeth. For example, a 60-2 crankshaft pattern would have NumberOfCrankTeeth = 60. For encoders this is the pulses per crank revolution.

Stroke(Boolean): (FALSE=2, TRUE=4) When FALSE, the EPT is tracking teeth from 0 to NumberOfCrankTeeth. When TRUE, the EPT is tracking teeth from 0 to 2*NumberOfCrankTeeth.

StallSpeed(RPM): A measured crankshaft speed less than StallSpeed causes the engine speed to be reported as zero, the CrankStalled output to report as TRUE, and EngineStatus to be “Stalled”.

CrankRunThreshold(RPM): A measured crankshaft speed less than CrankRunThreshold but higher than StallSpeed causes the EngineStatus to be reported as “Cranking”, and

CrankStalled(RPM): to display as “Spinning”. Any engine speed measured above CrankRunThreshold will display “Running” in EngineStatus.

InvertCam(Boolean): Inverts the incoming cam signal.

InvertCrank(Boolean): Inverts the incoming crank signal.

CamExtEnable(Boolean): When enabled, the cam signal is extended (or possibly shortened) by the number of rising edges of the crank signal as specified by CamExtension and referenced to the rising edge of the cam signal. When disabled, the cam signal is unaltered.

CamExtension(Teeth): The number of crank pulses to extend the camshaft signal.

CamOffsetEnable(Boolean): When enabled, the cam signal is offset by the number of rising edges of the crank signal as specified by CamOffset and referenced to the rising edge of the crank signal which immediately follows the rising edge of the cam signal. The offset cam signal will go high at the falling edge of the crankshaft signal immediately following the specified number of rising edges of the crankshaft. The offset cam signal will go low at the following rising edge of the crankshaft signal. When disabled, the camshaft signal is unaltered. When CamOffset is zero, offset cam signal will remain low.

CamOffset(Teeth): The number of crank signal pulses to offset the cam signal.

CrankFiltTime(usec): Number of microseconds that an incoming crank signal needs to be longer then before it will not be considered a glitch.

CamFiltTime(usec): Number of microseconds that an incoming cam signal needs to be longer then before it will not be considered a glitch.

GlobalTDCOffset(CA): Crank angle degrees between the absolute zero position (determine by the EPT function) and TDC1. When TDC1 is before absolute zero position, then GlobalTDCOffset is negative. When TDC1 is after absolute zero position, then GlobalTDCOffset is positive. The absolute zero position is specific to the crankshaft pattern type. For encoder pattern types, assuming an ABZ style encoder, absolute zero position is the rising edge of the first A/B pulse following the rising edge of the Z pulse, while the cam signal is TRUE. For N-M pattern types, absolute zero position is the rising edge of the first crank pulse following the M-tooth gap, while the cam signal is TRUE. For N+1 pattern types, absolute zero position is the rising edge of the first crank pulse following the +1 pulse while the cam signal is TRUE.

TDCX(CA): Crank angle degrees of offset between the TDC1 and TDCX. ”X” represents the channel number for the DI module and not necessarily the cylinder number. The channels within a single DI Driver module are multiplexed. Therefore a crank-angle position must be set which determines when the software switches operation to the next channel. DICutoff is used for this purpose. One DICutoff value is applied to each channel TDC with units of Degrees Before TDC (DBTDC). Even though some channels may not be used or enabled, each channel's TDC must be in positional order as channel number increases. Below are some example TDC values for each channel.

Example 1 - Correct (positional order is maintained):
TDC1 = 0
TDC2 = 240
TDC3 = 480

Example 2 - Correct (positional order is maintained):
TDC1 = 240
TDC2 = 480
TDC3 = 0

Example 3 - Incorrect (positional order is not maintained):
TDC1 = 0
TDC2 = 480
TDC3 = 240

SyncEnable(Boolean): When TRUE, sync is enabled and position tracking will take place if a valid crank and cam signal pattern is presented to the EPT VI at an engine speed greater than the stall speed.

SyncFlagClear(Boolean): When TRUE, any error within the EPT Data output will be cleared.

EngineSimEnable(Boolean): When TRUE, CrankSig and CamSig inputs are internally disconnected from the EPT processor and simulated signals are connected in their place.

SimSpeed(RPM): The simulated speed of the EPT when EngineSimEnable is true. The EPT will not sync to the simulator unless the effective engine speed is greater than the stall speed.

EngineSpeed(RPM): Engine speed output from the EPT.

EngineStatus(-) Combo box indicating the status of the engine to be cranking, stalled or running, depending on the values of StallSpeed and CrankRunThreshold.

MaxCAT(Ticks): Maximum Crank Angle Ticks per engine cycle. A Crank Angle Tick (CAT) is a single unit of angular measure reported by the EPT processor. Reported as a power-of-two angular ticks of crank position travel, having a resolution dependent on binary extrapolation level and the number of physical teeth per crankshaft rotation. For example, if using the N-M EPT VI, which has an extrapolation of 7, the number of CAT per crank tooth would be 2^7=128, and CurrentPosition would be evenly incremented by 128 CAT from one physical tooth to the next. If a 60-2 pattern were used, the maximum number of CAT per crankshaft rotation (cycle) would be 60*128=7680. If the engine was a 4-stroke, the total number of CAT per engine cycle would be 2*60*128=15360.

CrankCount(-): The latest tooth count referenced to crank tooth 0. Tooth 0 is determined according to the pattern that is being tracked and is discussed within the pattern specific EPT documentation. Also see the description for GlobalTDCOffset. CrankCount is provided in terms of a complete engine cycle and is reported as if there were no missing or extra teeth in the pattern. For example, if PATTERN = N-M, NumberOfCrankTeeth = 60 and Stroke = TRUE, then CrankCount will range from 0 to 119.

CrankStalled(Boolean): Set to TRUE when the engine speed is below stall speed.

SyncStopped(Boolean): Set to TRUE while position tracking is not taking place. In other words, no sync.

CamSig(Boolean): The digital input signal representing the camshaft trigger wheel pulse train.

CrankSig(Boolean): The digital input signal representing the crankshaft trigger wheel pulse train.

MissedCrankFlag(Boolean): Set to TRUE upon the detection of a pattern-specific crank feature while the CrankCount value has not yet reached the expected value. For example, a gap has been detected sooner than expected while tracking an N-M pattern. This condition causes a loss of sync. Re-sync is not allowed until the flag is cleared by setting SyncFlagClear to TRUE.

N-M Only

NumberOfMissingTeeth(Teeth): Specifies the number of missing adjacent teeth on the crankshaft trigger wheel for an N-M pattern type. For example, a 60-2 crankshaft pattern would have NumberOfMissingTeeth = 2. This parameter is only available for N-M pattern types.

AllowNoCamStart(Boolean): When AllowNoCamStart is TRUE, and Stroke is TRUE, the EPT will begin sync at the first occurrence of the pattern-specific unique crankshaft feature, such as a crank tooth gap or a plus1 tooth. If the cam signal is TRUE during the gap, then tooth number will start from 0. If the cam signal is FALSE during the gap, then tooth count will start from mid-cycle. When AllowNoCamStart is FALSE, and Stroke is TRUE, sync will only be started when the crank feature is found while the cam signal is true, therefore only starting tooth count from 0. Only set this input to TRUE if it is OK to be in sync 360 degrees out of phase in the case of a disconnected or faulty cam sensor. This parameter is only available for N-M and N+1 pattern types.

MissedGapFlag(Boolean): Set to TRUE upon CrankCount being incremented past the expected value for the location of a gap before the gap is detected. In other words, physical teeth are received at the location where a gap is expected. This condition causes a loss of sync. Re-sync is not allowed until the flag is cleared by setting SyncFlagClear to TRUE.

NoCamFlag(Boolean): Set to TRUE upon detecting 2 consecutive gaps without detecting a TRUE CamSig (Stroke = TRUE). CamSig is only sampled at gap detection. This condition does not cause a loss of sync. NoCamFlag is cleared by setting SyncFlagClear to TRUE.

N+1 Only

Plus1Location(Boolean): (FALSE=Adv, TRUE=Ret) When FALSE, the EPT is configured to look for a Plus1 tooth which is advanced from the mid-point between two evenly spaced crank teeth. An advanced Plus1 tooth should be advanced from the mid-point by approximately ¼ of the total normal tooth spacing. When TRUE, the EPT is configured to look for a Plus1 tooth which is retarded from the midpoint between two evenly spaced crank teeth. A retarded Plus1 tooth should be retarded from the mid-point by approximately ¼ of the total normal tooth spacing. This parameter is only available for Plus1 pattern types.

AllowNoCamStart(Boolean): When AllowNoCamStart is TRUE, and Stroke is TRUE, the EPT will begin sync at the first occurrence of the pattern-specific unique crankshaft feature, such as a crank tooth gap or a plus1 tooth. If the cam signal is TRUE during the gap, then tooth number will start from 0. If the cam signal is FALSE during the gap, then tooth count will start from mid-cycle. When AllowNoCamStart is FALSE, and Stroke is TRUE, sync will only be started when the crank feature is found while the cam signal is true, therefore only starting tooth count from 0. Only set this input to TRUE if it is OK to be in sync 360 degrees out of phase in the case of a disconnected or faulty cam sensor. This parameter is only available for N-M and N+1 pattern types.

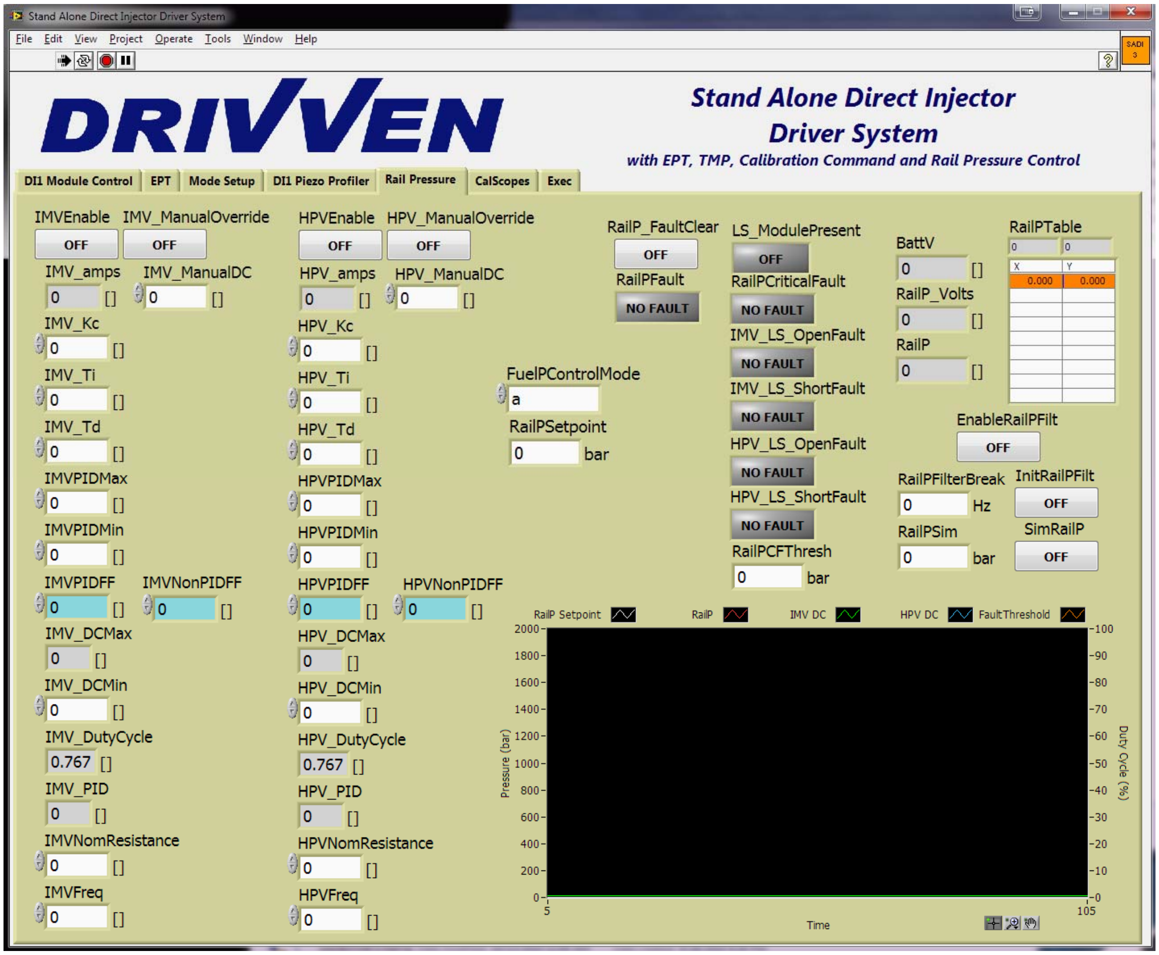
MissedPlus1Flag(Boolean): Set to TRUE upon CrankCount being incremented past the expected value for the location of a Plus1 tooth before the Plus1 tooth is detected. In other words, evenly spaced teeth are received at the location where a Plus1 tooth is expected. This condition causes a loss of sync. Re-sync is not allowed until the flag is cleared by setting SyncFlagClear to TRUE.

NoCamFlag(Boolean): Set to TRUE upon detecting 2 consecutive +1 teeth without detecting a TRUE CamSig (Stroke = TRUE). CamSig is only sampled at +1 tooth detection. This condition does not cause a loss of sync. NoCamFlag is cleared by setting SyncFlagClear to TRUE.

Encoder Only

MissedCamFlag(Boolean): Set to TRUE upon CrankCount being incremented past the expected value for the location of a CamSig rising edge before the CamSig rising edge is detected. In other words, the cam tooth was not received when expected. This condition causes a loss of sync. Re-sync is not allowed until the flag is cleared by setting SyncFlagClear to TRUE. In most encoder pattern scenarios, CamSig is actually the Z pulse of an ABZ encoder.

Rail Pressure Control Tab



IMVEnable OFF
IMV_ManualOverride OFF
IMV_amps 0 []
IMV_ManualDC 0 []
IMV_Kc 0 []
IMV_Ti 0 []
IMV_Td 0 []
IMVPIDMax 0 []
IMVPIDMin 0 []
IMVPIDFF 0 []
IMVNonPIDFF 0 []
IMV_DCMax 0 []
IMV_DCMin 0 []
IMV_DutyCycle 0.767 []
IMV_PID 0 []
IMVNomResistance 0 []
IMVFreq 0 []

HPVEnable OFF
HPV_ManualOverride OFF
HPV_amps 0 []
HPV_ManualDC 0 []
HPV_Kc 0 []
HPV_Ti 0 []
HPV_Td 0 []
HPVPIDMax 0 []
HPVPIDMin 0 []
HPVPIDFF 0 []
HPVNonPIDFF 0 []
HPV_DCMax 0 []
HPV_DCMin 0 []
HPV_DutyCycle 0.767 []
HPV_PID 0 []
HPVNomResistance 0 []
HPVFreq 0 []

RailP_FaultClear OFF
RailPFault NO FAULT
FuelPControlMode a
RailPSetpoint 0 bar

LS_ModulePresent OFF
RailPCriticalFault NO FAULT
IMV_LS_OpenFault NO FAULT
IMV_LS_ShortFault NO FAULT
HPV_LS_OpenFault NO FAULT
HPV_LS_ShortFault NO FAULT
RailPCFThresh 0 bar

BattV 0 []
RailP_Volts 0 []
RailP 0 []

RailPTable 0 0

X	Y
0.000	0.000

EnableRailPFilt OFF
RailPFilterBreak 0 Hz
InitRailPFilt OFF
RailPSim 0 bar
SimRailP OFF

Figure 19. Rail Pressure Tab User Interface

IMVEnable(Boolean): Enables or disables the Low-Side Driver channels responsible for controlling the IMV.

IMV_ManualOverride(Boolean): Bypasses the PID control loop and sets the Low-Side driver PWM output at a fixed Duty cycle defined by IMV_ManualDC.

IMV_amps(Amps): An estimate of the current flowing through the IMV based on the duty cycle, battery voltage and the nominal resistance of the valve. This value is used to calculate IMV_DCMax, which is the maximum duty cycle allowed through the IMV to protect the Low-Side driver.

IMV_Kc(-): Proportional gain for the IMV PID controller when FuelPControlMode is in IMV PID Mode. When FuelPControlMode is in HPV PID Mode this proportional gain is set to zero and the duty cycle is set only by IMVNonPIDFF.

IMV_Ti(%/loop time): Integral time-constant for the IMV PID controller.

IMV_Td(loop time): Derivative time-constant for the IMV PID controller.

IMVPIDMax(%): Maximum duty cycle allowed from the IMV PID controller output without feedforward.

IMVPIDMin(%): Minimum duty cycle allowed from the IMV PID controller output without feedforward.

IMVPIDFF(%): Feed forward duty cycle added to the PID controller duty cycle output when FuelPControlMode is in IMV PID Mode.

IMVNonPIDFF(%): Feed forward duty cycle added to the PID controller duty cycle output when FuelPControlMode is in HPV PID Mode.

IMV_DCMax(%): Calculated duty cycle that limits the Low-Side Driver from exceeding the current limitations on the IMV.

IMV_DCMin(%): Manual set-point that coerces the output of the PID controller to not drop below this value unless the IMV is disabled.

IMV_DutyCycle(%): Duty cycle that is written to the Low-Side driver regardless of what mode the controller is in. This duty cycle is only applied if IMVEnable is TRUE.

IMV_PID(%): Duty cycle that is output by the PID controller. This value is added to the feed forward value to calculate the IMV_DutyCycle. This is value shows you the influence of the PID controller on IMV_DutyCycle.

IMVNomResitance(Ohm): Nominal resistance of the IMV. This value is used to calculate the maximum duty cycle that can be used to control the IMV. A maximum duty cycle is calculated to prevent the current limit of the Low-Side Driver channel from being exceeded. It is important to measure this resistance correctly within +/-0.5 ohms.

IMVFreq(Hz): Frequency of PWM controlling IMV.

HPVEnable(Boolean): Enables or disables the Low-Side Driver channels responsible for controlling the HPV.

HPV_ManualOverride(Boolean): Bypasses the PID control loop and sets the Low-Side driver PWM output at a fixed Duty cycle defined by HPV_ManualDC.

HPV_amps(Amps): An estimate of the current flowing through the HPV based on the duty cycle, battery voltage and the nominal resistance of the valve. This value is used to calculate HPV_DCMax, which is the maximum duty cycle allowed through the HPV to protect the Low-Side driver.

HPV_Kc(-): Proportional gain for the HPV PID controller when FuelPControlMode is in HPV PID Mode. When FuelPControlMode is in HPV PID Mode this proportional gain is set to zero and the duty cycle is set only by HPVNonPIDFF.

HPV_Ti(%/loop time): Integral time-constant for the HPV PID controller.

HPV_Td(loop time): Derivative time-constant for the HPV PID controller.

HPVPIDMax(%): Maximum duty cycle allowed from the HPV PID controller output without feedforward.

HPVPIDMin(%): Minimum duty cycle allowed from the HPV PID controller output without feedforward.

HPVPIDFF(%): Feed forward duty cycle added to the PID controller duty cycle output when FuelPControlMode is in HPV PID Mode.

HPVNonPIDFF(%): Feed forward duty cycle added to the PID controller duty cycle output when FuelPControlMode is in IMV PID Mode.

HPV_DCMax(%): Calculated duty cycle that limits the Low-Side Driver from exceeding the current limitations on the HPV.

HPV_DCMin(%): Manual set-point that coerces the output of the PID controller to not drop below this value unless the HPV is disabled.

HPV_DutyCycle(%): Duty cycle that is written to the Low-Side driver regardless of what mode the controller is in. This duty cycle is only applied if HPVEnable is TRUE.

HPV_PID(%): Duty cycle that is output by the PID controller. This value is added to the feed forward value to calculate the HPV_DutyCycle. This is value shows you the influence of the PID controller on HPV_DutyCycle.

HPVNomResitance(Ohm): Nominal resistance of the HPV. This value is used to calculate the maximum duty cycle that can be used to control the HPV. A maximum duty cycle is calculated to prevent the current limit of the Low-Side Driver channel from being exceeded. It is important to measure this resistance correctly within +/-0.5 ohms.

HPVFreq(Hz): Frequency of PWM controlling HPV.

FuelPControlMode(IMV PID Mode| HPV PID Mode): Changes the rail pressure control method between PID controlling the IMV and keeping the HPV at a fixed value duty cycle(IMV PID Mode) and PID controlling the HPV and keeping the IMV at a fixed duty cycle(HPV PID Mode).

RailPSetpoint(bar): Desired rail pressure when rail pressure is PID controlled.

RailP_FaultClear(Boolean): Clears RailPFault once the fault has been resolved.

RailPFault(Boolean): Rail pressure fault, this error needs to be resolved and cleared before any Low-Side driver outputs will work.

LS_ModulePresent(Boolean): TRUE when the Low-Side Driver module is externally powered and detected by the controller.

RailPCriticalFault(Boolean): TRUE when the rail pressure has exceeded the RailPCFThresh.

IMV_LS_OpenFault(Boolean): TRUE when an open-circuit is detected on the channels of the Low-Side Driver that are controlling the IMV. See the Low-Side Driver manual for more information.

IMV_LS_ShortFault(Boolean): TRUE when a short circuit is detected on the channels of the Low-Side Driver that are controlling the IMV. See the Low-Side Driver manual for more information.

HPV_LS_OpenFault(Boolean): TRUE when an open-circuit is detected on the channels of the Low-Side Driver that are controlling the HPV. See the Low-Side Driver manual for more information.

HPV_LS_ShortFault(Boolean): TRUE when a short circuit is detected on the channels of the Low-Side Driver that are controlling the HPV. See the Low-Side Driver manual for more information.

RailPCFThresh(bar): An upper set-point that defines when the rail pressure has exceeded a critical level that could potentially damage hardware. If this threshold is exceeded, the low-side channels controlling the IMV and HPV turn off until the error has been cleared.

BattV(Volts): Battery voltage measured by the DI driver.

RailP_Volts(Volts): Voltage measured from the rail pressure sensor.

RailP(bar): Rail pressure in units of bar.

RailPTable(Volts|Bar): One-dimensional table that defines the conversion between rail pressure sensor voltage and units of pressure. X is the sensor voltage and Y is the units of pressure output.

EnableRailPFilt(Boolean): Enable filtering the incoming rail pressure sensor voltage.

RailPFilterBreak(Hz): Low cutoff frequency of a 2nd order filter on rail pressure sensor voltage.

InitRailPFilt(Boolean): Set to TRUE to reinitialize the 2nd order filter on rail pressure. The filter should be reset when changing RailPFilterBreak.

SimRailP(Boolean): When TRUE, simulates the rail pressure using the value in RailPSim. This is useful for tuning rail pressure PID control gains.

RailPSim(Boolean): The rail pressure that is simulated when SimRailP is TRUE.