DI Driver Module Kit User's Manual

D000020 Rev E

February 1, 2008

Drivven, Inc. • 12001 Network Blvd • Bldg E, 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.

Introduction

The Diesel Injector Driver Module Kit provides a CompactRIO (cRIO) module for driving typical common-rail, diesel solenoid injectors. The kit includes LabVIEW FPGA and RT VIs for controlling all driver channels. Each DI driver channel is individually controlled for timing and duration. This module is also capable of driving Siemens common rail diesel injectors with piezo actuators. The module cannot control piezo and solenoid actuator types simultaneously. When configured for piezo mode, channel 3 is not available and must have its terminals shorted together. The DI Driver Module RevE circuitry is available for stand-alone operation within a custom enclosure and can be operated with external 5V TTL commands. Please ask Drivven about this option.

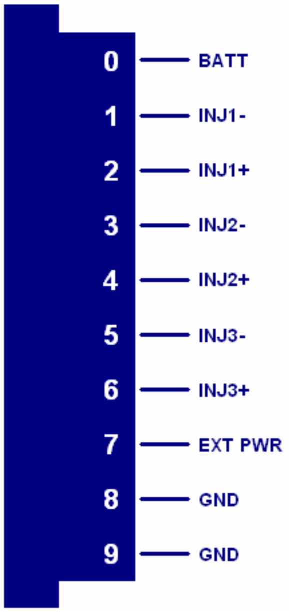
Features:

- 3-channel common rail, diesel solenoid injector drivers
- 2-channel Siemens common rail diesel piezo injector drivers (in piezo mode)
- Up to 150V internal boost power supply
- Up to 30A peak current / 10A hold current
- Operates from 6V to 32V battery
- Optional external input for high voltage supply (up to 150V)
  - Internal boost supply automatically shuts down when external high voltage is applied
- Circuit protection and diagnostics
  - Protected against INJ+/- short to battery / high voltage
  - Protected against INJ+/- short to GND
  - Internal power supply overload protection
  - Internal power supply over-charge protection
  - Module temperature protection
  - Open circuit detection
  - Fault flags reported for all above conditions
- Drivven's DI Calibrator application for calibrating injector current profile
- LabVIEW FPGA VI for engine-synchronous, multi-pulse injection control (up to 5 pulses). Interfaces directly with module.
- LabVIEW RT VI for run-time module initialization and calibration. Interfaces to FPGA VI.
- Available for stand-alone operation in custom enclosure – ask Drivven for details

Additional Siemens Common Rail Diesel Piezo Injector Features:

- 2-channel Siemens common rail diesel piezo injector drivers (in piezo mode)
- Array inputs to LabVIEW RT VI for valve open/close current profiles
- LabVIEW utility for tuning proper injector current profile

Pinout



Hardware

The DI Driver Module Kit provides three channels for driving typical common-rail diesel solenoid injectors in a National Instruments CompactRIO module. An internal boost power supply is included for providing up to 150V for driving peak solenoid currents up to 30A.

Powering the Module

The DI Driver module requires power from two different sources.

One source is from the CompactRIO backplane male high density D-Sub 15-pin (HD15) connector which mates with the module's female HD15 connector. This power source provides a regulated 5 volts and ground to various digital logic functions within the module. The CompactRIO 5V source is active whenever the CompactRIO or R-Series Expansion Chassis is properly powered. The module should only be powered at the HD15 connector by plugging it into a CompactRIO or R-Series Expansion Chassis. The module's HD15 connector should not be connected to any other device.

Another required power connection is at the external screw terminal connector. The terminals are labeled BATT (0) and GND (9). Typical power sources will be from automotive 12V or 24V battery systems. However, the module can accept power from a range of 6V to 32V.

The external battery power ground is completely isolated, within the module, from the CompactRIO 5V supply ground. However, the external battery ground and the CompactRIO ground may be connected externally.

The module will not be recognized by software without both power supplies active.

Note: Modules of revision A have a maximum BATT (0) input of 16V.

Warning: The external battery supply input terminals are not reverse voltage polarity protected. Such protection would compromise certain features of the module. Connecting power to the module in reverse polarity will certainly damage the module.

There is also a third optional power input to the external screw terminal connector. The terminal is labeled EXT PWR (7). EXT PWR (7) must have a reference externally to the same ground as GND (9). This power input can range from 6V to 150V and is used to drive the peak current portion of the injector current profile. This external power input may be useful if the peak current level and duration requirements, in combination with maximum engine speeds, exceed the load capabilities of the internal boost power supply. This topic is discussed in detail later in this manual. When a voltage above 6V is applied to the EXT PWR (7) terminal, then the internal power supply will automatically shut off, even if it is enabled via software. Even when EXT PWR (7) is connected to a power source, BATT (0) must still be connected to 6V to 32V in order for the module to be properly powered.

Platform Compatibility

CompactRIO modules from Drivven are compatible within two different platforms from National Instruments. One platform is CompactRIO, consisting of a CompactRIO controller and CompactRIO chassis as shown in Figure 1a below.

Figure 1a. CompactRIO platform compatible with Drivven CompactRIO modules.

The other platform is National Instruments PXI which consists of any National Instruments PXI chassis along with a PXI RT controller and PXI-78xxR R-Series FPGA card. An R-Series expansion chassis must be connected to the PXI FPGA card via a SHC68-68-RDIO cable. The CompactRIO modules insert into the R-Series expansion chassis. This platform is shown in Figure 1b below.

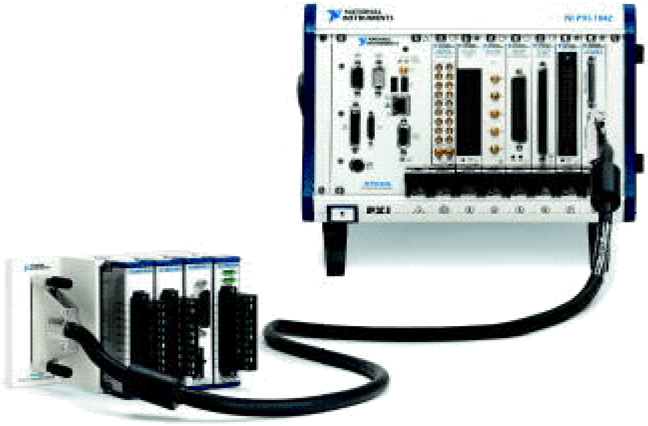


Figure 1b. PXI platform compatible with Drivven CompactRIO modules.

Drivven CompactRIO modules are not compatible with the National Instruments CompactDAQ chassis.

Drivven CompactRIO modules REQUIRE one of the hardware support systems described above in order to function. The modules may not be used by themselves and/or interfaced to third party devices at the backplane HD15 connector. These efforts will not be supported by Drivven or National Instruments.

Internal Boost Power Supply

The DI Driver Module contains an internal boost power supply which can be commanded to maintain a voltage level from battery voltage up to 150V. The boost power supply can be enabled or disabled at any time via software. It is disabled automatically whenever critical faults occur and can only be enabled thereafter by clearing the faults. The high voltage is stored within capacitors and used to drive the high-voltage, peak-current portion of the injector current profile.

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.

Internal Boost Power Supply Performance

There are six critical factors which determine how well the internal boost power supply will perform for a particular injector solenoid application. Those factors are:

- Injector solenoid resistance
- Injector solenoid inductance
- Peak current required to open the injector valve
- Working voltage requirement
- Frequency of injection events
- Injector solenoid back-boost

We will discuss each of these factors one at a time.

Injector Solenoid Resistance

Typical common rail diesel injector solenoids will have a resistance of 1 ohm or less. This resistance will waste, as heat, a portion of the energy supplied to the solenoid. It also affects the maximum current achieved in the coil, which may or may not be an issue, depending on the voltage applied. In general, lower solenoid resistances are better for common rail solenoid injector applications.

Injector Solenoid Inductance

Typical common rail injector solenoids will have an inductance of 1mH or less. Inductance acts to resist current change through the coil. The higher the inductance, the longer it will take to achieve the required peak current level. However, inductance is not a bad thing because it is proportional to the magnetic force generated for opening the injector valve.

Peak Current Requirement

Higher peak-currents require more energy from the power supply.

Working Voltage Requirement

A higher working voltage maintained by the power supply will be able to drive the peak current level in a shorter amount of time, thereby providing quicker valve opening times and more predictable fuel injection quantities over a given injection duration. However, boost power supplies operate less efficient at higher voltages. Therefore the power supply must work much harder to maintain a higher working voltage. Typical times to reach 20 amps in common-rail diesel injectors are on the order of 30 to 100 microseconds, with a working voltage of 100V.

Frequency of Injection Events

The frequency of injection events is directly proportional to the work required by the power supply to maintain the working voltage.

Injector Back-Boost

In many scenarios, it is possible to get a significant and useful back-boost charge from the injector solenoid when it is turned off. This back-boost from the solenoid mostly depends on the hold-current level during the injection event and the working voltage of the boost power supply. In most cases, if the current profile is correctly calibrated, there will be a small back-boost to the power supply which will reduce the work required to maintain the working voltage.

It is possible to incorrectly calibrate the module to use very little of the high voltage supply for driving the peak-current while setting a high hold-current level. This configuration can lead to back-boosting the power supply over the required working voltage. This can be observed during calibration and should be adjusted appropriately to prevent this from happening. If the working voltage exceeds 155V, then it will generate a fault and shut down automatically. Again, proper calibration of the current profile should not lead to back-boosting the supply higher than the working voltage.

Power Supply Faults and Protections

There are a few critical faults related to the operation of the internal power supply which will cause all operations of the module to shutdown automatically. The internal boost power supply and injection control can be re-enabled by manually clearing the faults via software.

Power Supply Charge Fault (PSCharge)

If the power supply is actively attempting to recharge and detects that the voltage is not rising, then the charge fault will be set. This fault would most likely occur if there were an internal problem with the module, such as a capacitor failure. This fault could also occur if the injector load was so great (peak-current, duration of peak-current, frequency of injection, etc.) that the power supply could not maintain the working voltage. In this latter scenario, the overload fault may be tripped before the charge fault, depending on the actual conditions.

Power Supply Overload Fault (PSOverLoad)

The DI driver module maintains an integrator of power supply usage. An internal counter increments with each power supply voltage boost and decrements according to a fixed time interval. If the integrator winds up to 60,000 counts, then an overload fault will be set. This is an indication that the module temperature would soon rise beyond its maximum operating temperature if not stopped. This integrator may be monitored within the DI Calibrator application. For light and moderate loads, the integrator will remain close to zero. The module temperature fault may be tripped before the overload fault, depending on the actual conditions.

Module Temperature Fault (ModuleTemperature)

Due to the CompactRIO module enclosure design, there is limited ability for heat to escape the module. The power supply circuitry was designed with components carefully selected for efficiency and compactness. Still, the primary source of heat within the DI Driver Module is the internal boost power supply. If the internal module temperature rises above approximately 85C, then the module temperature fault will be set.

High Voltage Limit Fault (HighVoltageLimit)

If the charge on the internal power supply exceeds 155V, then the high voltage limit fault will be set. This fault could occur due to internal problems with the module or due to excessive injector solenoid back-boosting.

The above faults are loosely tied together, in that certain conditions can lead to two or more of the above faults. For example, the power supply may be loaded such that the board temperature will reach its limit before the overload integrator limit is reached.

Internal Boost Power Supply Benchmarks

For a better understanding of what the power supply is capable of, in terms of driving typical common rail diesel injectors, below are some bench test results.

Table 1. Bench Test Results (Incomplete)

Test Condition	Test 1	Test 2	Test 3	Test 4	Test 5
External Battery Voltage (V)	13.8	13.8	13.8	13.8	13.8
HVTargetNonInject (V)	70	100	150	100	100
HVTargetInject (V)	70	100	150	100	100
PeakCurrent (A)	22	22	22	30	22
HVPeakTime (msec)	0.044	0.034	0.024	0.044	.034
HoldCurrent (A)	10	10	10	10	10
BackBoostTime (msec)	0.20	0.20	0.20	0.20	0.20
PeakTime (msec)	0.18	0.18	0.18	0.18	0.18
Number of injectors operated	3	3	3	3	3
Number of fuel pulses per cycle	1	1	1	1	5
Simulated engine speed (RPM)	6000	6000	6000	6000	2000
Length of test (min)					
Ambient temperature (C)	25	25	25	25	25
Final board temperature (C)					
Temperature stabilized?					
Injector under test	Bosch P/N 0445 110 072 Mercedes Benz P/N 611 070 09 87				

The internal power supply requires approximately 8 milliseconds to charge from 12V to 100V, and approximately 15 milliseconds to charge from 12V to 150V. This charge-up process will occur immediately when enabling the power supply. As long as the power supply is enabled, the requested working voltage will be maintained. Depending on the working voltage, the user may hear a slow frequency of faint clicks from the module. This is normal noise from the power supply.

The above table is a guide to the capability of the internal boost power supply. There are many different possible solenoid current profiles that are required by as many different injector solenoids. The DI Calibrator application can assist in determining whether the internal power supply will meet those requirements. Drivven will determine calibration parameters for customer’s injectors at no cost if the calibration data can be re-used. This does not include fuel flow measurements. Drivven must be provided with current profile information and a test solenoid or injector.

When the internal supply is disabled, the high voltage will bleed down to battery voltage from 150V in approximately 2 minutes.

While it is not ideal to use an external high voltage power supply, there is a screw terminal to the module for this purpose, in case the internal power supply cannot meet the demands of the application. However, before deciding to use an external power supply, consider using only two of the three available channels instead of all three. For example, consider using three DI driver modules for a six cylinder engine. This will lighten the load on each internal power supply.

Preventing Back-Boost Overcharging

There are actually two different working voltage set-points for the internal boost power supply. One set-point is called HVTargetNonInject while the other is called HVTargetInject. HVTargetNonInject is the high voltage set-point during which injection events ARE NOT in progress. HVTargetInject is the high voltage set-point during which injection events ARE in progress. For many applications these controls can be set to the same value, which will cause the power supply controller to maintain the same working voltage at all times. However, some injector calibrations may provide more solenoid back-boost to the power supply at the end of the injection event. If this back-boost occurs while the working voltage is already charged to its set-point, then eventual overcharging could result. Therefore, it may be necessary to set the HVTargetInject control to a value less than HVTargetNonInject, which will prevent the power supply from charging during the injection event, or limit the working voltage to a lesser value during injection. Power supply charging will then continue to maintain the higher working voltage after the solenoid back-boost if necessary. Determining the correct value for HVTargetInject may be an iterative process. If back-boost overcharging is occurring, then lower HVTargetInject until overcharging stops.

Connecting an External Power Supply

The external power supply connects to the same internal capacitance as the internal boost supply. Expect an exciting spark when connecting a live high voltage power supply to the module. It is strongly recommended that a good permanent connection be made before powering the external power supply. An external high voltage supply greater than 6V will be detected and cause the internal boost supply to automatically shutdown.

Recommended External High Voltage Supplies

Xantrex XFR150-8 or similar
Source: Test Equity for approximately $1500.00

Diesel Injector Drivers

Injector Driver Circuit Description

The DI Driver Module contains three common rail diesel solenoid injector drivers. The channels share some circuitry, making it impossible for injection events to overlap among channels within the same module.

There is a high voltage circuit and a battery voltage circuit which drives current through the injector solenoid. The high voltage circuit is only used at the beginning of the injector command to drive the first current peak. The battery voltage circuit is used thereafter. There is a current sensing circuit which is used for injector solenoid current control.

Injector Solenoid Current Profile Description

Figure 2 shows a typical injector solenoid current profile and the associated terms and time durations involved.

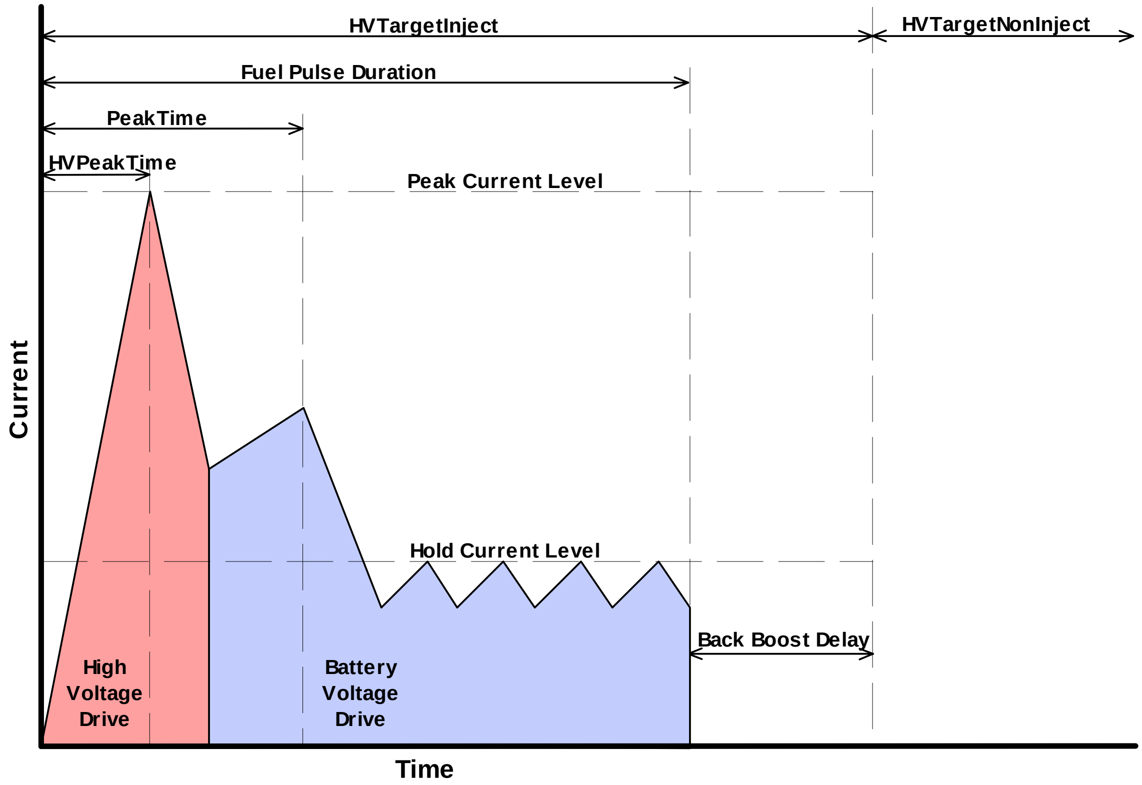


Figure 2. Injector solenoid current profile with labeled control parameters.

HVPeakTime and PeakCurrent

The first timed portion of the current profile is the high-voltage-peak-time (HVPeakTime). This is the time period during which high voltage is applied to the solenoid. This fixed-duration should be carefully calibrated to achieve the peak current required to consistently open the injector valve. Please note that achieving the first peak of current should be a timed event, based on HVPeakTime. The current is not sensed fast enough to provide precision control of the current during high voltage driving. However, the high voltage drive will turn off when either HVPeakTime expires OR PeakCurrent level is sensed. Therefore, if HVPeakTime were longer than necessary to achieve PeakCurrent, then the high voltage drive will eventually turn off, but only after severely overshooting the PeakCurrent level, depending on the inductance of the solenoid. It is best to calibrate HVPeakTime to achieve exact, consistent first peaks. Do not depend on current sensing to achieve a consistent first peak under high voltage drive.

PeakTime and PeakCurrent

The second timed portion of the profile, which includes the first portion, is the peak-time (PeakTime). This is the duration for which the PeakCurrent level is used as a sensed target current. When HVPeakTime expires, the PeakCurrent level remains as the target current, using the battery voltage instead of the high voltage supply. This is seen in the profile by the slower current rise rate after the first peak. Some applications may not require PeakTime to be any longer than HVPeakTime. However, some applications may require additional high current to be driven through the solenoid to achieve consistent valve opening. Also note that PeakTime may not necessarily need to be long enough for the PeakCurrent level to be achieved again by the battery voltage. PeakTime may only need to be long enough to achieve some intermediate current level (as shown).

As shown, when HVPeakTime expires, the current begins to fall. When the current is measured to be below the PeakCurrent threshold, then another current pulse is started 50 microseconds later (using battery voltage). Each battery-driven current pulse is driven in a similar manner to achieve peak-current and hold-current levels thereafter.

HoldCurrent

The third timed portion of the profile is the hold current portion. Hold current control is in effect for the remainder of the fuel pulse duration. Hold current is driven by battery voltage.

BackBoostTime

The fourth timed portion of the profile is the BackBoostTime. This time period directly follows the end of the injection duration and allows the back-emf of the injector solenoid to be directed to the internal power supply for back-boosting the high voltage. If BackBoostTime were set to zero, then injector current would be allowed to circulate until it dissipates, which would lead to very unpredictable injector valve closing. Therefore BackBoostTime should be calibrated long enough for the energy within the injector solenoid to be fully dumped to the power supply. If BackBoostTime is too short, you will see the injector current spike back up and circulate. BackBoostTime should be extended until the circulation cannot be seen on the scope trace. A typical BackBoostTime is approximately 0.2 milliseconds.

Injector Types Supported

Table 1 was derived by using typical common rail diesel injector solenoids having approximately 0.5 ohms resistance and 0.3mH inductance. However, many different types of injector solenoids can be driven with this module, including gasoline direct injectors. Drivven will help customers determine the appropriate calibration parameters for any type of solenoid at no cost if the calibration data can be re-used.

Injector Driver Circuit Faults and Protections

There a several scenarios which can lead to short circuits with the DI driver module. Each possible short condition is detected by the module and a critical fault is reported. Each short circuit fault will cause all power supply and injection control operations to shutdown automatically. The power supply and injection control can be re-enabled by manually clearing the faults via software.

Short Circuit Fault Conditions

INJ+ shorted to high voltage supply or battery: This condition will immediately cause a ShortCircuit critical fault. Since current is flowing through the injector solenoid, current rise times are limited by the solenoid itself and the short will typically be detected at about 30A.

INJ- shorted to high voltage supply or battery: This condition will immediately cause a ShortCircuit critical fault. Since current is bypassing the injector solenoid, current rise times are extremely fast and could peak as high as 200A before detection. However, the module can handle this current spike and shutdown appropriately.

INJ+ shorted to ground: This condition will cause a HighVoltageDriver critical fault or LowVoltageDriver critical fault during an injection event. The fault reported depends on when exactly the short condition occurs – during the high-voltage or low-voltage portion of the current profile.

INJ- shorted to ground: This condition appears to the module as an OpenCircuit condition, during an injection event, since the short is bypassing the internal current sensing of the module. The OpenCircuit condition will be detected and reported upon termination of the HVPeakTime and will shutdown the remainder of that injection pulse. Since current is flowing through the solenoid, and HVPeakTime is properly calibrated, then no damage would occur to the module or injector. The next pulse will be enabled and tested again for a similar condition. If this condition is not detected again, then the OpenCircuit non-critical fault will clear itself.

INJ+ shorted to INJ-: This condition will cause a HighVoltageDriver critical fault or LowVoltageDriver critical fault during an injection event. The fault reported depends on when exactly the short condition occurs – during the high-voltage or low-voltage portion of the current profile.

INJ+ / INJ- Open Circuit: This condition is detected when the current during the high voltage portion of an injection pulse does not exceed 2A. If this condition is detected, the remainder of the pulse is terminated in case this is actually an INJ- short to ground, which appears similarly to the module. The OpenCircuit non critical fault is reported for the appropriate channel and is automatically cleared upon the next pulse if the condition is removed.

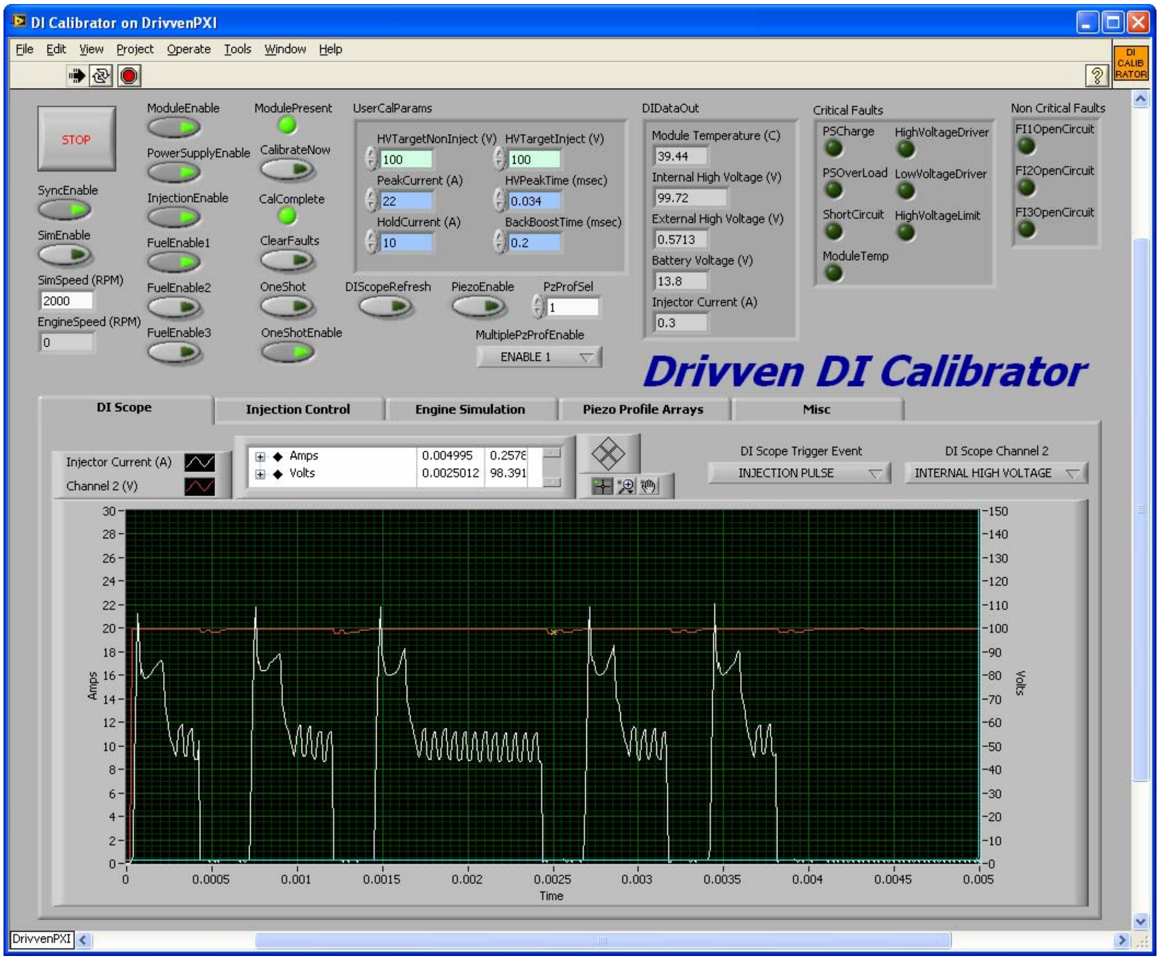
Injector Solenoid Current Profile Calibration Procedure

This section only discusses the procedure for calibrating injector solenoid control. The software portion of this manual will provide more details about particular control parameters which are not covered here. For example, this section does not provide the details of which cluster a particular parameter is contained within, or information about converting control parameters from engineering units to integer values submitted to the FPGA interface.

There are several manufacturers of common rail diesel solenoid injectors. Bosch, Denso, Siemens and Delphi are examples. Most injector solenoids require different current profiles to achieve consistent valve opening.

Included with the DI Driver Module Kit, is a LabVIEW calibration application called DI Calibrator. The overall application includes a LabVIEW FPGA project named DI_Calibrator.lvproj which includes a top level FPGA VI named DI_Calibrator.vi. The FPGA application was written for a PXI-7831R FPGA card, but can be easily bound to another National Instruments FPGA target. The FPGA DI_Calibrator.lvproj project also assigns slot 1 of a CompactRIO expansion chassis to a single DI Driver Module. Different CompactRIO I/O project configuration may be required if the FPGA target is changed.

Also included in the overall DI_Calibrator.lvproj project is an RT level VI named DI_RT_Calibrator.vi. This VI provides a convenient display interface to the FPGA application and must be used. The front panel is shown below in figure 3. The block diagram of DI_RT_Calibrator can also be used as an example of interfacing between the RT and FPGA levels when creating your engine control application.



SimSpeed (RPM) 2000

EngineSpeed (RPM) 0

UserCalParams

HVTargetNonInject (V) 100

HVTargetInject (V) 100

PeakCurrent (A) 22

HVPeakTime (msec) 0.034

HoldCurrent (A) 10

BackBoostTime (msec) 0.2

PzProfSel 1

MultiplePzProfEnable ENABLE 1

DIDataOut

Module Temperature (C) 39.44

Internal High Voltage (V) 99.72

External High Voltage (V) 0.5713

Battery Voltage (V) 13.8

Injector Current (A) 0.3

Amps	0.004995	0.2578
Volts	0.0025012	98.391

DI Scope Trigger Event INJECTION PULSE

DI Scope Channel 2 INTERNAL HIGH VOLTAGE

Figure 3. DI Calibrator Front Panel

This calibration procedure assumes that the user has already obtained an oscilloscope trace of the current and voltage profile of the injector to be used, or has the current and voltage specifications available. This procedure is not for the purpose of determining the correct calibration from scratch. It is only for the purpose of implementing a similar current and voltage profile as that of a pre-determined profile from the injector manufacturer. This tool could be useful in developing a proper current profile from scratch, but that procedure is beyond the scope of this document, as it would require a fuel flow bench.

Getting this application up and running is beyond the scope of this document, since these instructions are well covered in LabVIEW and LabVIEW FPGA documentation. However, the top level LabVIEW FPGA VI must be compiled for your FPGA target first, and this procedure is also covered in LabVIEW FPGA documentation.

Step 1: Connect the module. Once the DI Calibrator application is running, the DI Driver module should be inserted into slot 1 (or according to the DI Calibrator FPGA project slot assignment). One to three injectors can be connected to the correct INJ+/- terminals of channels one through three. We recommend only one injector be used until a proper current profile is calibrated. Battery voltage from a typical automotive battery in the range of 6V to 32V should be applied to the BATT (0) and GND (9) terminals of the module. Optionally, a high voltage source in the range of 12V to 150V can be applied to the EXT PWR (7) terminal. We recommend that all power connections be fused appropriately. Remember that different power sources and voltage levels can provide different injector performance results. Therefore, it is strongly recommended to use the same power sources for calibration as what will be used in the actual engine control setup.

Step 2: Begin communicating with the module. This is done by setting the ModuleEnable boolean to TRUE. If the external battery power is applied and the module is inserted in the correct slot, then the ModulePresent indicator will be set TRUE. This will trigger a calibration procedure. If this is the first time the module has been enabled since it was powered, you will notice that the critical fault indicators will be set briefly until a calibration is performed. The faults are cleared automatically by the DI CAL VI after the calibration is complete. When the calibration has been completed, then the CalComplete indicator will be set TRUE. When the module is enabled, you can also monitor the DIDataOut indicators for reasonable values. After you make appropriate calibration assignments as discussed below, you can press the CalibrateNow button to perform another calibration. This will cause the CalComplete indicator to be set FALSE until the calibration procedure is complete. The ModulePresent indicator will go FALSE if the module is removed from its slot. Now that we are communicating with the module, we can begin calibration. Don’t forget to press the Calibrate Now button when making a change to the UserCalParams! Otherwise your change will not be realized.

Step 3: Make calibration assignments. There are seven calibration values with which we need to be concerned. Each one will be described below in reference to the current profile shown in figure 4. The parameters contained within the UserCalParams cluster are parameters which are communicated once to the module upon setting the CalibrateNow boolean. The UserCalParams cluster contains six parameters which are discussed below. The PeakTime (msec) parameter, found on the Injection Control tab, is continuously communicated to the module.

HVTargetNonInject (V): This is the nominal working voltage target of the internal power supply and is maintained while injection events ARE NOT taking place. It can be in the range of 12V to 150V. This voltage is used to drive the first current peak through the injector solenoid. This value should be set to the peak voltage value obtained from the OEM injector operation. Higher values for HVTargetNonInject will achieve the first peak current point faster. However, maintaining higher working voltages is less efficient for the internal power supply. If an external power supply is used, then HVTargetNonInject can be set to zero.

HVTargetInject (V): This is the working voltage target of the internal power supply which is maintained while injection events ARE taking place. It can be in the range of 12V to 150V, but cannot be higher than HVTargetNonInject. This calibration parameter can be used to prevent internal power supply overcharging due to injector solenoid back-boosting. Please refer to the section titled **Preventing Back-Boost Overcharging** for further information. In general, this parameter can be initially set equal to HVTargetNonInject (V). Then, it can be reduced gradually, if necessary, to prevent overcharging. If an external power supply is used, then HVTargetInject can be set to zero.

PeakCurrent (A): This is the current level which will be targeted by the injector driver circuit during the PeakTime (msec) period. Remember that 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. PeakCurrent (A) is used as a target current level primarily during the battery voltage phase of the PeakTime (msec) period. This value should be set to the peak current level obtained from the OEM injector operation.

HoldCurrent (A): This is the current level which will be targeted by the injector driver circuit after the PeakTime (msec) period expires, until the end of the injection pulse. The HoldCurrent (A) is always driven using battery voltage. This value should be set to the hold current level obtained from the OEM injector operation.

HVPeakTime (msec): This is the time period for which the injector driver circuit will apply high voltage to the injector solenoid. However, this time period could be shortened automatically if the target PeakCurrent (A) is sensed first. Current sensing is not fast enough to provide consistent high-voltage current peaks. Therefore, HVPeakTime (msec) should be the primary governor of the first current peak. It is recommended that HVPeakTime (msec) be set initially to a very small value, and incremented in small steps until the desired peak current is achieved. As a guideline, a typical value for HVPeakTime (msec) is 0.030 msec to 0.100 msec. It is recommended that the initial value be set to 0.010 msec and incremented in 0.005 msec steps until the desired peak current is achieved. The actual resolution of HVPeakTime (msec) is 0.0032 msec. Battery voltage is used by the driver circuit after HVPeakTime (msec) expires.

BackBoostTime (msec): This is the time period after the injection pulse for which injector solenoid back-emf is directed to the internal power supply. This parameter can be set to zero initially. By doing so, you can see that current re-circulates through the solenoid for a while at the end of the injection pulse, until it dissipates to zero. This is not good for consistent fuel control, as the injector valve will remain open for some unpredictable amount of time until the current dissipates. The BackBoostTime (msec) can be incremented such that current will be seen to drop immediately to zero and then possibly spike back up at the end of the BackBoostTime (msec). The current will then re-circulate after the spike until it completely dissipates. This is also undesirable. BackBoostTime (msec) should be incremented just until the current spike is eliminated. Setting BackBoostTime (msec) to longer values than necessary can lead to a loss of the high voltage drive at the beginning of the next injection pulse. Typical BackBoostTime (msec) values are approximately 0.1 msec to 0.4 msec.

PeakTime (msec): This is the time period for which PeakCurrent (A) will be used as a current target level. This parameter should be greater than or equal to HVPeakTime (msec). After the first current peak is reached using high voltage, battery voltage will be used to drive PeakCurrent (A) until PeakTime (msec) expires. Often, the first current peak will be very short, using high voltage. Then the battery can be used to drive high currents to ensure a consistent injector valve opening. PeakTime (msec) does not necessarily need to be long enough for the actual PeakCurrent (A) level to be reached again by battery voltage. It could be set to a value such that current reaches some intermediate level by the time PeakTime (msec) expires. It is recommended to study the OEM injector operation in order to set this parameter to a value that best copies that behavior.

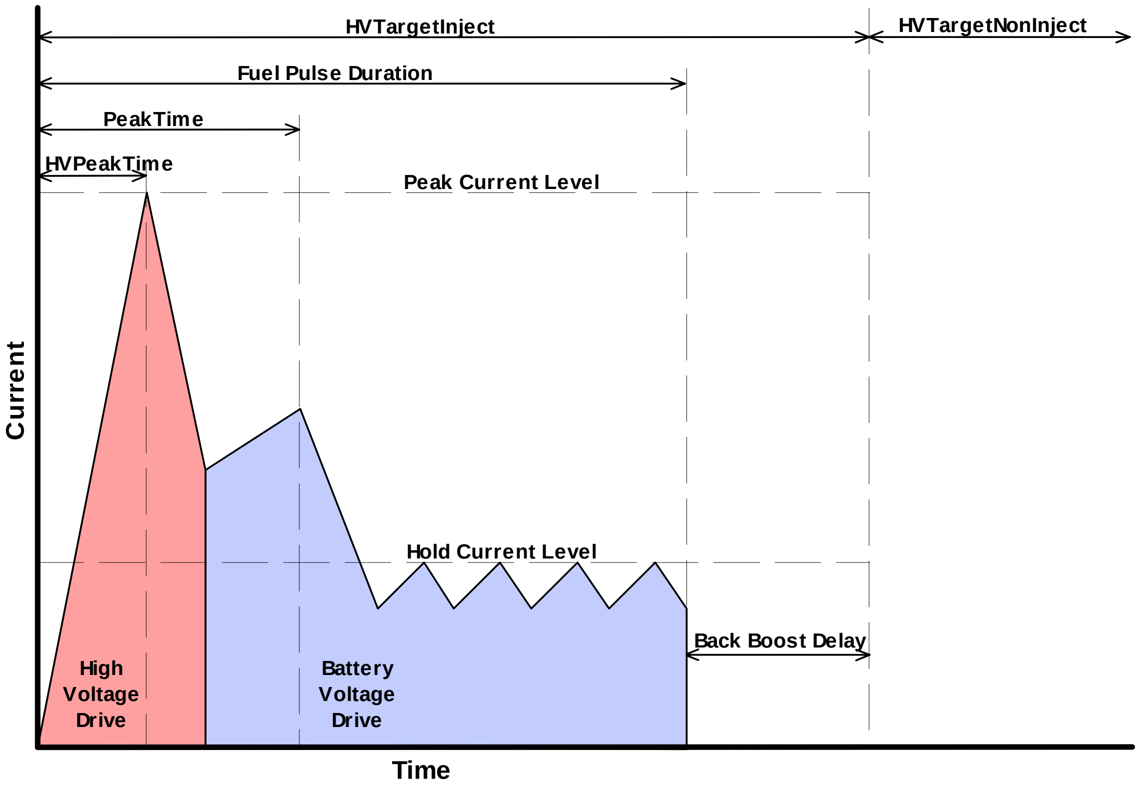


Figure 4. Injector solenoid current profile with labeled control parameters.

Step 4: Calibrating. After all seven calibration parameters are assigned, press the CalibrateNow boolean. When CalComplete returns to TRUE, you know that the calibration parameters have been sent to the module. There are actually many more calibration parameters that are transmitted to the module than the above seven. There are 75 additional parameters. They are not tunable however. One parameter is sent to the DI Module for each timed-loop iteration. Therefore the timed-loop period will affect the amount of time required for complete calibration. CalComplete will remain TRUE until another calibration is performed. You can also observe that when CalibrateNow is pressed again, CalComplete will go FALSE until calibration completes again.

Step 5: Test internal boost power supply. Set DI Scope Trigger Event to “POWER SUPPLY ENABLED.” Set DI Scope Channel 2 to “INTERNAL HIGH VOLTAGE.” This will cause the module to record internal high voltage on the red DIScope trace when the power supply is enabled. Set PowerSupplyEnable to TRUE and observe the High Voltage (V) indicator of the DIDataOut cluster. This should read very close to the HVTargetNonInject (V) parameter. Press the DIScopeRefresh boolean and observe the traces updated to the DIScope. You may need to use the plot zoom tools to zoom in and observe the voltage trace. When you are finished with the internal high voltage plot, set DI Scope Trigger Event to “INJECTION PULSE.”

Step 6: Prepare for first injector tests. It is important that the first injector tests be performed with one-shot injection pulses which are manually controlled instead of a series of multiple pulses. This is to prevent any damage to the injector in case calibrations are inappropriate. Injectors can typically withstand a single incorrect pulse as opposed to a series of incorrect pulses. After each single pulse, the current and voltage data should be observed on the DIScope to ensure that calibrations are correct. To generate one-shot fuel pulses manually, set the following:

SyncEnable set to FALSE (engine should not be simulating)
PowerSupplyEnable set to TRUE

InjectionEnable set to TRUE (enables injection control within the module)
OneShotEnable set to TRUE

Step 7: OneShot injection pulses and analysis. With each press of the OneShot boolean, a 1.0 msec injection pulse will be sent to channel one. You should hear an injector click with each pulse. After each injection pulse, press the DIScopeRefresh button to update the DIScope. When OneShotEnable is pressed, the DIScope X scale is set to a maximum of 1.5msec. Thereafter you can update the scale to another maximum value. When OneShotEnable is depressed, the DIScope X scale is set to a maximum of 200msec.

Analyze the injection pulse on the DIScope to see if the current profile matches your target profile.

A Note about DIScope current sampling. During normal operation, the injector current is sampled at 14usec intervals. However, during OneShot calibration operation, all other analog signals are temporarily disabled and only injector current is sampled at 7usec intervals. This allows the calibrator to see better resolution of the injector current waveform. At this improved resolution, the peak current should be observed to within 0.5 A. If repeated one-shots are performed while refreshing the scope trace, then the peak can be consistently observed.

If the scope is refreshed during normal engine simulation operation, then a slower current sampling will be observed and sometimes peaks will not be observed according to their actual peak. To confirm the actual peak level, use the OneShot mode.

Step 8: Engine simulation and continuous injection pulses. Set the SimSpeed (RPM) to some reasonable low value of engine speed to be simulated. It is recommended to initially use 500 RPM. Set the SimEnable and SyncEnable booleans to TRUE. Observe that the EngineSpeed (RPM) is identical to SimSpeed (RPM) and that CrankCount is changing. CrankCount is the simulated crankshaft pulse count. It should cycle from 0 to 15. Depending on the engine speed and the timed-loop period, CrankCount may be observed to stand still or change very slowly. Observe that CAD is changing. This is the simulated Crank Angle Degrees of the engine from 0 to 720 degrees. Typically the MissedCrankFlag and MissedCamFlag boolean indicators will never be set. However, if SimSpeed is changed dramatically by thousands of RPM, it is quite possible to get such an error. If this happens, just press the EPTErrorFlagClr boolean to clear the errors. The diesel engine is now under simulation and continuous injector tests can begin. To generate continuous injection events, set the following:

SimSpeed (RPM) set to 500 RPM
SimEnable set to TRUE
SyncEnable set to TRUE
PowerSupplyEnable set to TRUE
InjectionEnable set to TRUE (enables injection control within the module)
DISOI (DBTDC) set to 0 (0 degrees advance with respect to TDC for start of main injection pulse)
DICutoff (DBTDC) set to -120 (120 degrees after TDC for cutting off all pulse activity)

Set the remaining multi-pulse timing parameters as required according to the software section of this manual. From here, setting the FuelEnable1 boolean to TRUE will begin continuous injection events. Pressing the DIScopeRefresh button at any time will update the DIScope with the latest recorded current and voltage data. The plot zoom tools can be used as necessary to see the details of the traces. It is recommended not to perform continuous injection pulses until the required current profile is achieved by using OneShot injection events.

Adjustments may be made to the UserCalParams as necessary, but it is recommended that fuel pulses be shutdown during calibration and that the new UserCalParams be tested using OneShot injection pulses before proceeding to continuous injection pulses.

By operating continuously, you can observe how the internal boost power supply handles the load at the highest expected engine speed. Be careful not to operate the injector past the maximum expected operating conditions such that the solenoid overheats. The PS Over-Load Integrator Chart will continuously show whether the power supply is approaching an overload condition. If the integrator value reaches 60,000, then a PSOverLoad critical fault will be set and all module operations will be shutdown.

During continuous operation, injection events may pause briefly while the DIScope is being refreshed.

Step 9: Shutting down. A quick way to discharge the high voltage stored within the module is to operate the injector continuously while turning off the internal boost power supply (clear PowerSupplyEnable to FALSE). Observe the Internal High Voltage (V) indicator within the DIDataOut cluster decay quickly down to a safe voltage. Then the module and injector are safe to handle. Otherwise, you should give the module approximately 3 minutes for the high voltage to decay internally.

Step 10: Calibration transfer. Take note of the seven calibration parameters and use them for calibrating your DI Driver Modules at the startup of your engine control application. Refer to the Software section of this manual for information about implementing the DI Driver Module Kit VIs within your engine control application.

Piezo Injector Mode

The documentation so far has been related to controlling solenoid injectors. This section will describe how to use the module and software to drive Siemens common rail diesel piezo injectors or other similar piezo injectors. Most of the terminology and features described above still apply. However, some functionality will change during piezo mode, and some new features and terminology are introduced. Below are lists of the parameters that remain the same, are no longer applicable, and new for piezo injector control.

Control parameters which have identical function between solenoid and piezo operation:
ModuleEnable
PowerSupplyEnable
InjectionEnable
Fuel Enable1
Fuel Enable2
CalibrateNow
ClearFaults
OneShotEnable
OneShot
HVTargetNonInject
HVTargetInject
OffTime

Parameters which are not applicable for piezo operation:
FuelEnable3
PeakCurrent
HVPeakTime
HoldCurrent
BackBoostTime
PeakTime

Additional parameters which must be configured for piezo operation:
PiezoEnable – Switches module operation from soleoid to piezo operation
MultiplePzProfEnable – Enables one, two or three charging profiles. For Siemens piezo injector, set to "ENABLE 1."
PzProfSel – Selects between PzProfOn1, PzProfOn2 or PzProfOn3. For Siemens piezo injector, set to a value of 1.
PzProfOn1 – Duty cycle profile for charging the piezo stack
PzProfOn2 – Duty cycle profile for charging the piezo stack
PzProfOn3 – Duty cycle profile for charging the piezo stack
PzProfOff – Duty cycle profile for discharging the piezo stack

Control Strategy

The control strategy for the piezo injector is a fast PWM strategy for charging and discharging the piezo stack. Electrically, a piezo stack is similar to a capacitor. When a capacitor is in the discharged state, and a voltage is applied, it is similar to a short circuit. As the capacitor is charged to the supply voltage, it becomes an open circuit. The goal is to charge the piezo stack at a controlled rate in order to cause it to expand and open the valve. It is also necessary to discharge the stack at a controlled rate to contract and close the valve. The stack must be charged and discharged at a controlled rate so that the stack is not damaged from too rapid expansion and contraction, or excessive current levels.

Operating the Siemens Piezo Injector

The Siemens piezo injectors require a maximum charge of 120V to 140V. The rate of charge or discharge should be controlled so that the peak current (absolute value) never exceeds 10A. The total time for charging or discharging the stack should not be less than 180 usec. At the writing of this documentation, the DI Driver module has only been tested with a Siemens piezo injector. However, other similar injectors from other manufacturers may be used, as long as they do not have higher voltage requirements than 150V.

Charge and Discharge Profiles

A profile of PWM duty cycles must be created for charging and discharging the stack at a controlled rate. The fixed PWM period is 2 usec, and each element of the profile array represents a 2 usec step in time. The array is fixed at 128 elements long for a total profile time of 256 usec. The resolution of duty cycle values is 5%, or 0.1 usec. There are three "On" arrays to select from, which can be used for specifying the opening charge rate. There is a single "Off" array for specifying the closing discharge rate. Multiple opening profiles were originally implemented to provide different injection rates during run time. However, the Siemens piezo injector is not designed to allow controlled injection rates. Therefore, only one opening profile is useful. Therefore, we recommend only using PzProfOn1 with PzProfOff, and setting MultiplePzProfEnable to "ENABLE 1," and setting PzProfSel to a value of 1. Figure 5 below shows an example pulse train created for charging and discharging a piezo stack with the DI Driver module.

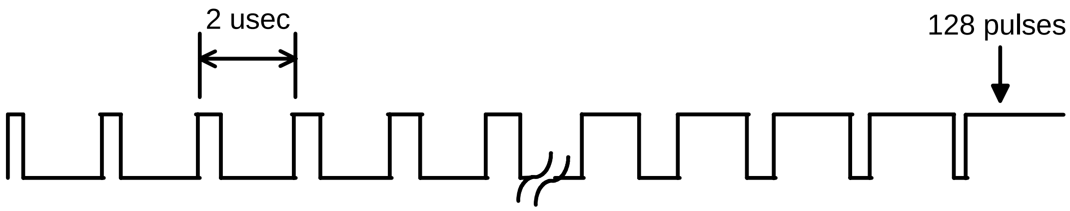


Figure 5. Example of charging/discharging duty cycle profile. Duty cycles start small (0%) and increase to 100%.

In-Series Inductance

In order to achieve smoother current profiles, an inductor is required in series with one of the wires to the piezo injector. It has been found that an inductor of approximately 85 uH and a current rating of at least 5A is appropriate. However, experimentation may show that different inductor values provide optimal current profile control and smoothing. A very high inductance value could be used such that PWM operation would not be necessary. However, it is difficult to find available high inductance parts with high current ratings. Also, it is easier to modify the charge and discharge profiles than to select the optimal series inductor. The series inductor is only for the purpose of providing smoother current control with the fast PWM operation. Inductors which are wound on a ferrite rod with large wire may provide the most mechanical strength. These can be easily soldered in series with the injector wire and wrapped with heat-shrink tubing for insulation. The inductor shown in figure 6 is an example.

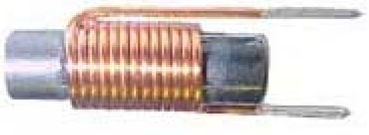


Figure 6. Example inductor wound on ferrite rod using large wire.

Discharging with Channel 3
Channel 3 is used to disharge the piezo stacks of channel 1 and channel 2 by shorting across the + and – terminals of channel 3. This means that the DI Driver module can only operate two channels of piezo injectors.

Figure 7 below shows the proper connection of piezo injectors to the DI Driver module.

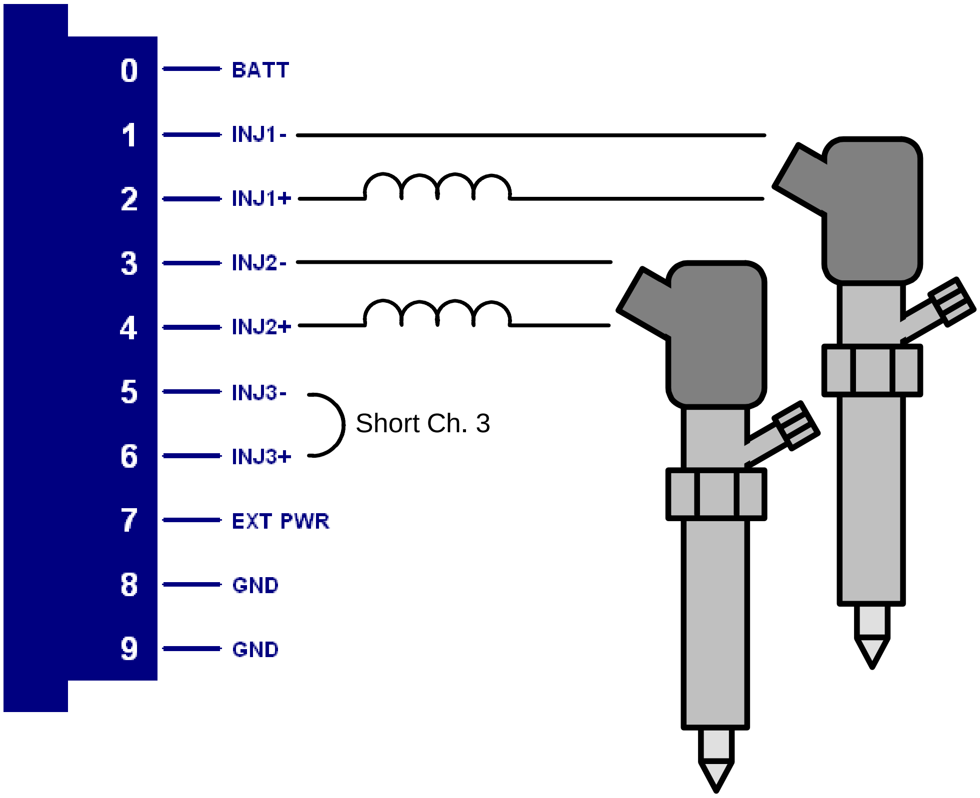


Figure 7. Connecting piezo injectors to the DI Driver module. In-series inductance is required and channel 3 must be shorted.

Calibration Procedure
In order to drive the piezo injector using the DI Calibrator, follow the above 10-step procedure within the section titled “Injector Solenoid Current Profile Calibration Procedure.” However, note the following differences:
Step 3: Make calibration assignments. HVTargetInject and HVTargetNonInject should be set to 120-140V (for the Siemens piezo injector). All other parameters discussed within section 3 are ineffective and may be ignored for piezo operation.
Step 6: Prepare for first injector tests. The DIScope will still function, but will not show good resolution data. External current and voltage probes should be used with an oscilloscope for proper verification. To generate one-shot fuel pulses manually, set the following:

PiezoEnable set to TRUE
MultiplePzProfEnable set to ENABLE 1
PzProfSel set to 1
PzProfOn1 set to experimental profile (using PiezoProfiler described below)
SyncEnable set to FALSE (engine should not be simulating)
PowerSupplyEnable set to TRUE
InjectionEnable set to TRUE (enables injection control within the module)
OneShotEnable set to TRUE

Step 7: OneShot injection pulses and analysis.
Analyze the injection pulse using external current and voltage probes with an oscilloscope to verify that the current profile matches your target profile.

Creating duty cycle profiles using PiezoProfiler.vi
A VI was created to help with creating and modifying piezo charging and discharging duty cycle profiles. There are four identical sections of the VI front panel. The first three sections are for modifying PzProfOn1, PzProfOn2 and PzProfOn3. The last section is for modifying PzProfOff. For driving Siemens piezo injectors, it is only necessary to create a single ON profile. Therefore the first and last sections are the only sections which need to be used.

NOTE: The PiezoProfiler requires the RT target to have VI Server enabled. Consult the LabVIEW documentation for enabling VI Server on a particular target.

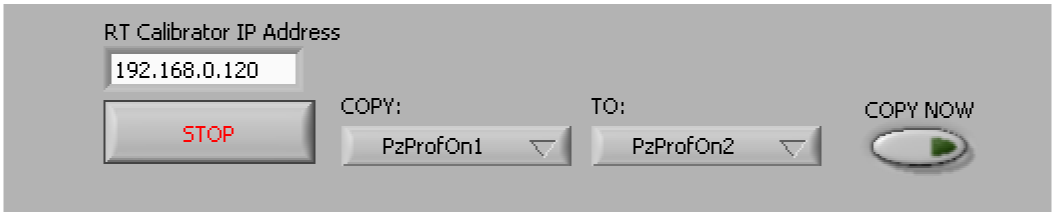

RT Calibrator IP Address
192.168.0.120
COPY: PzProfOn1
TO: PzProfOn2

Figure 8. IP Address configuration, Stop button, and profile copy controls.

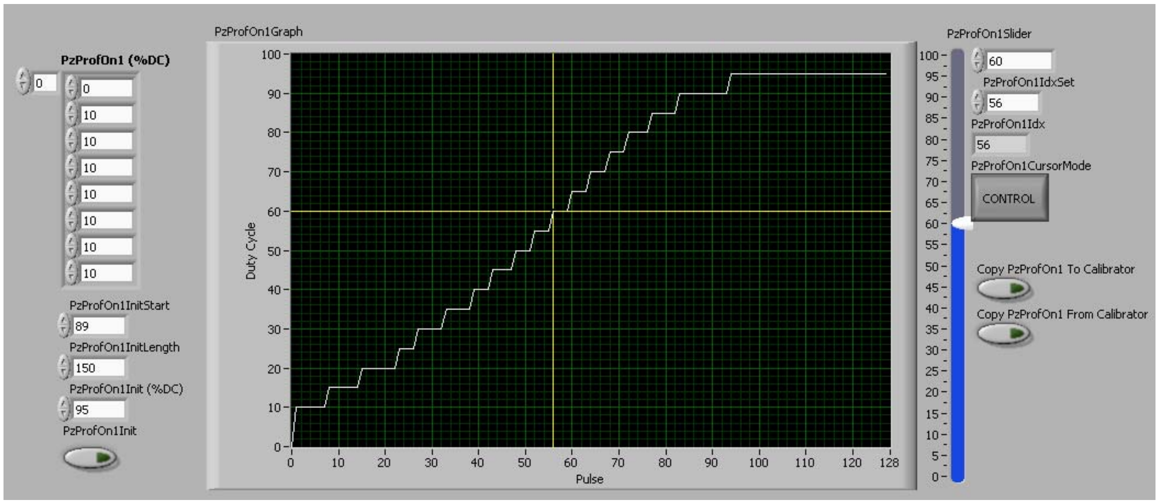

PzProfOn1 (%DC)
0
0
10
10
10
10
10
10
10
PzProfOn1InitStart
89
PzProfOn1InitLength
150
PzProfOn1Init (%DC)
95
PzProfOn1Slider
60
PzProfOn1IdxSet
56
PzProfOn1Idx
56
PzProfOn1CursorMode
CONTROL

Figure 9. Top section of PiezoProfiler.vi wfor modifying the PzProfOn1 profile array.

Before starting the PiezoProfiler, make sure that the DI Calibrator application is running on your RT target and that the IP address control of the PiezoProfiler is set to the IP address of your RT target. Then you may start the PiezoProfiler.

NOTE: The IP address cannot be changed while the PiezoProfiler is running. It must be set before running.

Figure 8 shows the local copy controls for copying profiles from one section to the other within the PiezoProfiler.

On the left of figure 9, a portion of the PzProfOn1 array is shown. This array is always updated to the data being displayed within the PzProfOn1Graph. Underneath the array are controls to allow the user to initialize portions of the profile with a constant value. The user may enter the start pulse (PzProfOn1InitStart) and the number of pulses (PzProfOn1InitLength) to initialize with the value in PzProfOn1Init. When the PzProfOn1Init button is pressed, the specified portion of the profile will be initialized.

On the right of figure 9, the button PzProfOn1CursorMode changes the way the user interacts with the PzProfOn1Graph.

When the button is in CONTROL mode, then the user may move the cursor to a different pulse location by changing the PzProfOn1IdxSet. The yellow cursor will move to the pulse location commanded. Then the user may change the value of PzProfOn1Slider to modify the profile duty cycle at that pulse location.

When the button is in DRAG mode, then the user may use the mouse to drag the cursor to a new pulse location. Then the user may change the value of PzProfOn1Slider to modify the profile duty cycle at that pulse location.

When the user is satisfied with the profile, it may be copied to the identical array within the DI Calibrator application using the “Copy PzProfOn1 To Calibrator” button. This function is implemented using VI Server. The RT target must have VI Server enabled in order for this copy function to work. Also, the “RT Calibrator IP Address” control must be configured for the correct RT target address. Also, the DI Calibrator must be running on the RT target.

After copying the profile to the DI Calibrator on the RT target, the user can verify that the profile array on the DI Calibrator has been updated. Then the CalibrateNow button should be pressed to update the module with the new profile.

Another button named “Copy PzProfOn1 From Calibrator” allows the user to copy a profile from the DI Calibrator on the RT target to the PiezoProfiler.

As you are working with various profiles, be sure to stop the PiezoProfiler occasionally to save the current values as default. This will prevent you from accidentally losing your work. The user may also make modifications to the PiezoProfiler program in order to facilitate working with various profiles.

Achieving a smooth profile requires several iterations of this process. It is recommended to start with 0% duty cycle for the entire profile. Then increase the duty cycle gradually at the lower pulse counts for each testing iteration. When the desired current increase is achieved, then move on to a higher pulse number. This method will continue until 90% - 100% duty cycle is reached at the higher pulse counts.

When the desired current and voltage waveforms are achieved, then the user may copy the array to the engine control software to wire into the correct profile terminal of the di_rt_cal_revx.vi.

Figure 10-12 below shows example diagrams of voltage and current waveforms which are a guidline for calibrating piezo injectors. Figure 13-15 below shows voltage and current waveforms created on a Siemens piezo injector. The working voltage was set to 140V. The duty cycle profile used to create these waveforms are saved as the default data in the PiezoProfiler.

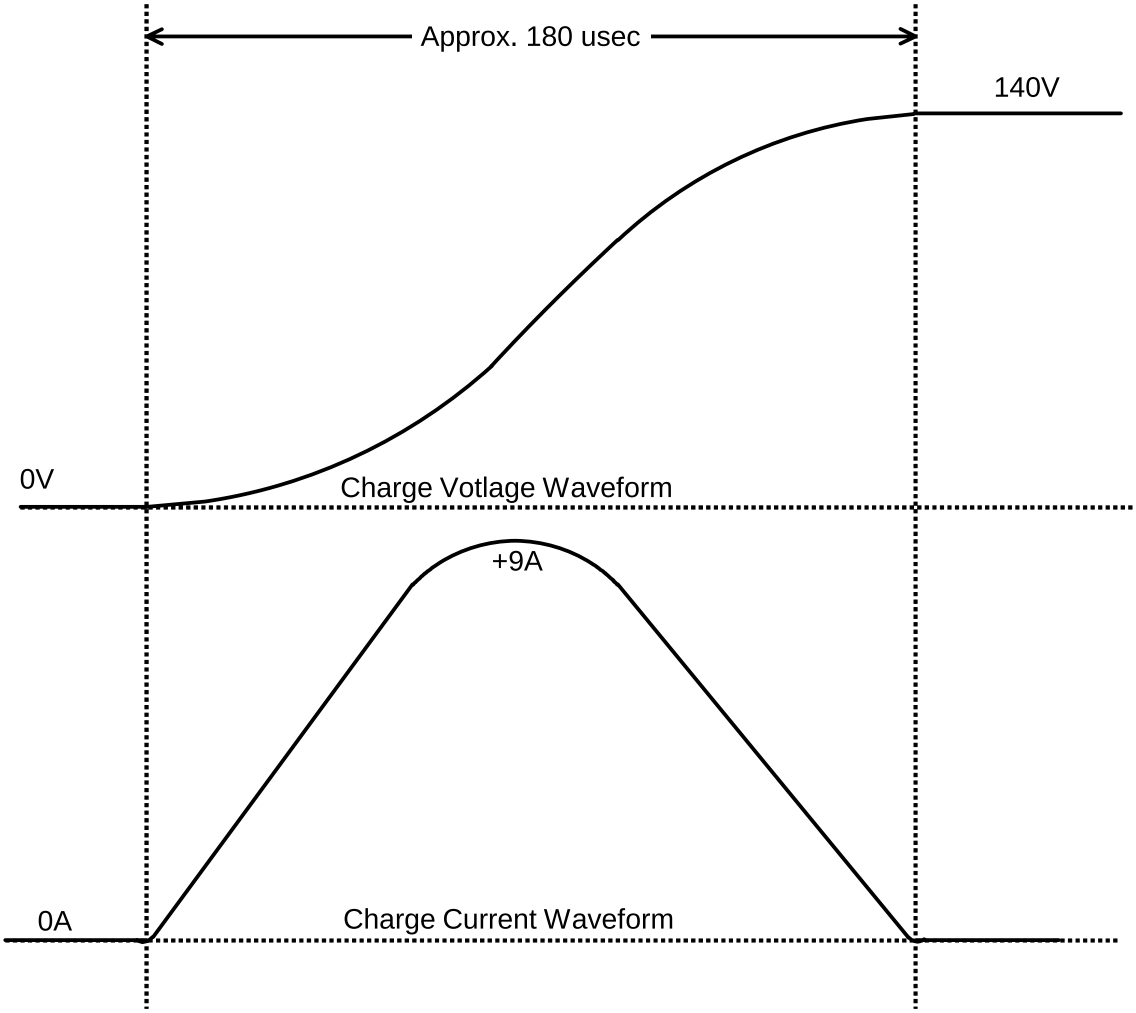


Figure 10. Example voltage and current charge waveforms for guideline.

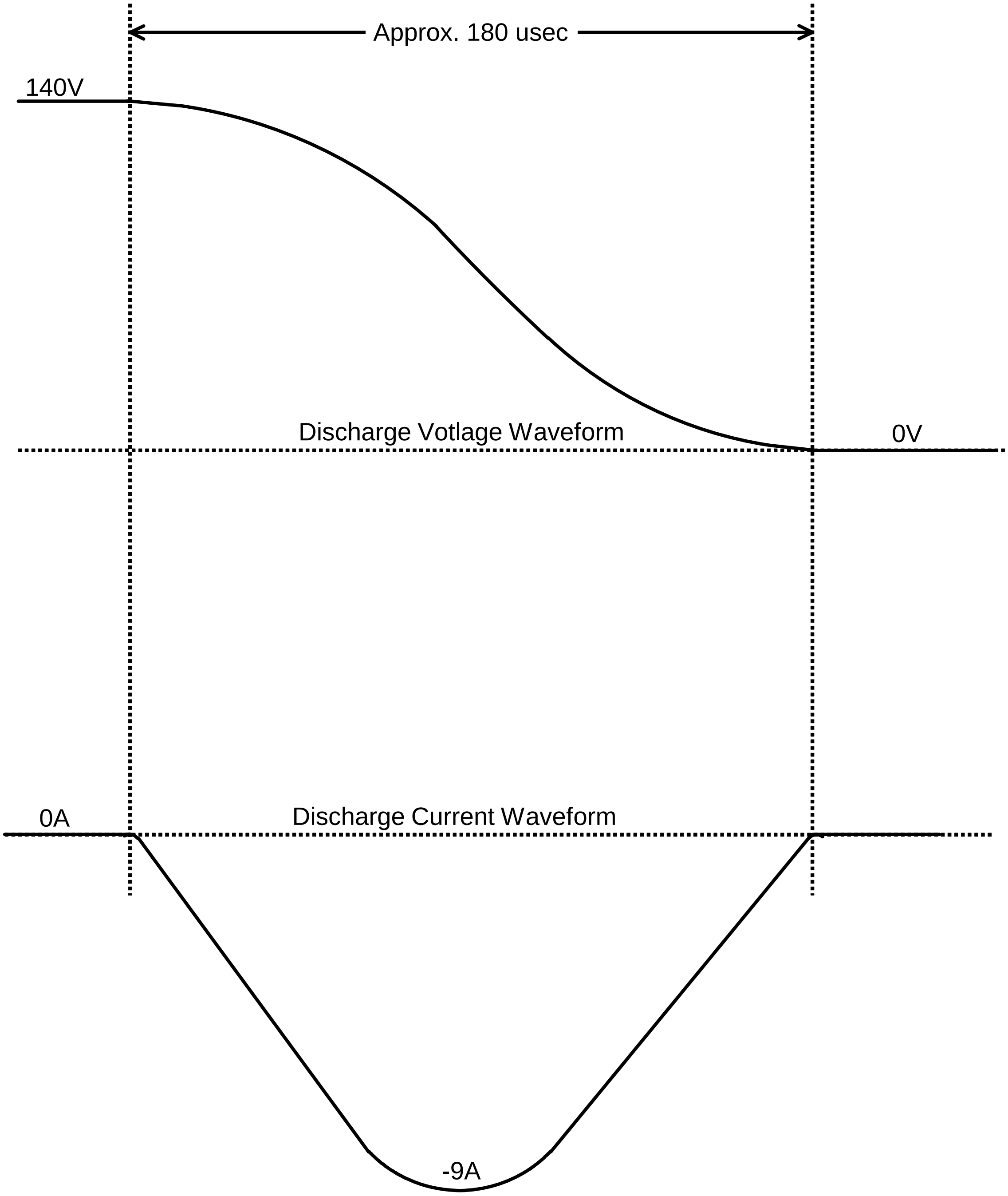


Figure 11. Example voltage and current discharge waveforms for guideline.

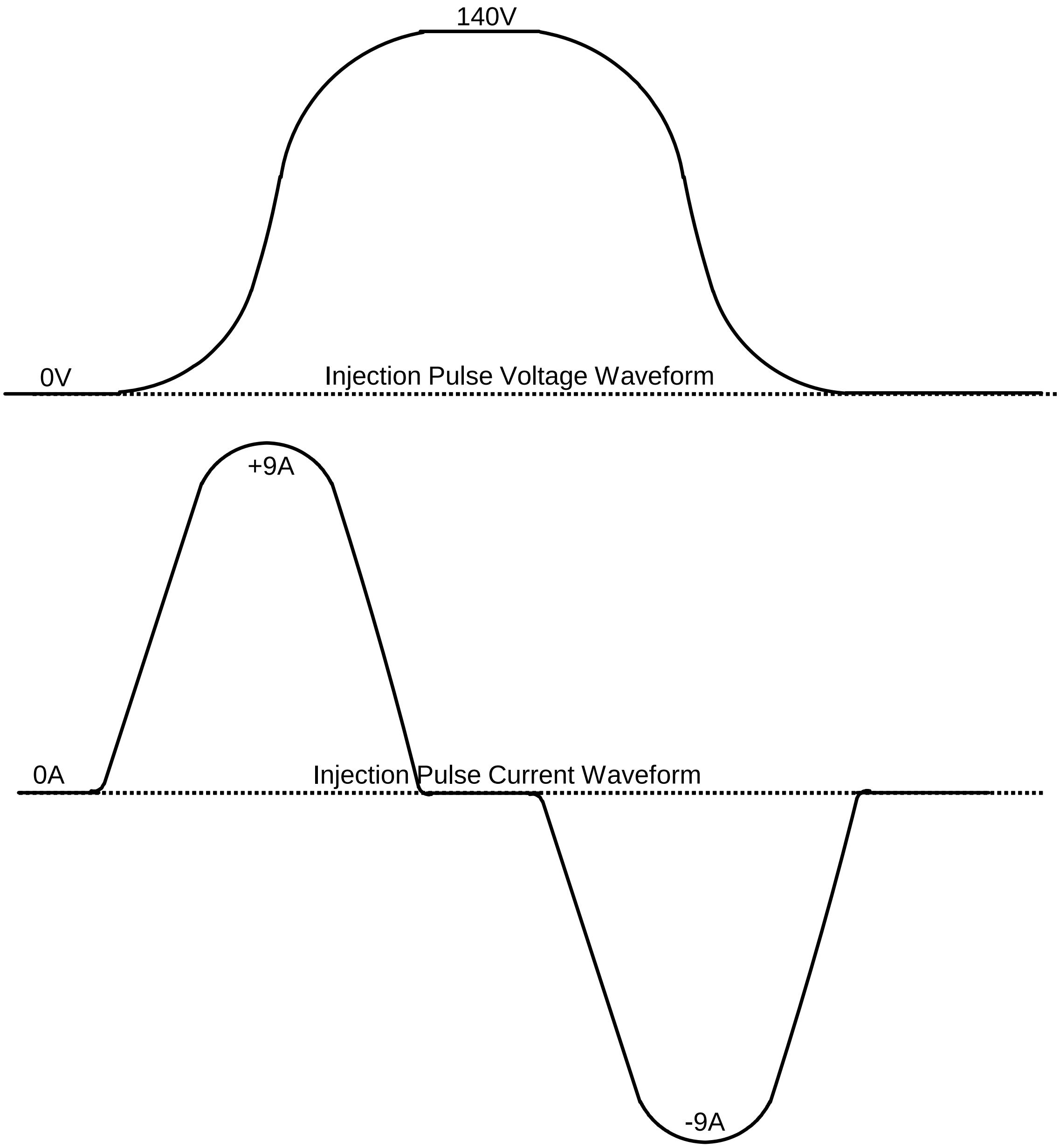


Figure 12. Example voltage and current injection pulse waveforms for guideline.

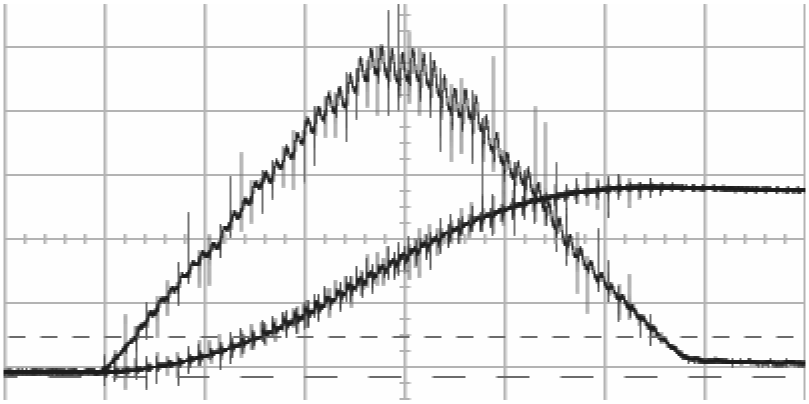


Figure 13. Actual voltage and current charge waveform created using Siemens piezo injector.

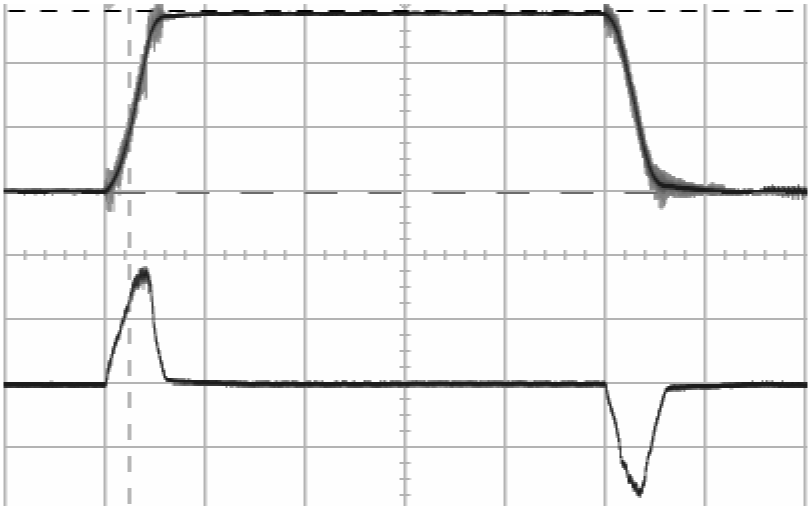


Figure 14. Actual voltage and current injection pulse waveform created using Siemens piezo injector.

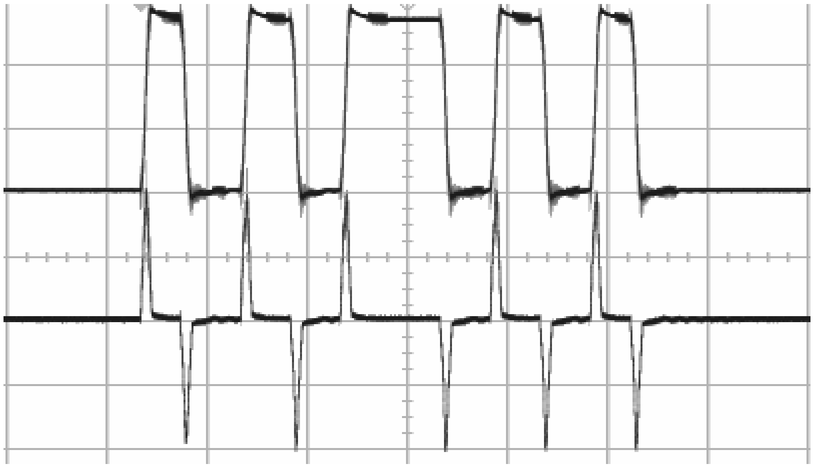


Figure 15. Actual voltage and current multi-pulse injection waveform created using Siemens piezo injector.

Software

The DI Injector Driver Module Kit is provided with three LabVIEW driver VIs, comprised of one LabVIEW FPGA VI and two LabVIEW RT VIs. The LabVIEW FPGA VI (di_vt_revx.vi) is for interfacing directly with the module and for providing a control interface to the LabVIEW RT level. The LabVIEW RT DI CAL VI (di_vt_rt_cal_revx.vi) is for initializing and calibrating the module while handling communications between the RT and FPGA levels. The LabVIEW RT DI DATA CONVERT VI (di_vt_rt_data_convert_revx.vi) is for converting data from the FPGA VI to engineering units. Figures 16-18 show the icons which represents these VIs. Additional optional support VIs provided by drivven may also be used at the RT level and are discussed later in this software section.

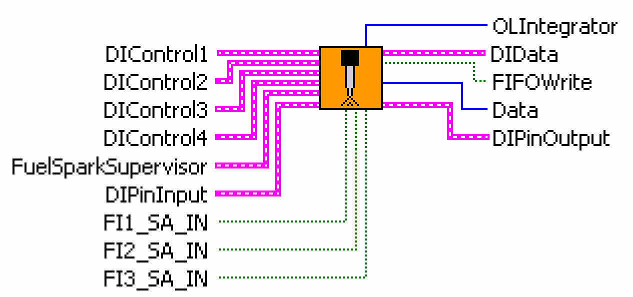


Figure 16. di_vt_revx.vi icon with leads.

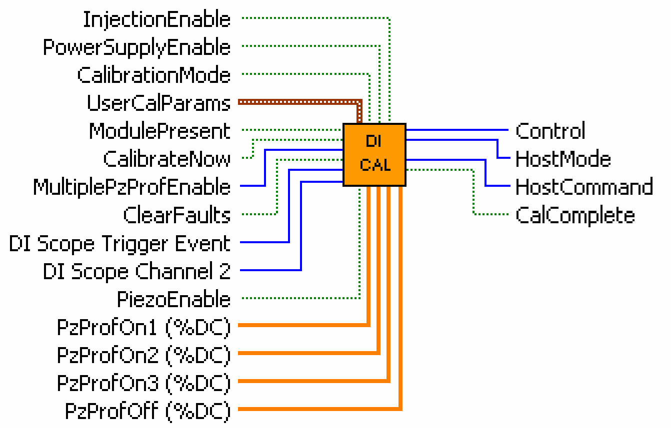


Figure 17. di_vt_rt_cal_revx.vi icon with leads.

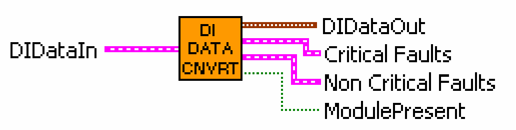


Figure 18. di_vt_rt_data_convert_revx.vi icon with leads.

VERY IMPORTANT NOTES:

The FPGA VI requires:

- LabVIEW 8.2 Full Development or later
- LabVIEW RT Module 8.2 or later
- LabVIEW FPGA Module 8.2 or later
- NI-RIO 2.1 or later

The FPGA VI must be placed within a Single Cycle Loop (SCL) of a LabVIEW FPGA block diagram. The SCL must execute at the default clock rate of 40 MHz.

The FPGA VI requires a pre-synthesized netlist file having a matching name and an extension of .ngc. The netlist file must be located in the same directory as the matching VI.

The FPGA VI requires the installation of a special CompactRIO module support package called cRIO-generic. Please follow the steps below to install the cRIO-generic package:

1. Confirm that LabVIEW is closed.
2. Add the line **`cRIO_FavoriteBrand=generic`** to the LabVIEW INI file. The LabVIEW INI file is typically found at C:\Program Files\National Instruments\LabVIEW 8.0\LabVIEW.ini.
3. Upon restarting LabVIEW, the cRIO-generic module will appear in the list of available modules within the LabVIEW FPGA “New C Series Module” configuration dialog. All Drivven CompactRIO modules require adding an associated cRIO-generic module to your LabVIEW Project. Within the Project Explorer, A cRIO-generic module can be added to a PXI FPGA expansion chassis or a CompactRIO chassis. This is best understood by observing an example project provided with your module kit.

WARNING!

When writing values to an FPGA cluster from the RT level, every parameter within the cluster must be explicitly written. If any parameter is not explicitly written, then the default value for that particular data type will be used. This could cause unexpected behavior.

Brief Glossary of Terms

CAD: Crank Angle Degrees. 360 CAD per two stroke cycle or one crankshaft rotation. 720 CAD per 4-stroke cycle, or two crankshaft rotations.

CAT: Crank Angle Ticks. Unit of angular measure reported by the CurrentPosition output of the EPT VI. Reported as a specified power-of-two angular ticks per crank tooth. 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 incremented by 128 CAT from one tooth to the next. If a 60-2 pattern were used, the total 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 cycle would be 120*128=15360.

MAX_CAT: Maximum Crank Angle Ticks per engine cycle.

The FPGA VI supplied with this kit cannot generate fuel commands without the supervision of an engine position tracking (EPT) VI from Drivven. The EPT VI provides the necessary output cluster to be wired to the FuelSparkSupervisor input cluster.

FPGA VI Implementation

The FPGA VI must be contained within a single cycle loop and clocked at 40 MHz. The PinInput and PinOutput clusters are wired to LabVIEW FPGA I/O pins which are configured for a cRIO controller chassis or a cRIO R-Series expansion chassis. Refer to the LabVIEW FPGA documentation for details about configuring cRIO I/O pins.

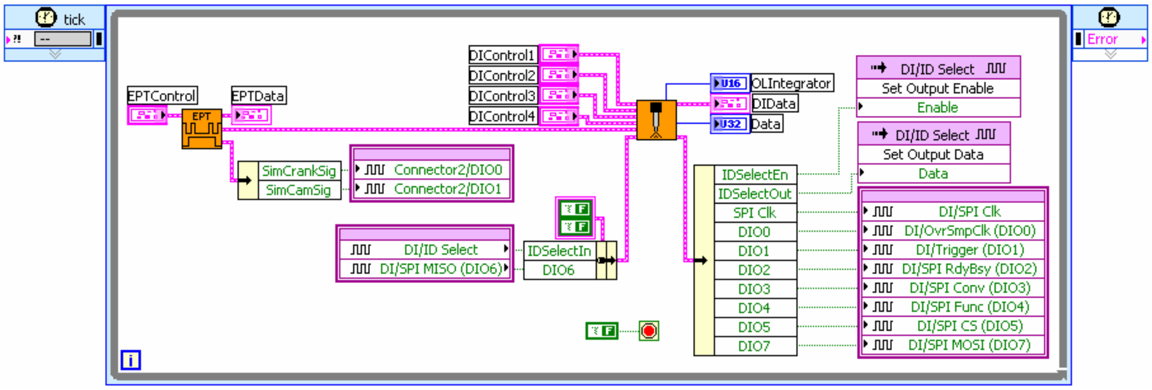


Figure 19. Example FPGA block diagram implementation of di_vt_revx.vi.

DIPinInput (Cluster)
These boolean controls must be connected to their corresponding FPGA I/O Node input item.

DIPinOutput (Cluster)
The boolean indicator named IDSelectEn must be connected to a Set Output Enable method of an FPGA I/O Method Node. The boolean indicator named IDSelectOut must be connected to a Set Output Data method of an FPGA I/O Method Node. The remaining boolean indicators must be connected to their corresponding FPGA I/O Node output item.

WARNING!
Great care must be taken to ensure that LabVIEW FPGA I/O node output items are only wired from a single logic source. There is no circumstance in which FPGA I/O node output items should be driven by multiple sources when interfacing to cRIO modules, otherwise strange behavior or module damage could result. Two LabVIEW FPGA code snippets are shown below which illustrate this issue. Figure 20a shows the correct implementation of FPGA I/O node blocks, whereas a group of three outputs to an ADCombo module are controlled while another group of eight outputs to a Spark module are controlled. Each of the output items are selected only once in the entire block diagram. On the other hand, figure 20b shows a coding mistake that should be avoided. Notice the group of ADCombo output items where a Spark module output item is selected instead of the correct ADCombo module output item. The same Spark module output item is also selected in the Spark group below. This means that the Spark (DIO5) output is being driven by two different logic sources and will cause strange behavior of the spark module, or possible damage.

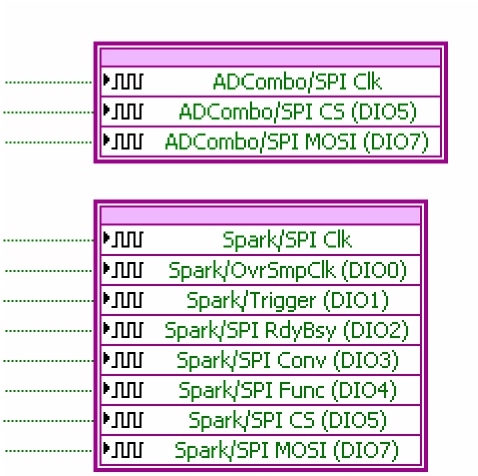


Figure 20a. Representative FPGA output nodes for ADCombo and Spark modules with correct output item selection.

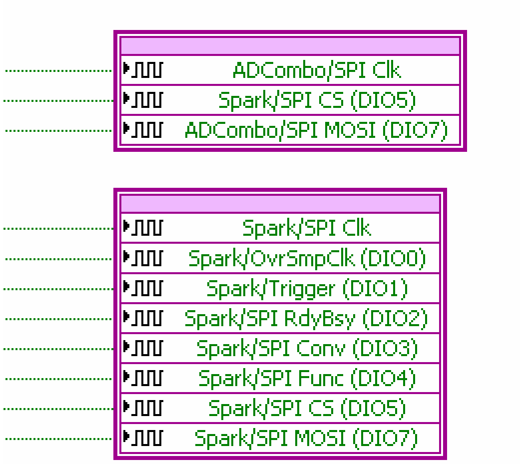
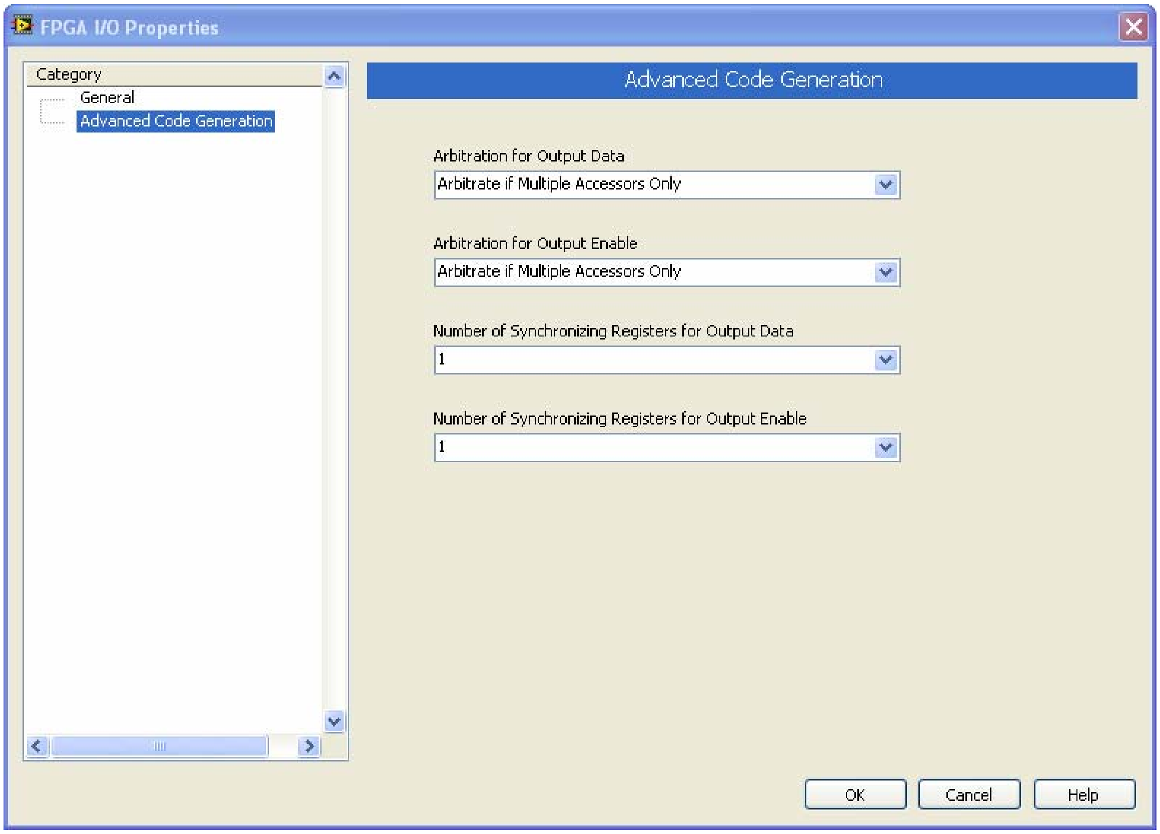


Figure 20b. Representative FPGA output nodes for ADCombo and Spark modules with incorrect output item selection within the ADCombo output node. The Spark (DIO5) output is selected in multiple nodes and therefore being driven by multiple sources. This will cause strange behavior or damage to the spark module.

One way to help prevent such coding mistakes is to prefix all FPGA I/O item names with an appropriate unique module name via the FPGA I/O creation dialog or via the project explorer, after the I/O item is created. This will make the coding mistake recognizable from the block diagram. Another way this situation can be prevented, even when a coding mistake is made, is by making sure that all FPGA output node items are configured to "Arbitrate if Multiple Accessors Only." When outputs are configured this way and they are used within a Single Cycle Loop (as is required by Drivven cRIO module kits), then a compile error will be generated if multiple sources are driving FPGA output node items. Then appropriate corrective action can be taken. FPGA output node items can be configured via the FPGA I/O properties dialog, by right clicking on the FPGA I/O item within the project explorer. FPGA output node properties should be set according to the following dialog screen shot.



Arbitration for Output Data: Arbitrate if Multiple Accessors Only

Arbitration for Output Enable: Arbitrate if Multiple Accessors Only

Number of Synchronizing Registers for Output Data: 1

Number of Synchronizing Registers for Output Enable: 1

Figure 21. FPGA I/O Properties dialog configuration for cRIO modules.

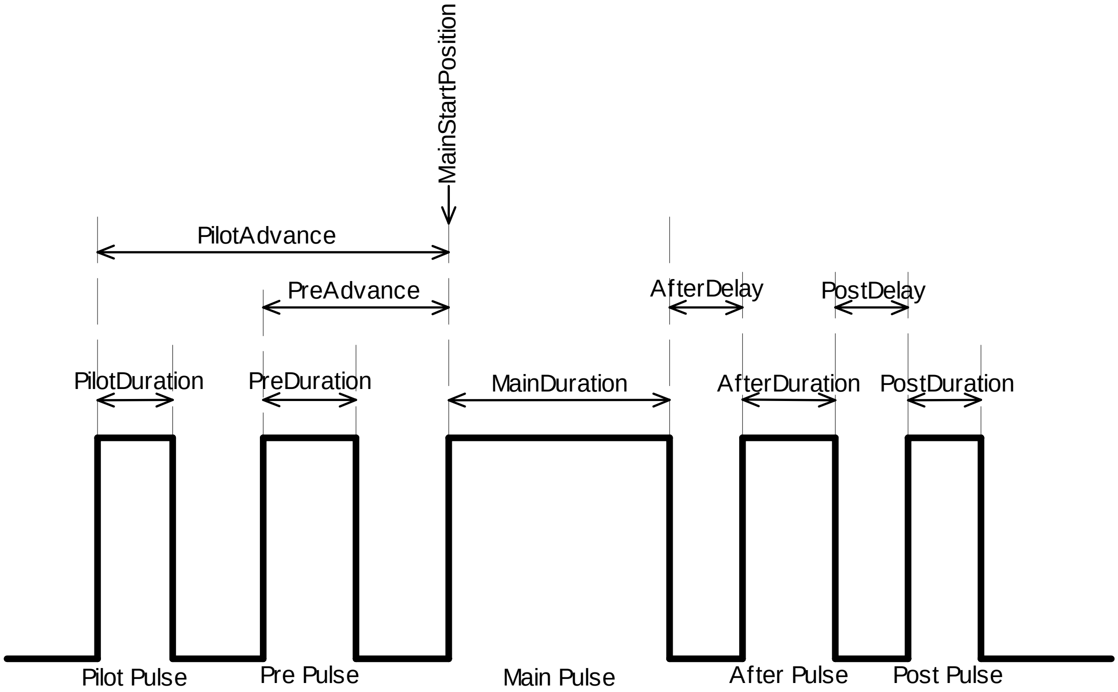
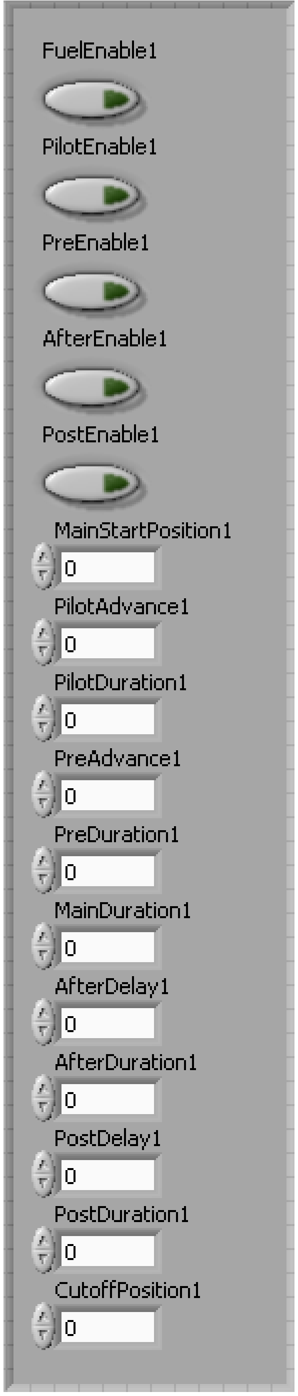


Figure 22. Fuel pulse timing diagram.

DIControl1, DIControl2, DIControl3 (Cluster)
The DIControl1, DIControl2 and DIControl3 Clusters should be terminated with control clusters and made available as complete clusters for interfacing to the LabVIEW RT level. No FPGA code interface is required with any of the members of these clusters. However, their elements will be described in detail here for proper interfacing at the RT level. Each of these three clusters contain the control elements for the pulse timing of driver channels one, two and three respectively. DIControl1 cluster is shown below. Refer to the diagram in figure 22 for understanding the timing parameters for generating a five-pulse diesel injection event.



MainStartPosition1: 0
PilotAdvance1: 0
PilotDuration1: 0
PreAdvance1: 0
PreDuration1: 0
MainDuration1: 0
AfterDelay1: 0
AfterDuration1: 0
PostDelay1: 0
PostDuration1: 0
CutoffPosition1: 0

FuelEnable (boolean): When TRUE, all five fuel pulses are potentially enabled, depending on the other four enable booleans. When FALSE (default), all five fuel pulses are disabled. Note that there is not a “MainEnable” boolean since the main pulse is always present when the FuelEnable boolean is TRUE. The FuelEnable boolean can be used as the channel’s global pulse enabling control.

PilotEnable (boolean): When TRUE, the pilot pulse is enabled. When FALSE (default), the pilot pulse is disabled.

PreEnable (boolean): When TRUE, the pre pulse is enabled. When FALSE (default), the pre pulse is disabled.

AfterEnable (boolean): When TRUE, the after pulse is enabled. When FALSE (default), the after pulse is disabled.

PostEnable (boolean): When TRUE, the post pulse is enabled. When FALSE (default), the post pulse is disabled.

MainStartPosition (uint16): The main fuel pulse is generated with a leading edge coinciding with MainStartPosition. The length of the pulse will be according to MainDuration. The units of MainStartPosition are CAT. The timing of the pilot and pre pulses is referenced to MainStartPosition. The timing of the after pulse is referenced to the end of the main pulse.

Drivven provides a utility VI which can be implemented at the LabVIEW RT level for performing the conversion of MainStartPosition timing or CutoffPosition timing, in CAD before TDC, to absolute CAT. The VI named Offset2CAT.vi can be used to convert advance, with respect to an absolute TDC, to CAT. This VI icon is shown in figure 23.

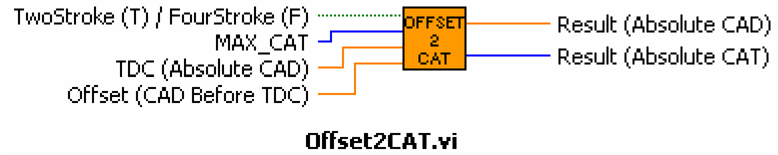


Figure 23. Offset to CAT conversion VI.

PilotAdvance (uint32): 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 in terms of 40 MHz clock ticks and is internally limited to 18 bits. Values larger than 18 bits will roll over from zero.

Duration(uint32 ticks) = Duration(msec) * 40,000.

Drivven provides a utility VI which can be implemented at the LabVIEW RT level for performing this calculation. The VI named time2ticks.vi can be used to convert Time in milliseconds to uint32 ticks. This VI icon is shown in figure 24.

Figure 24. Time to Ticks conversion VI.

PilotDuration (uint32): Determines the length of the pilot fuel pulse. PilotDuration is entered in terms of 40 MHz clock ticks and is internally limited to 18 bits. Values larger than 18 bits will roll over from zero. Use the time2ticks.vi VI for converting time in milliseconds to uint32 ticks.

PreAdvance (uint32): 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 in terms of 40 MHz clock ticks and is internally limited to 18 bits. Values larger than 18 bits will roll over from zero. Use the time2ticks.vi VI for converting time in milliseconds to uint32 ticks.

PreDuration (uint32): Determines the length of the pre fuel pulse. PreDuration is entered in terms of 40 MHz clock ticks and is internally limited to 18 bits. Values larger than 18 bits will roll over from zero. Use the time2ticks.vi VI for converting time in milliseconds to uint32 ticks.

MainDuration (uint32): Determines the length of the main fuel pulse. MainDuration is entered in terms of 40 MHz clock ticks and is internally limited to 18 bits. Values larger than 18 bits will roll over from zero. Use the time2ticks.vi VI for converting time in milliseconds to uint32 ticks.

AfterDelay (uint32): 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 in terms of 40 MHz clock ticks and is internally limited to 18 bits. Values larger than 18 bits will roll over from zero. Use the time2ticks.vi VI for converting time in milliseconds to uint32 ticks.

AfterDuration (uint32): Determines the length of the after fuel pulse. AfterDuration is entered in terms of 40 MHz clock ticks and is internally limited to 18 bits. Values larger than 18 bits will roll over from zero. Use the time2ticks.vi VI for converting time in milliseconds to uint32 ticks.

PostDelay (uint32): 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 in terms of 40 MHz clock ticks and is internally limited to 18 bits. Values larger than 18 bits will roll over from zero. Use the time2ticks.vi VI for converting time in milliseconds to uint32 ticks.

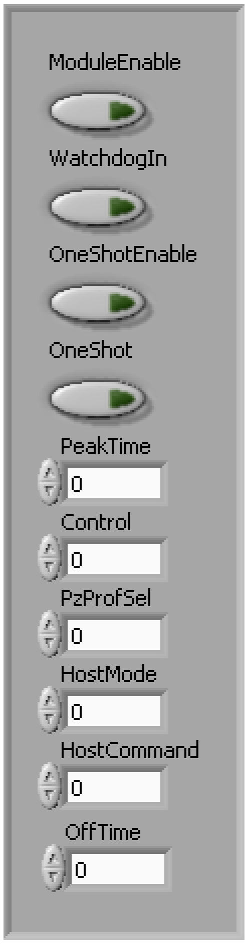
PostDuration (uint32): Determines the length of the post fuel pulse. PostDuration is entered in terms of 40 MHz clock ticks and is internally limited to 18 bits. Values larger than 18 bits will roll over from zero. Use the time2ticks.vi VI for converting time in milliseconds to uint32 ticks.

CutoffPosition (uint16): All fuel pulse activity for the active channel is "Cutoff" at CutoffPosition and reset for the next channel. CutoffPosition must always be at least 45 CAD after MainStartPosition. If this minimum spacing is not maintained, then fuel commands will be generated with incorrect timing. The units of CutoffPosition are CAT. CuttoffPosition may need to be significantly more than 45 CAD to prevent the after and post pulses from being cutoff, depending on their delays, duration and the engine speed.

Since diesel fuel pulses will not overlap among the three possible driver channels, a single fuel pulse generator core is used for serving all three channels and is multiplexed to each channel. The CutoffPosition is used for the multiplexing trigger event. Therefore the cutoff position cannot be so long after the pulse group that it masks the pulses of the next channel.

Use the Offset2CAT.vi VI to convert cutoff position, with respect to an absolute TDC, to CAT.

DIControl4 (Cluster)
The DIControl4 Cluster should be terminated with a control cluster and made available as a complete cluster for interfacing to the LabVIEW RT level. No FPGA code interface is required with any of the members of this cluster. However, their elements will be described in detail here for proper interfacing at the RT level.



ModuleEnable (boolean): If a DI driver module is inserted in the proper slot, externally powered, and ModuleEnable is TRUE, then software begins communicating with the module and allows the module to operate. When the module is properly recognized, then the ModulePresent boolean within the DIData cluster will be set to TRUE.

WatchdogIn (boolean): WatchdogIn must be toggled at a rate greater than 5Hz. This should only be performed at the RT level. DO NOT toggle the watchdog at the FPGA level. Toggling the watchdog at the FPGA level would bypass the software safety feature for which it is intended.

OneShotEnable (boolean): This parameter is only used by the DI Calibrator application and should not be used during normal run-time operation. A constant of FALSE should be wired to this input.

OneShot (boolean): This parameter is only used by the DI Calibrator application and should not be used during normal run-time operation. A constant of FALSE should be wired to this input.

PeakTime (uint8): 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.

Drivven provides a utility VI which MUST be implemented at the LabVIEW RT level for calculating the proper FPGA PeakTime. The VI named di_peaktime2ticks.vi can be used to convert PeakTime in milliseconds to uint8 ticks. This VI icon is shown in figure 25.

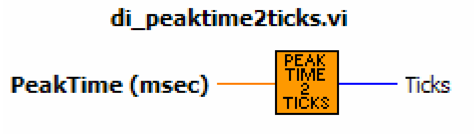


Figure 25. PeakTime to Ticks conversion VI.

Control (uint8): The value for this parameter is provided by the DI CAL VI (di_vt_rt_cal_revx.vi) at the RT level. The Control output of DI CAL VI should be wired directly to this FPGA parameter.

PzProfSel (uint8): Selects one of three current profiles for opening piezo type injectors.

HostMode (uint8): The value for this parameter is provided by the DI CAL VI (di_vt_rt_cal_revx.vi) at the RT level. The HostMode output of DI CAL VI should be wired directly to this FPGA parameter.

HostCommand (uint32): The value for this parameter is provided by the DI CAL VI (di_vt_rt_cal_revx.vi) at the RT level. The HostCommand output of DI CAL VI should be wired directly to this FPGA parameter.

OffTime (uint16): Determines the length of inactivity following each fuel pulse. It is recommended to set this value to at least 100 usec. OffTime provides a minimum amount of time for the command to be off to allow the injector valve to shut off before turning the injector back on. OffTime is entered in terms of 40 MHz clock ticks and is internally limited to 16 bits. Values larger than 16 bits will roll over from zero.

DIData (Cluster)
The DIData Cluster should be terminated with an indicator cluster and made available as a complete cluster for interfacing to the LabVIEW RT level. No FPGA code interface is required with any of the members of this cluster. At the RT level, this cluster should be wired directly to the DIDataIn input cluster of the DI DATA CNVRT VI (di_vt_rt_data_convert_revx.vi). The DI DATA CNVRT VI will provide the proper data conversion for display in engineering units.

Non-Clustered Output Parameters
Data (uint32): This stand-alone output of the FPGA VI is only used for the DI Calibrator application. It should not be used for normal run-time operation.

FIFOWrite (boolean): This stand-alone output of the FPGA VI is only used for the DI Calibrator application. It should not be used for normal run-time operation.

OLIntegrator (uint16): This stand-alone output of the FPGA VI is only used for the DI Calibrator application. It should not be used for normal run-time operation.

RT VI Implementation

Drivven provides two LabVIEW RT VIs which must be implemented at the RT level to properly interface with the DI Driver Module.

The first VI discussed is the DI CAL VI (di_vt_rt_cal_revx.vi) which provides the necessary values to the Control, HostMode and HostCommand parameters at the FPGA level.

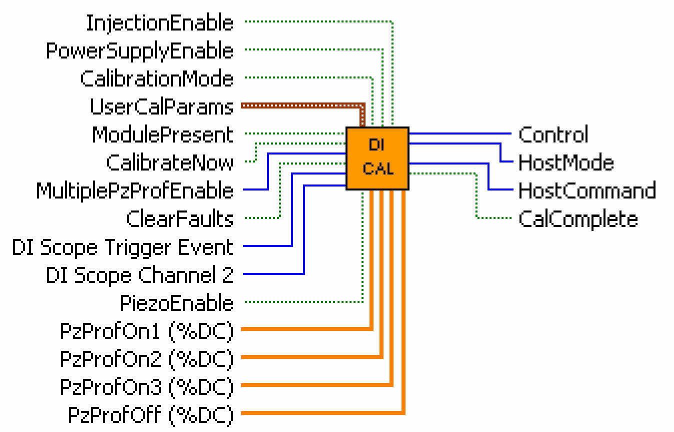


Figure 26. di_vt_rt_cal_revx.vi VI icon with leads.

Non-Clustered Input Parameters

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.

PowerSupplyEnable (boolean): When TRUE, the module internal boost power supply is enabled and will maintain the working voltage specified by HVTargetInject and HVTargetNonInject. When FALSE, the module internal boost power supply is disabled.

CalibrationMode (boolean): When TRUE, the module will sample only the injector current during injection events. This provides potentially better current control with very low inductance injector solenoids because the current is sampled at twice the normal frequency. The internal boost supply will not operate during injections events when CalibrationMode is enabled, but will resume at the end of injection. When CalibrationMode is FALSE, injector current is sampled at the normal frequency and the working voltage can be controlled during injection events. This boolean is named “CalibrationMode” because it was originally implemented to show current traces within DI Calibrator with a better resolution. Drivven recommends that this value be set to TRUE.

ModulePresent (boolean): This input boolean must be wired directly from the ModulePresent output boolean of the DI DATA CONVERT (di_vt_rt_data_convert_revx.vi).

CalibrateNow (boolean): When this one-shot is pressed, the DI CAL VI will begin communicating the various calibration parameters to the module. The CalComplete output boolean will be FALSE during calibration and will be set TRUE when calibration is complete.

MultiplePzProfEnable (uint8): Determines whether one, two, or three piezo open profiles are loaded to the DI Driver module during the calibration process. Multiple profiles require more time to send the calibration data. A value of 0 enables **PzProfOn1**. A value of 1 enables **PzProfOn1** and **PzProfOn2**. A value of 2 enables **PzProfOn1**, **PzProfOn2** and **PzProfOn3**. PzProfSel is used to select between profiles.

ClearFaults (boolean): When this one-shot is pressed, the DI CAL VI will attempt to clear any critical fault conditions which are set.

DI Scope Trigger Event (uint8): This input is only used for the DI Calibrator application. It should not be used for normal run-time operation. It may be left unconnected.

DI Scope Channel 2 (uint8): This input is only used for the DI Calibrator application. It should not be used for normal run-time operation. It may be left unconnected.

PiezoEnable (boolean): When TRUE, piezo injector functionality is enabled and solenoid injector functionality is disabled. Refer to the section titled "Piezo Injector Mode" for details of using piezo mode.

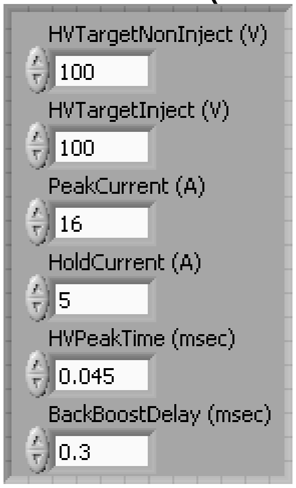
PzProfOn1 (%DC) (single array): Array of duty cycles for controlling current to the piezo stack during the open phase.

PzProfOn2 (%DC) (single array): Array of duty cycles for controlling current to the piezo stack during the open phase.

PzProfOn3 (%DC) (single array): Array of duty cycles for controlling current to the piezo stack during the open phase.

PzProfOff (%DC) (single array): Array of duty cycles for controlling current to the piezo stack during the close phase.

UserCalParams (Cluster)



HVTargetNonInject (V)
100
HVTargetInject (V)
100
PeakCurrent (A)
16
HoldCurrent (A)
5
HVPeakTime (msec)
0.045
BackBoostDelay (msec)
0.3

HVTargetNonInject (V): The working voltage set point of the internal boost power supply while no injection events are taking place. Refer to the hardware section of this manual for instructions of proper use of this parameter.

HVTargetNonInject (V): The working voltage set point of the internal boost power supply while no injection events are taking place. Refer to the hardware section of this manual for instructions of proper use of this parameter.

PeakCurrent (A): The target current level which is driven through the injector solenoid during PeakTime. Refer to the hardware section of this manual for instructions of proper use of this parameter.

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. Refer to the hardware section of this manual for instructions of proper use of this parameter.

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. Refer to the hardware section of this manual for instructions of proper use of this parameter.

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. Refer to the hardware section of this manual for instructions of proper use of this parameter.

Non-Clustered Output Parameters

Control (uint8): This output must be wired directly to the Control parameter of the FPGA interface.

HostMode (uint8): This output must be wired directly to the HostMode parameter of the FPGA interface.

HostCommand (uint32): This output must be wired directly to the HostCommand parameter of the FPGA interface.

CalComplete (boolean): When TRUE, the calibration procedure to the module has completed. When FALSE, either a calibration has not yet been attempted or a calibration is currently in progress. This output is for reference only and is not required for module control.

The second RT VI discussed is the DI DATA CNVRT (di_vt_rt_data_convert_revx.vi) which converts the data within the DIData cluster of the FPGA interface to engineering units. The DIData cluster from the FPGA should be wired directly to the DIDataIn input cluster of this VI.

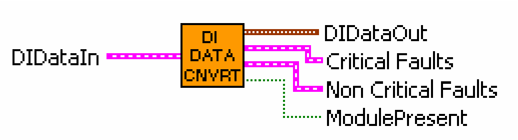
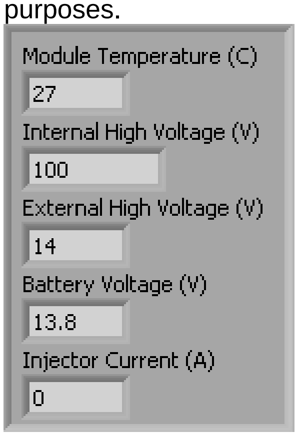


Figure 27. di_vt_rt_data_convert_revx.vi VI icon with leads.

DIDataOut (Cluster)
These parameters are for reference only. There is no need to use them for module control purposes.



Module Temperature (C)
27
Internal High Voltage (V)
100
External High Voltage (V)
14
Battery Voltage (V)
13.8
Injector Current (A)
0

Module Temperature (C): The module board temperature in degrees C. The module will shutdown operations if this temperature exceeds approximately 85C.

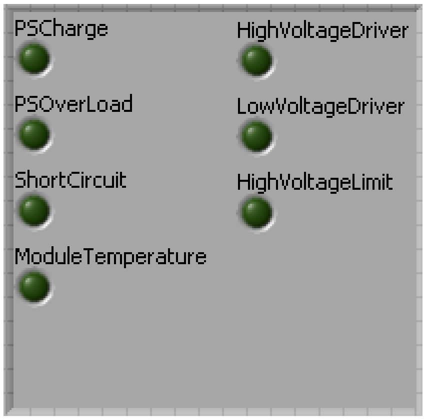
Internal High Voltage (V): The internal working voltage supplied by the internal boost power supply.

External High Voltage (V): The externally supplied voltage to the EXT PWR (7) terminal of the module. If External High Voltage exceeds 6V, then the internal boost power supply will automatically shut down.

Battery Voltage (V): The voltage supplied to the BATT (0) terminal of the module.

Injector Current (A): The current flowing through the active injector channel. This parameter does not offer much since the update rate to this parameter is much slower than actual injection events.

Critical Faults (Cluster)
Any of the boolean faults contained within the Critical Faults cluster will cause all operations of the DI Driver Module to shut down.



PSCharge (boolean): The internal boost power supply is not able to increase its supplied voltage. Refer to the hardware section of this manual for additional information on this parameter.

PSOverLoad (boolean): The internal boost power supply is being loaded beyond its long term limit. If operation were to continue, the module would overheat. Refer to the hardware section of this manual for additional information on this parameter.

ShortCircuit (boolean): A short circuit condition has been detected with the INJ +/- terminals of the module. Refer to the hardware section of this manual for additional information on this parameter.

ModuleTemperature (boolean): The internal temperature of the module has exceeded its limit of 85C. Refer to the hardware section of this manual for additional information on this parameter.

HighVoltageDriver (boolean): The high voltage portion of the injector driver circuit has been overloaded. This is another indication of a short circuit condition which has been detected before current flow has exceeded its limit. Refer to the hardware section of this manual for additional information on this parameter.

LowVoltageDriver (boolean): The battery voltage portion of the injector driver circuit has been overloaded. This is another indication of a short circuit condition which has been detected before current flow has exceeded its limit. Refer to the hardware section of this manual for additional information on this parameter.

HighVoltageLimit (boolean): The internal boost power supply voltage has exceeded its limit of 155V. This is typically caused by excessive back-boosting from the injector solenoid and is likely caused by an improper calibration of the driver module for the particular injector solenoid being used. Refer to the hardware section of this manual for additional information on this parameter.

Non Critical Faults (Cluster)
The boolean faults contained within the Non Critical Faults cluster will not cause any operations of the DI Driver Module to shut down.

FIXOpenCircuit (boolean): The module injector driver circuit has detected an open circuit between the INJ +/- terminals of the module. This fault is automatically cleared when the fault condition is removed.

Non-Clustered Output Parameters

ModulePresent (boolean): When TRUE, then software has properly detected a DI Driver Module and can begin calibration and operation. When FALSE, then software has not yet detected the presence of a DI Driver Module and module operations cannot continue. In order to be detected, the driver module must be properly inserted in its slot, powered at the BATT (0) terminal, and ModuleEnable must be TRUE. After the ModulePresent boolean is set to TRUE, if the power at BATT (0) terminal is removed then the ModulePresent boolean will reset to FALSE. If power is reapplied while ModuleEnable boolean remains TRUE, then ModulePresent will return to TRUE and the module will be recalibrated. After the ModulePresent boolean is set to TRUE, if the module is removed from its slot, then the ModulePresent boolean will be set to FALSE. If the module is reinserted, then it will be detected again and recalibrated. This boolean must be wired directly to the ModulePresent input boolean of the DI CAL VI (di_vt_rt_cal_revx.vi).

Position Conversion and Notes:

MainStartPosition and CutoffPosition are not entered as crank angle degrees (CAD), but crank angle ticks (CAT) which can be calculated from a CAD value. Each CAT represents a fractional CAD. A conversion factor referred to as Crank Angle Conversion (CAC) is required to convert between CAD and CAT. The units of CAC are degrees per tick. The following equations apply:

CAD (degrees) = CAT (ticks) * CAC (degrees per tick)
CAT (ticks) = CAD (degrees) / CAC (degrees per tick)

If the engine is a two-stroke then CAC is calculated as:

CAC (degrees per tick) = 360 (degrees) / MAX_CAT (ticks)

If the engine is a four-stroke then CAC is calculated as:

CAC (degrees per tick) = 720 (degrees) / MAX_CAT (ticks)

MainStartPosition and CutoffPosition are with respect to the absolute zero (0) CAT position which corresponds to the location of tooth 0 on the crank trigger wheel. Tooth 0 will be documented for each type of pattern within the EPT documentation.

When interfacing to the DI FPGA VIs from the CPU, the programmer can use the Offset2CAT VI for converting advance offset values, with respect to TDC, directly to CAT values. The Offset2CAT VI is shown in figure 23. This VI requires knowledge of the EPT Stroke parameter and MAX_CAT values, as well as the TDC of the particular cylinder being targeted.

Fuel Command Scheduling:
The DI FPGA VI provides features that ensure the best possible fuel command delivery, even while the CPU makes modifications to MainStartPosition and Duration asynchronously to engine position.

Modifications to Duration:

1. Duration can be modified at any time.
2. If Duration is modified during the fuel pulse to a value less than the already accumulated duration, then the pulse is immediately terminated.
3. If Duration is modified during the fuel pulse to a value greater than the already accumulated duration, then the pulse is continued to the new value of Duration, unless CutoffPosition is encountered first.
4. An off-time is enforced at the end of each fuel pulse, specified by the OffTime parameter.

Modifications to MainStartPosition:
MainStartPosition can be modified at any time. However, the value must not be advanced by more than 45 CAD within a single engine cycle. This value is referred to as the History Window. The FPGA VI continually checks the requested MainStartPosition with respect to the current crank position. If the MainStartPosition is modified by the CPU to a position in the past, the FPGA VI uses the History Window to determine whether a late fuel pulse should be started.

1. For example, let's assume that a fuel pulse is scheduled for a MainStartPosition of 200 Absolute CAD (ACAD). Let's also assume that the CurrentPosition of the EPT VI is 190 ACAD when the CPU modifies MainStartPosition to 180 ACAD, which is in the recent past by 10 CAD. Since this is less than the 45 CAD History Window, then the FPGA VI will immediately start the fuel pulse even though it is late. The pulse width will still be exactly according to Duration.
2. As another example, let's assume that a fuel pulse is scheduled for a MainStartPosition of 200 ACAD. Let's also assume that the CurrentPosition of the EPT VI is 190 ACAD when the CPU modifies MainStartPosition to 120 ACAD, which is in the far past by 80 CAD. Since this is greater than the 45 CAD History Window, the FPGA VI will not generate a late pulse, effectively skipping a cycle without a fuel pulse. The following cycle will have a pulse delivered starting at 120 ACAD.

CutoffPosition must be set with the following conditions in mind:

1. CutoffPosition must be set to a position at least 45 CAD after MainStartPosition. If this minimum spacing is not maintained, then fuel commands will be generated with incorrect timing. In reality, the CutoffPosition may need to be significantly larger than 45 CAD after MainStartPosition to allow for the after and post fuel pulses. This will depend on the duration of these pulses as well as the engine speed.

Examples

The following screen capture in figure 28 shows a LabVIEW FPGA block diagram demonstrating the interface between EPT VIs and the DI VI. No external signals are wired to the EPTInSig cluster, thus requiring simulation of these signals. However, those signals may be provided from the AD Combo or VR/Hall Module Kits. This FPGA application is entirely contained within a single cycle loop, clocked at the required 40 MHz. Notice that the EPT VI provides the FuelSparkSupervisor cluster to be wired directly to the compatible fuel and spark VIs. Refer to the LabVIEW FPGA documentation for details about configuring cRIO I/O pins. A similar example VI is provided for any EPT, Fuel or Spark product ordered from Drivven.

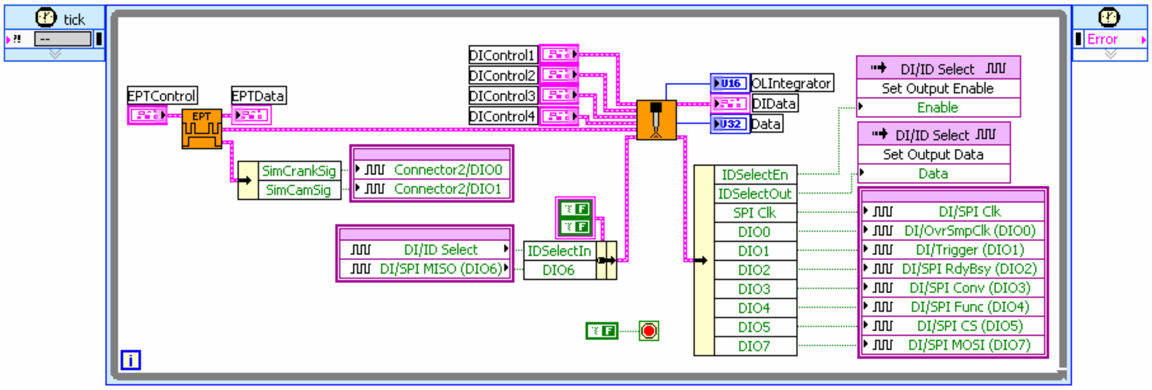


Figure 28. LabVIEW FPGA Block diagram example.