ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *National Instruments / Drivven D000006* at our website: *Click HERE*

PFI Driver Module Kit User's Manual

D000006 Rev E

July 17, 2007

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

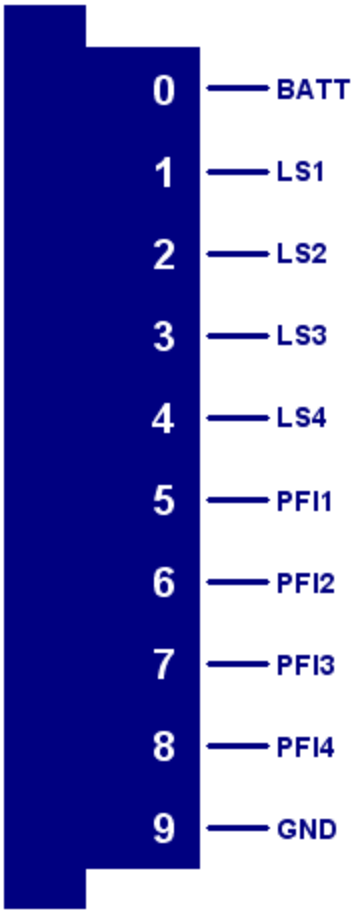
Introduction

The Port Fuel Injector Driver Module Kit provides a CompactRIO (cRIO) module for driving low and high impedance Port Fuel Injectors (PFI) and general purpose automotive solenoid valves. The kit includes a LabVIEW FPGA VI for controlling each driver channel. Each PFI driver is individually controlled for timing and duration. Each general purpose low-side solenoid driver is PWM controlled, capable of 0-100% duty cycle operation.

Features:

- 4-channel low or high impedance PFI driver
  - Peak/hold current control for low impedance injectors
  - 4-amp peak / 1-amp hold current profile (standard configuration)
  - Tunable peak duration
  - Open/short circuit detection and reporting with short circuit disable
- 4-channel general purpose low-side solenoid driver
  - 1.5-amp continuous duty
  - PWM controlled with 0-100% duty cycle operation
  - Open/short circuit detection and reporting with short circuit disable
- External power supply of 6-16V (Rev C)
- External power supply of 6-32V (Rev D and later)

Pinout



Hardware

The PFI Driver Module Kit provides four port fuel injector drivers and four general purpose low-side solenoid drivers in a National Instruments CompactRIO module.

Powering the Module

The PFI 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 C and earlier have a maximum power 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.

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.

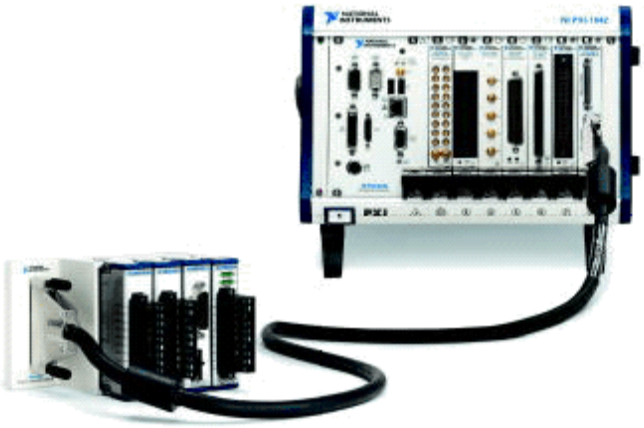


Figure 1b. PXI platform compatible with Drivven CompactRIO modules.

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 cannot be supported by Drivven or National Instruments.

Port Fuel Injector Drivers

Each PFI driver channel is capable of driving low or high impedance port fuel injectors.

Driving Low Impedance Injectors
Low impedance injectors will have approximately 2 ohms resistance across the solenoid. The low resistance allows higher current through the solenoid to enable faster valve opening times. However, the high current is not needed for the entire duration of the injection event in order to keep the valve open. The initial current level is referred to as the peak current and the current level for the remaining injection duration is referred to as the hold current. The standard configuration for the PFI drivers is a peak current of 4 amps and a hold current of 1 amp.

At the beginning of each injection event, the driver attempts to control the injector at 4 amps for the requested peak time (refer to Software for PFI controls). When the peak time expires, the current control folds back to the hold level of 1 amp for the remaining duration of the event. The requested peak time is updated at 125 usec intervals.

If for the entire duration of the injection event, neither peak nor hold current levels are reached, then an open circuit will be reported for the channel. It is possible that an open condition may be falsely reported under certain conditions, even though a fuel injector is properly connected. Examples of this would be very low battery voltage, poor connections which introduce added resistance, or total injection duration which is less than the requested peak time such that the peak current is never reached.

Each PFI driver channel automatically protects itself from an accidental short circuit to battery. At the beginning of each injection event, the current rise rate is monitored in order to detect a short circuit. If the current rises faster than expected for a typical port fuel injector solenoid, then a short circuit error flag will be reported for that channel and the channel will be disabled until software clears the error flag. Short circuit detection is also performed continuously at a later time within each injection event. The short circuit error flag for each channel will not be cleared while an injection event is being commanded for that channel. It is strongly recommended that the user not continually clear the short circuit error flags, as this may lead to driver circuit damage in the event of an actual short circuit. Even though the short circuit detection and disable feature is present, it is strongly recommended that the wiring harness include a 20 amp fuse for four low impedance injectors as shown in Figure 5.

Figure 2 is a representation of a typical peak/hold current trace if one of the injector leads were monitored with a current probe and oscilloscope.

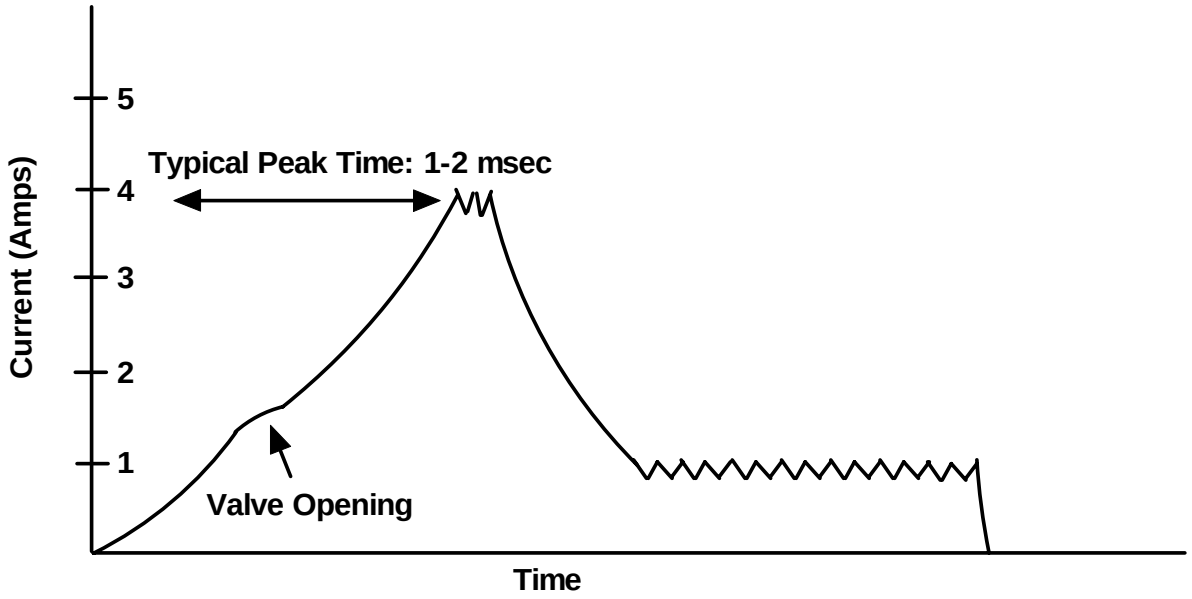


Figure 2. Peak/hold current profile.

Driving High Impedance Injectors

High impedance injectors may be used with the PFI driver circuits without any hardware or software configuration changes. High impedance injectors will have approximately 12 ohms resistance across the solenoid. Typically, less than 1 amp is required to open and hold the solenoid valve.

The software-requested peak time will not affect proper operation of a high impedance injector. However, depending on the actual resistance of the solenoid and the actual battery voltage, the injector may saturate at varying current levels. If the peak time were set to a non-zero value, then the driver circuit would attempt to control current to 4 amps during the requested peak time. Obviously, this current level will never be reached because the solenoid resistance is too high. The hold current level of 1 amp may or may not be reached, depending on the previously mentioned variables of resistance and battery voltage. If the hold current level is reached, then current control will take effect as with a low impedance injector.

The open circuit flag should be ignored when driving high impedance injectors because the hold current level of 1 amp may never be reached, thus reporting a false open circuit.

Short circuit detection operates identically to that of low impedance injector control and the channel will be disabled if a short is detected, until the error flag is cleared in software. Even though the short circuit detection and disable feature is present, it is strongly recommended that the wiring harness include a 7 amp fuse for four high impedance injectors as shown in Figure 5.

It is recommended that peak time be set to zero when controlling high impedance injectors.

Figures 3 and 4 represent a current trace of a high impedance injector if one of the injector leads were monitored with a current probe and oscilloscope.

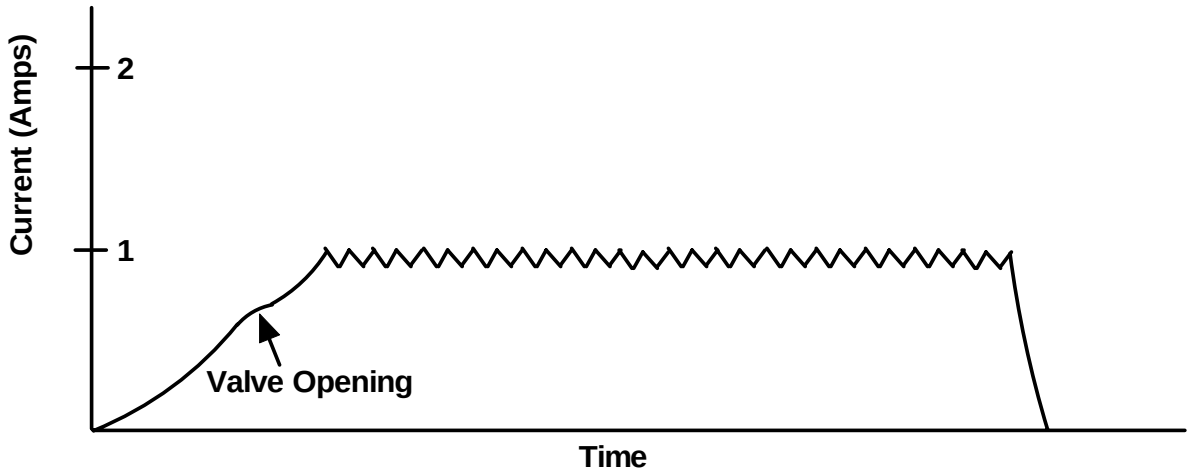


Figure 3. Hold current reached and controlled.

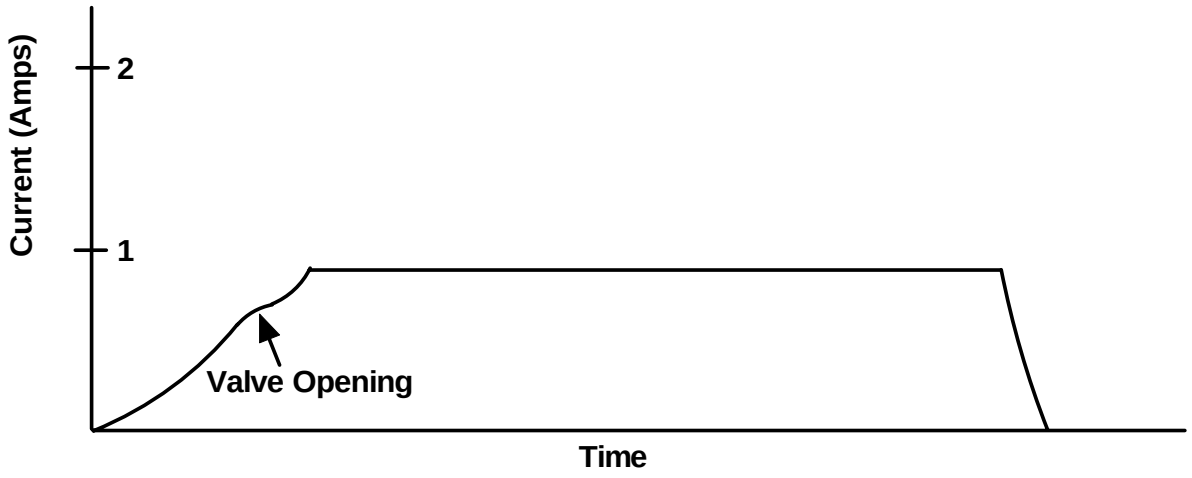


Figure 4. Hold current never reached.

WARNING: Do not attempt to drive low impedance injectors with the general purpose low-side switch drivers. The low-side drivers are only rated for 1.5 amps. The resulting higher current levels could damage the low-side drivers.

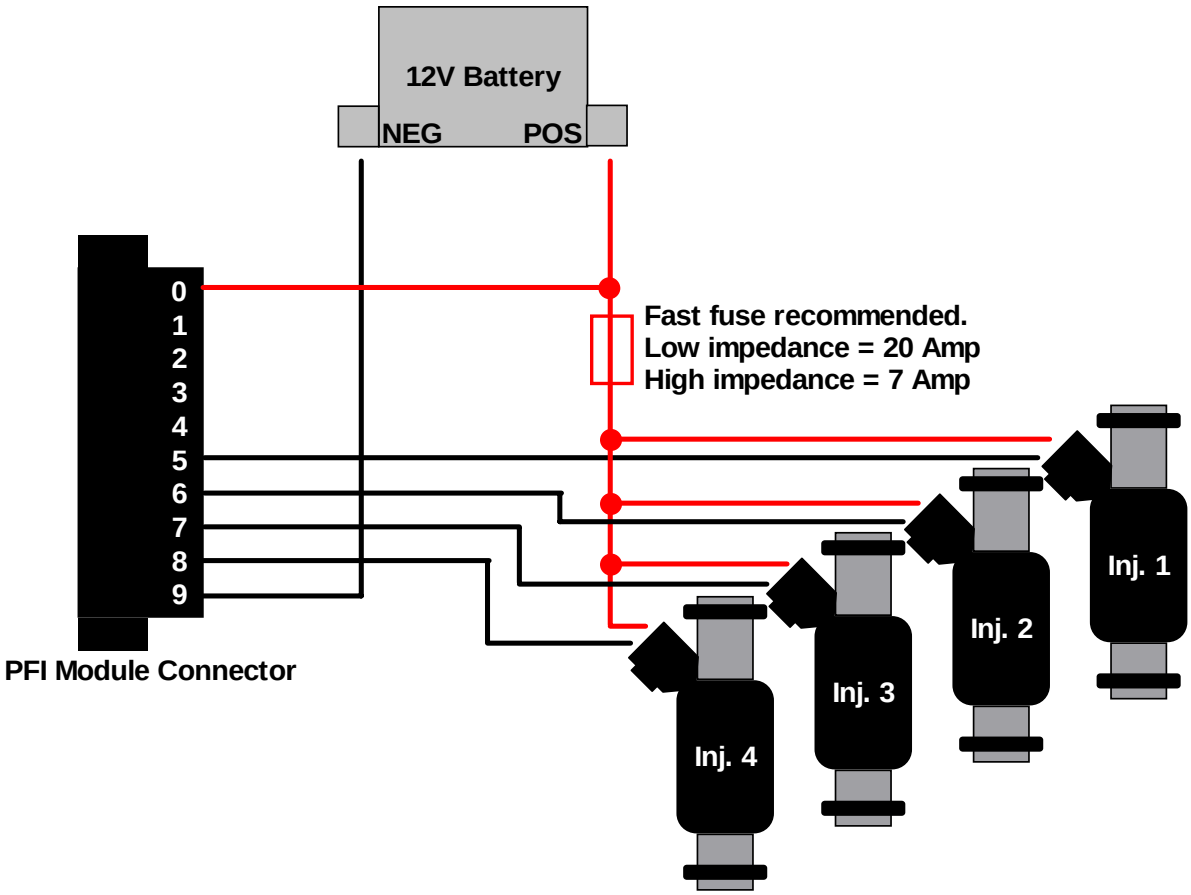


Figure 5. Connecting port fuel injectors to the driver module.

Low-Side Drivers

The low-side drivers are capable of driving a wide variety of automotive relays and actuators. Examples of actuators include EGR valves, turbo wastegate valves, fuel pressure regulators, or any single-solenoid-actuator which draws less than 1.5 amps at the highest expected battery voltage.

WARNING: Do not attempt to drive low impedance injectors or solenoids with the general purpose low-side drivers. The low-side drivers are only rated for 1.5 amps. Higher current levels could damage the low-side drivers.

Each channel is independently controlled within the module via pulse-width modulation (PWM). PWM duty cycle from 0 to 100% is possible. Software on the RIO FPGA device communicates several parameters for each channel to operate. Those parameters are Enable, Period and PulseWidth. All parameters are communicated serially at 125 usec intervals. Each PWM controller utilizes a 24-bit timer operating at 4 MHz. This provides a resolution of 250 nsec and a minimum PWM frequency of 0.24 Hz. Since 0% and 100% duty cycles are possible, relays and actuators may be controlled in continuous on or off states.

Each low-side driver channel automatically tests for open circuit conditions. When the low-side switch is commanded open, the circuit is tested for an open condition. Upon detection of an open circuit, an open circuit error flag will be reported. The error flag may be cleared by software. Driver circuit operation is not affected by an open circuit error flag that is left asserted.

Each low-side driver channel automatically protects itself from an accidental short circuit to battery. When the low-side switch is commanded closed, the circuit is tested for a short. Upon detection of a short, a short circuit error flag will be reported for the channel and the channel will be disabled until software clears the error flag. The short circuit error flag for each channel will not be cleared while the switch is being commanded closed. It is strongly recommended that the user not continually clear the short circuit error flags, as this may lead to driver circuit damage in the event of an actual short circuit. Even though the short circuit detection and disable feature is present, it is strongly recommended that the wiring harness include a 7 amp fuse for four solenoids as shown in Figure 6.

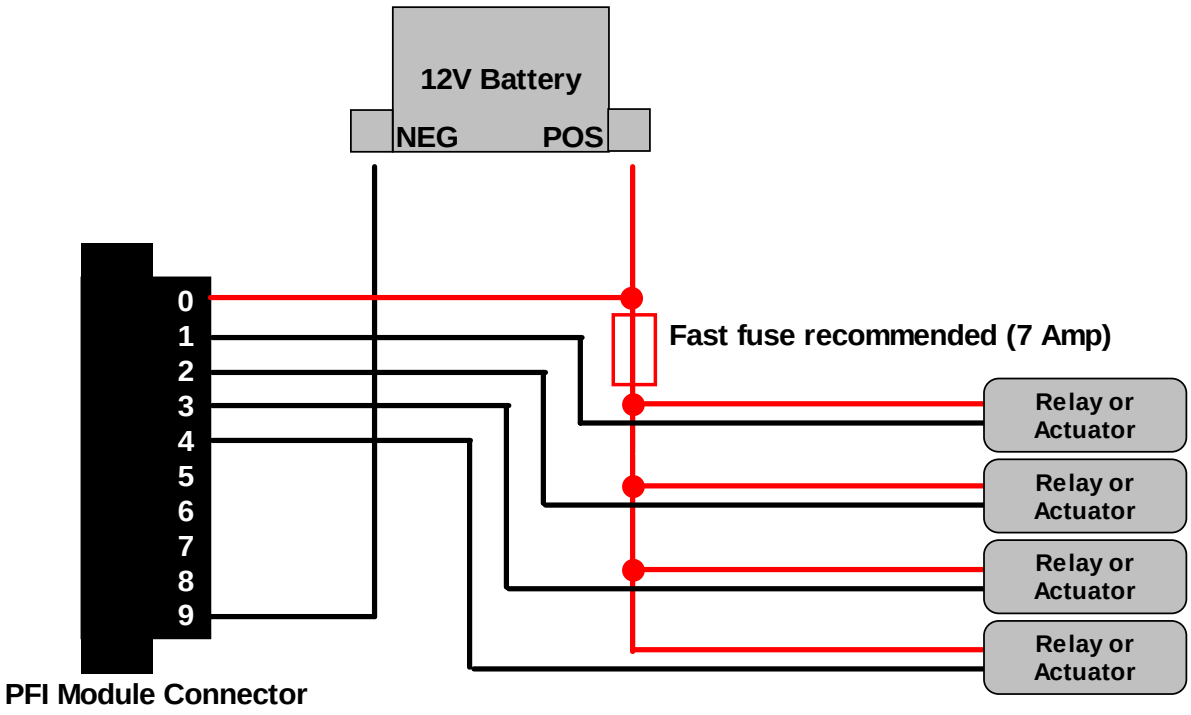


Figure 6. Connecting general purpose solenoids to the low-side driver channels

Software

The Port Fuel Injection Driver Module Kit is provided with two different LabVIEW FPGA VIs. Each VI includes an interface for controlling both PFI and low-side drivers. The VI having the "ea" designator supports end-angle fuel pulse timing, while the VI having the "sa" designator supports start-angle timing. End-angle fuel pulse timing requires more FPGA resources, therefore an optimization option is provided for the programmer. Figure 7 shows the icon which represents both supplied VIs, having identically named terminals. However, the contents of the PortFuelControl cluster is different.

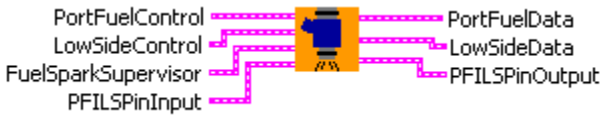


Figure 7. PFI VI icon with leads.

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 VIs 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.

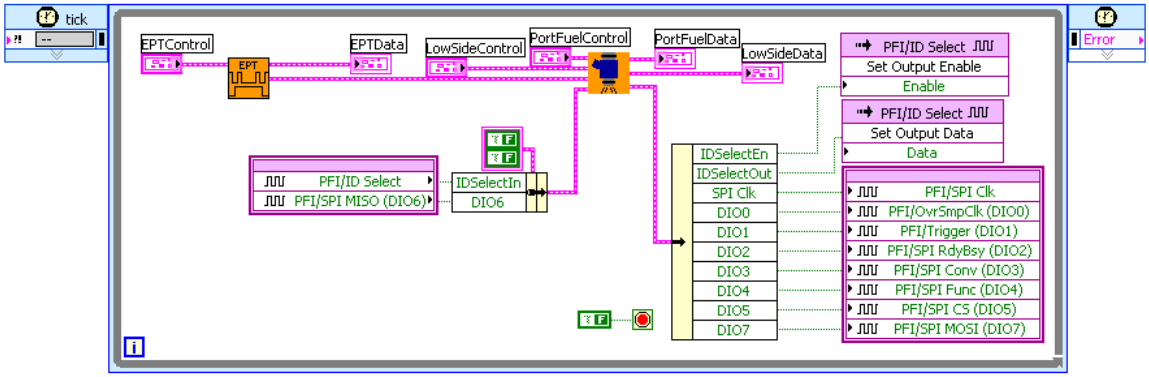


Figure 8. Example block diagram implementation of PFI VIs.

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

PFILSPinOutput (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 9a 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 9b 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.

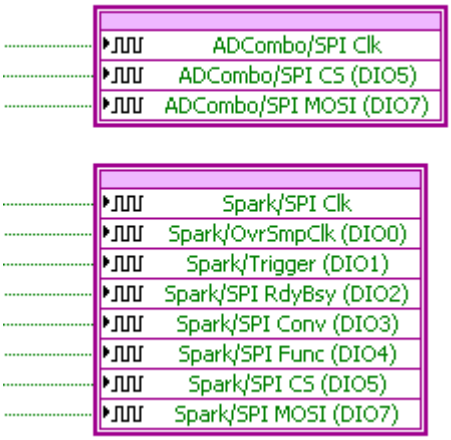


Figure 9a. Representative FPGA output nodes for ADCombo and Spark modules with correct output item selection.

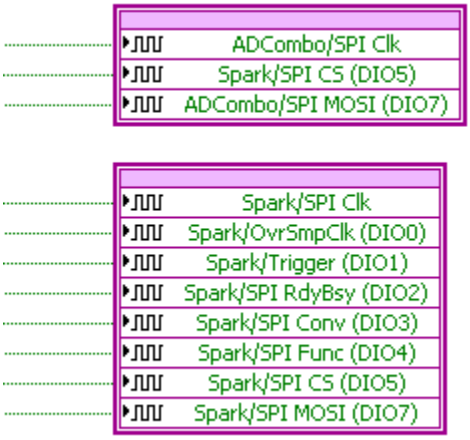


Figure 9b. Representative FPGA output nodes for ADCombo and Spark modules with incorrect output item selection within the ADCombo output node. The Spark (DIO5) output is selected in multiple nodes and therefore being driven by multiple sources. This will cause strange behavior or damage to the spark 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.

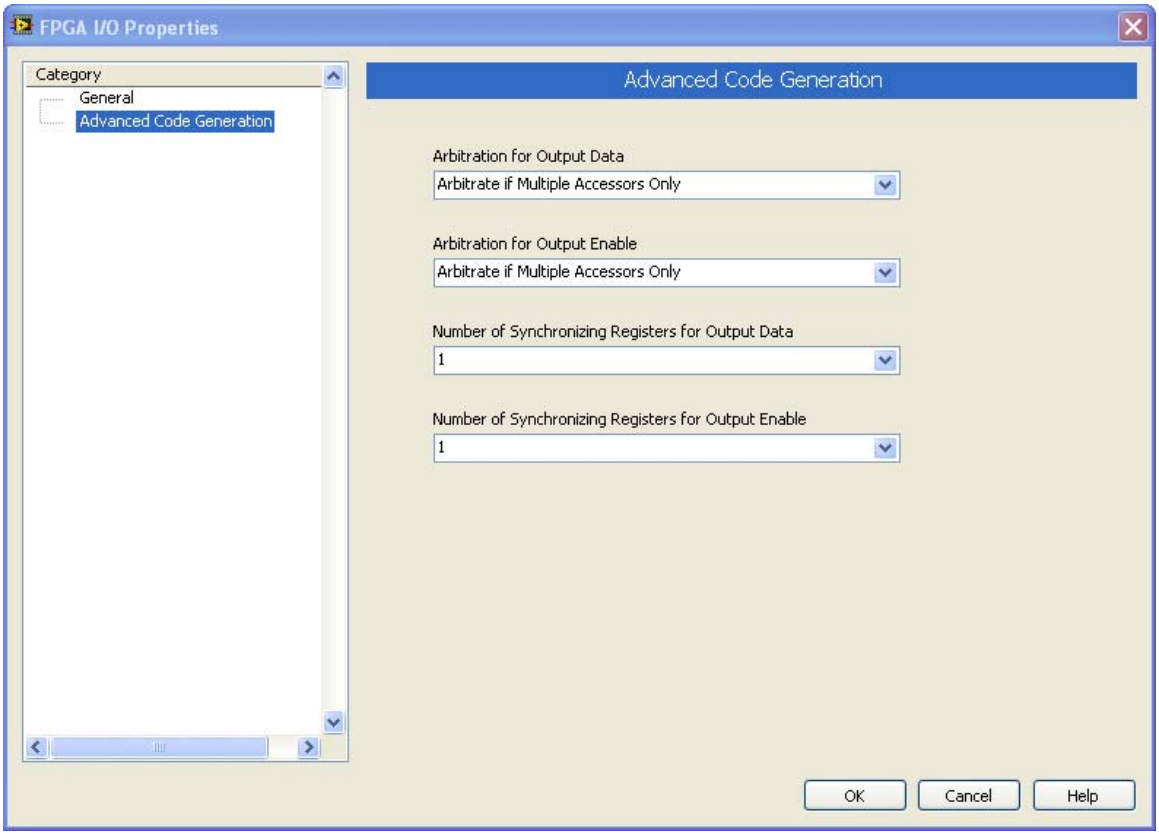
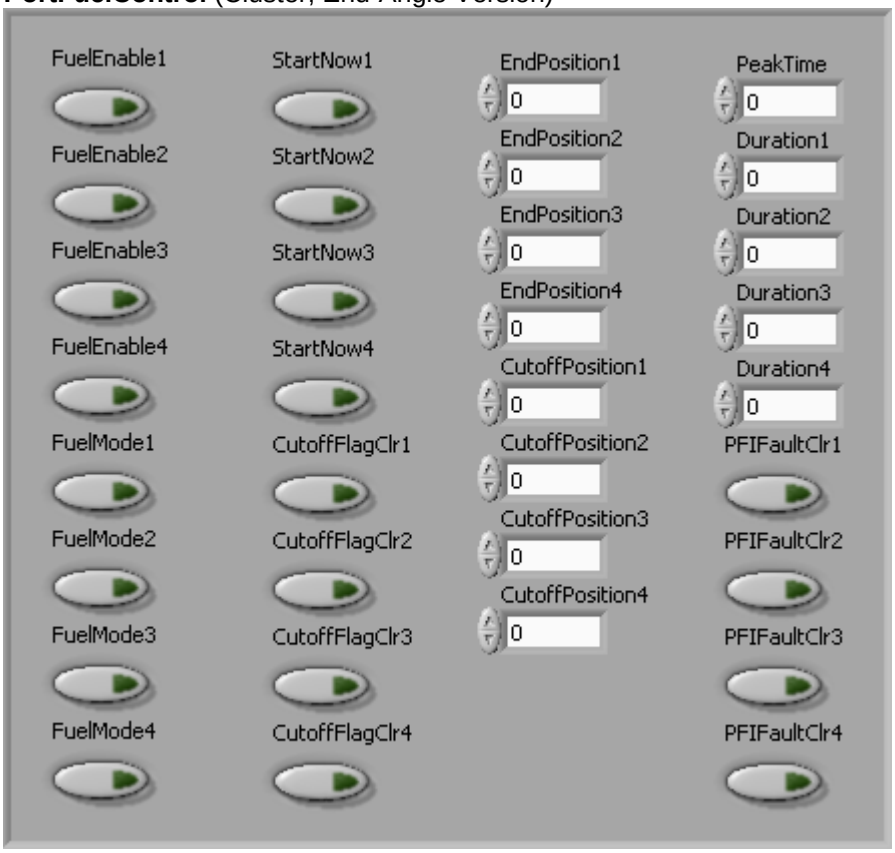


Figure 10. FPGA I/O Properties dialog configuration for cRIO modules.

PortFuelControl (Cluster, End-Angle Version)



FuelEnable (boolean): When TRUE, the fuel command is enabled. When FALSE (default), the fuel command is disabled.

FuelMode (boolean): When FALSE (default), fuel pulses are generated according to StartPosition or EndPosition, depending on the VI used. This is the normal mode of operation. When TRUE, a single fuel pulse is generated immediately upon receiving a TRUE StartNow input. The pulse will be generated with the requested Duration. StartNow must be reset to FALSE in order to generate another pulse via this method.

StartNow (boolean): See FuelMode.

CutoffFlagClr (boolean): When TRUE, the CutoffFlag within the PortFuelData cluster is cleared.

StartPosition (uint16): This control will be available when using the pfi_vt_sa_revx.vi. Fuel pulses are generated with a leading edge coinciding with StartPosition. The length of the pulse will be according to Duration. The units of StartPosition are CAT.

EndPosition (uint16): This control will be available when using the pfi_vt_ea_revx.vi. Fuel pulses are generated with a trailing edge coinciding with EndPosition. The length of the pulse will be according to Duration. The leading edge will be determined by the requested Duration. When EndPosition is used, Duration will always take precedence over EndPosition in the presence of engine speed fluctuations. If the engine speed increases after the fuel pulse has started, then the trailing edge will occur after EndPosition in order to ensure that Duration is achieved. Likewise, if the engine speed decreases after the fuel pulse has started, then the trailing edge will occur before EndPosition. The units of EndPosition are CAT.

CutoffPosition (uint16): All fuel pulse activity is "Cutoff" at CutoffPosition and reset for the next cycle. CutoffFlag is also asserted and can be cleared by setting CutoffFlagClr to TRUE.

CutoffPosition must always be at least 45 crank angle degrees (CAD) after StartPosition or EndPosition, depending on the VI used. If this minimum spacing is not maintained, then fuel commands will be generated with incorrect timing. The units of CutoffPosition are CAT.

Drivven provides a utility VI which can be implemented at the LabVIEW RT level for performing the conversion of injection timing and cutoff timing values from TDC offsets 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 11.

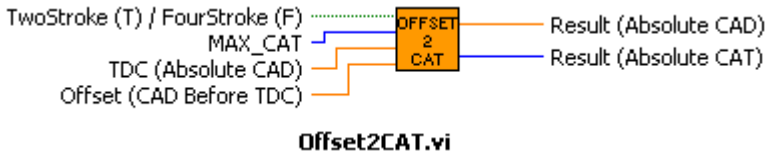


Figure 11. Offset to CAT conversion VI.

PeakTime (uint8): Determines the length of time that the driver circuit will use peak current as the current control threshold. PeakTime is entered as a uint8 value in terms of 4 MHz clock ticks divided by 64. Therefore, the maximum value entered for PeakTime of 255 equates to an effective value of 16320. At 4 MHz this corresponds to a maximum delivered peak time of 4.08 msec. The following equation applies:

PeakTime(uint8 ticks) = PeakTime(msec) * 62.5

Drivven provides a utility VI which can be implemented at the LabVIEW RT level for performing this calculation. The VI named peaktime2ticks.vi can be used to convert PeakTime in milliseconds to uint8 ticks. This VI icon is shown in Figure 12.

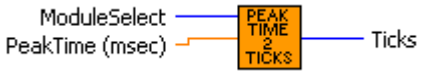


Figure 12. PeakTime to Ticks conversion VI.

Duration (uint32): Determines the length of the fuel command delivered to the driver circuit. Duration is entered in terms of 40 MHz clock ticks and is internally limited to 24 bits. Values larger than 24 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 13.

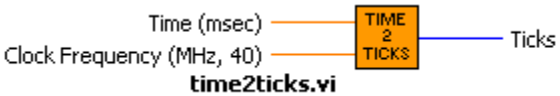


Figure 13. Time to Ticks conversion VI.

PFIFaultClr (boolean): When TRUE, clears the corresponding port fuel injector short circuit fault flag.

Position Conversion and Notes:

StartPosition, EndPosition 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)

StartPosition, EndPosition 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 PFI 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 11. 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 PFI VIs provide features that ensure the best possible fuel command delivery, even while the CPU makes modifications to StartPosition, EndPosition 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. If Duration is modified after the end of the fuel pulse, but before CutoffPosition, to a value less than the already accumulated duration for the cycle, then no action is taken because the fuel pulse has completed. The new duration will take effect on the next pulse.

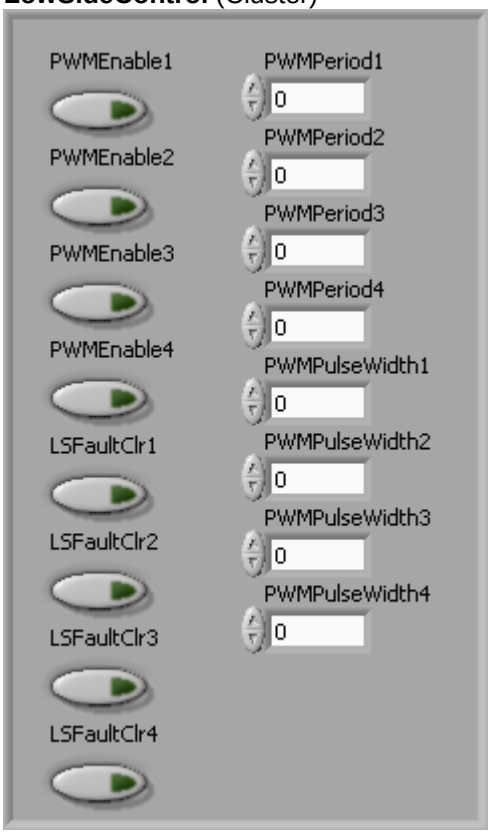
Modifications to StartPosition or EndPosition:
StartPosition and EndPosition can be modified at any time. However, the value must not be advanced by more than 45 CAD within a single engine cycle. This value is referred to as the History Window. The PFI VI continually checks the requested Start/EndPosition with respect to the current crank position. If the Start/EndPosition is modified by the CPU to a position in the past, the PFI 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 StartPosition of 200 Absolute CAD (ACAD). Let's also assume that the CurrentPosition of the EPT VI is 190 ACAD when the CPU modifies StartPosition to 180 ACAD, which is in the recent past by 10 CAD. Since this is less than the 45 CAD History Window, then the PFI 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 StartPosition of 200 ACAD. Let's also assume that the CurrentPosition of the EPT VI is 190 ACAD when the CPU modifies StartPosition to 120 ACAD, which is in the far past by 80 CAD. Since this is greater than the 45 CAD History Window, the PFI 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 StartPosition or EndPosition, depending on the VI used. If this minimum spacing is not maintained, then fuel commands will be generated with incorrect timing.

LowSideControl (Cluster)



PWMEnable (boolean): When TRUE, the PWM controller is enabled. When FALSE (default), the PWM controller is disabled.

PWMPeriod (uint32): The time period between leading edges of low-side switch closure. PWMPeriod is entered in terms of 4 MHz clock ticks and is internally limited to 24 bits. This provide a maximum period of 4.19 seconds or a frequency of 0.24 Hz, and a resolution of 250 nsec. Values larger than 24 bits will roll over from zero.

PWMPulseWidth (uint32): The time during each PWMPeriod in which the low-side switch is closed. The low-side switch is closed from the beginning of PWMPeriod until PWMPulseWidth expires. PWMPulseWidth is entered in terms of 4 MHz clock ticks and is internally limited to 24 bits. While PWMPulseWidth is 0 the low-side switch will remain open. While PWMPulseWidth is greater than or equal to PWMPeriod, the low-side switch will remain closed. Values larger than 24 bits will roll over from zero.

Drivven provides a utility VI, named lowside2ticks.vi, which can be implemented at the LabVIEW RT level for performing the conversion from frequency in hertz and duty cycle in percent to PWMPeriod ticks and PWMPulseWidth ticks. This VI icon is shown in Figure 14.

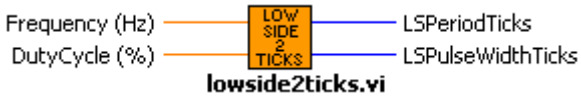


Figure 14. Lowside PWM conversion VI.

LSFaultClr (boolean): When TRUE, clears the corresponding low-side driver short circuit fault flag.

PWM Controller Notes:
The control values for the independent PWM controllers can be updated at any time by the CPU. The FPGA updates the control values at 125 usec intervals. The PWM controllers ensure glitch free operation of the PWM signal to the low-side drivers. For example, if a PWM controller pulse width is updated to a larger value immediately after PWMPulseWidth expires, the PWM command to the driver will not be asserted again until the next period.

PortFuelData (Cluster)

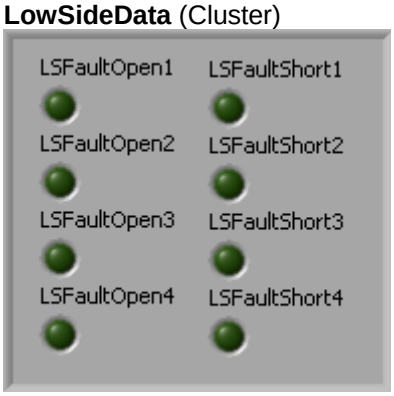


CutoffFlag (boolean): Set to TRUE when CurrentPosition from the EPT VI equals CutoffPosition. CutoffFlag can be used by the CPU as interrupt flags if necessary to perform calculations synchronous with crank position. CutoffFlag is cleared by setting CutoffFlagClr TRUE.

PFIFaultOpen (boolean): Set to TRUE when an open circuit is detected for the port fuel injector. The flag is cleared when the condition is removed.

PFIFaultShort (boolean): Set to TRUE when a short circuit is detected for the port fuel injector. A short circuit automatically disables the channel until the flag is cleared by the corresponding PFIFaultClr.

LowSideData (Cluster)



LSFaultOpen (boolean): Set to TRUE when an open circuit is detected for the low-side driver. The flag is cleared when the condition is removed.

LSFaultShort (boolean): Set to TRUE when a short circuit is detected for the low-side driver. A short circuit automatically disables the channel until the flag is cleared by the corresponding LSFaultClr.

FuelSparkSupervisor (Cluster)
This cluster input must be wired directly from the FuelSparkSupervisor cluster output of a Drivven EPT VI.

Examples

The following screen capture in Figure 15 shows a LabVIEW FPGA block diagram demonstrating the interface between EPT VIs and fuel/spark VIs. 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.

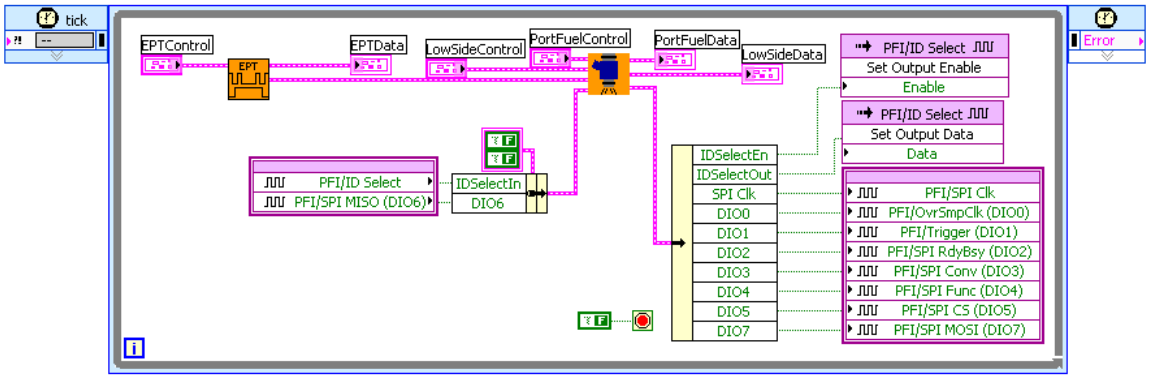


Figure 15. LabVIEW FPGA Block diagram example.

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP