Throttle Driver Module Kit User's Manual
D000017 Rev B
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

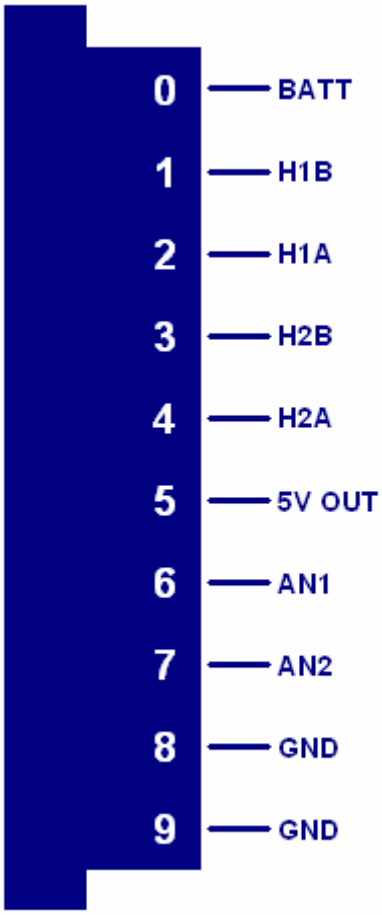
The Electronic Throttle Driver Module Kit provides a CompactRIO (cRIO) module for driving passenger car electronic throttle bodies up to 70mm in diameter. A typical example of electronic throttle bodies which this module is capable of driving is the Bosch DV-E5 series which range from 32mm to 68mm in diameter. However, the module is capable of driving most other electronic throttle bodies within that size range.

The Electronic Throttle Driver Module Kit includes a LabVIEW FPGA VI for controlling two H-Bridge driver channels independently. Also provided are a set of RT VIs which allow the user to calibrate the throttle control algorithm in engineering units. The features included are listed below:

Features:

- 2-Ch. H-Bridge Drivers for dual electronic throttle control
- 2-Ch. analog input for throttle position feedback
- Short circuit and over-temperature protection with fault reporting
- Battery voltage, current sensing and module temperature measurements
- LabVIEW FPGA VI for h-bridge control and interface
- LabVIEW RT VI for electronic throttle position control
- External power supply of 6-32V

Pinout



Hardware

The Electronic Throttle Driver Module Kit provides two H-Bridge drivers and analog position feedback in a National Instruments CompactRIO module.

Powering the Module

The Electronic Throttle 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.

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.

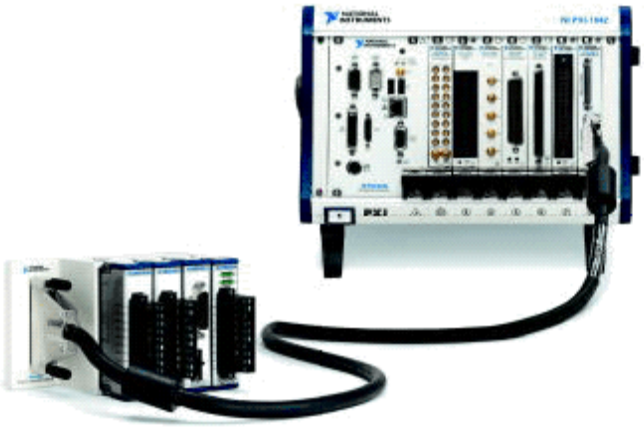


Figure 1b. PXI platform compatible with Drivven CompactRIO modules.

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.

H-Bridge Drivers

The Electronic Throttle Driver Module Kit provides a CompactRIO (cRIO) module for driving passenger car electronic throttle bodies up to 70mm in diameter. A typical example of electronic throttle bodies which this module is capable of driving is the Bosch DV-E5 series which range from 32mm to 68mm in diameter. However, the module is capable of driving most other electronic throttle bodies within that size range.

Each h-bridge circuit is capable of driving 3A continuously and 6A peak. It provides current sensing feedback as well as over-current and over-temp protection. In the case of a short circuit where 10A is exceeded, or a temperature of 140 C is exceeded, a fault flag will be generated and the circuit will shutdown until the fault condition is removed. However, Drivven recommends monitoring the current and temperature values and disabling the circuit programmatically if current exceeds 6A or the internal module temperature exceeds 85 C.

Each h-bridge circuit automatically eliminates “shoot-through” current and provides internal clamp diodes for inductive loads.

Analog Inputs

The Electronic Throttle Driver Module provides two external analog inputs for accepting 0-5V signals. The primary purpose of these inputs is for measuring potentiometer voltages. A regulated 5V output and ground terminal is provided for powering the potentiometer(s) of an electronic throttle body.

Other analog signals are also measured internal to the module and reported by the supplied VIs. All measured analog signals are listed below:

- Battery Voltage (V)
- H-Bridge 1 Current (A)
- H-Bridge 2 Current (A)
- H-Bridge 1 Fault Line (reported with boolean, T=Fault, F=No Fault)
- H-Bridge 2 Fault Line (reported with boolean, T=Fault, F=No Fault)
- External Analog Input 1 (0-5V, Screw Terminal 6)
- External Analog Input 2 (0-5V, Screw Terminal 7)
- Internal Module Temperature

All eight analog inputs are sampled with a single multiplexed A/D converter at an aggregate rate of 17.84 KHz. Therefore each analog signal is sampled at 2.23 KHz. The A/D result can be used directly at the FPGA level in ADC counts, or at the RT level in engineering units.

The external analog inputs are protected from -6V to +12V inputs.

The external analog inputs are independent from throttle control algorithms, and may be used in any manner the user deems necessary. The intended purpose of the external analog inputs is for the convenience of connecting all wires of an electronic throttle directly to the module. Then the user may programmatically use the analog values for throttle control feedback. Other analog inputs from another module may also be used to provide throttle position. Most electronic throttle bodies have redundant position signals. When using only one electronic throttle, both position signals may be connected to the module and redundant throttle position is available. However,

when two electronic throttle bodies are used, only one position signal from each module may be connected to the module, and redundant throttle position is not available. However, the additional position signals may be connected to other analog inputs from other modules, making redundant throttle position available for both throttles.

Figure 2 and figure 3 below show the typical connections for controlling single or dual throttles, respectively. Table 1 below shows the module terminal connections to a standard Bosch DV-E5 electronic throttle body.

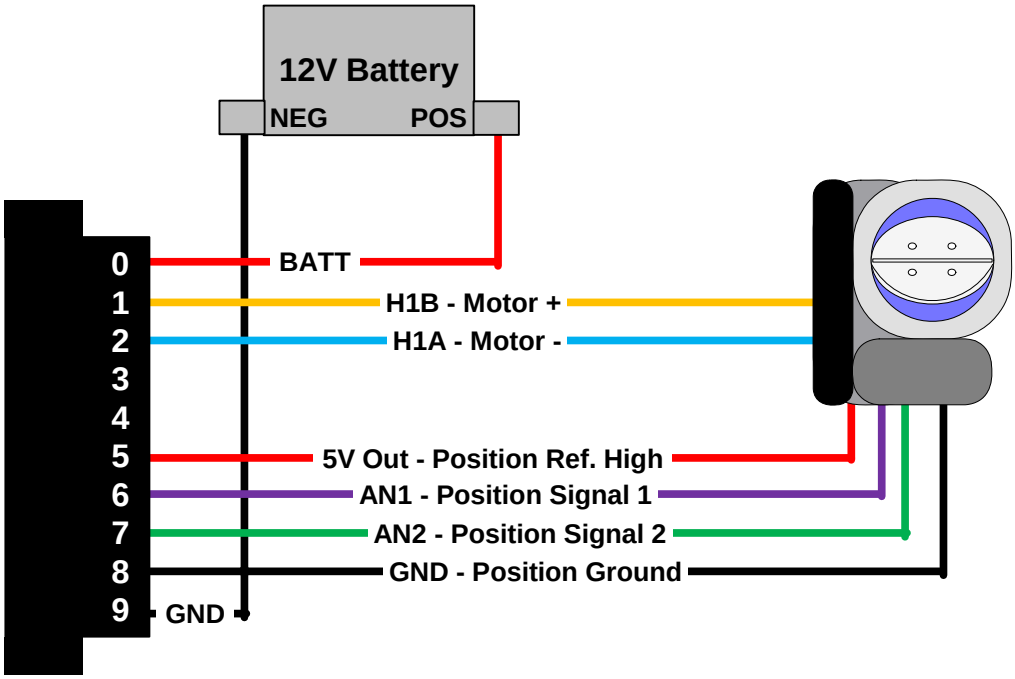


Figure 2. Module terminal connections to a single electronic throttle body.

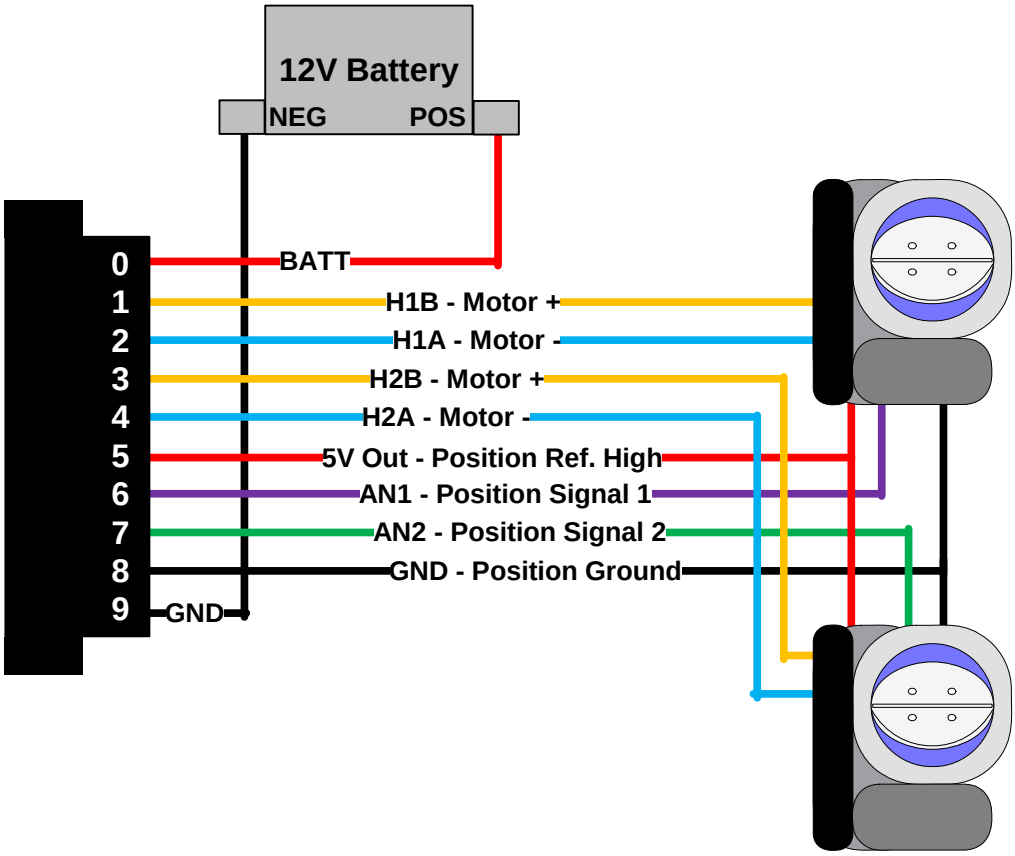


Figure 3. Module terminal connections to dual electronic throttle bodies.

Table 1. Connection table specific for standard Bosch DV-E5 electronic throttle bodies

DV-E5 Pin	Description	Module Terminal (Throttle1 / Throttle2)	Description
1	Motor -	T2 / T4	H1A / H2A
2	Potentiometer Ground	T8	GND
3	Potentiometer Reference	T5	5V Out
4	Motor +	T1 / T3	H1B / H2B
5	Potentiometer 1 Signal	T7	AN2
6	Potentiometer 2 Signal	T6	AN1

Standard Bosch DV-E5 electronic throttle bodies are available through Drivven according to the following part numbers list in table 2. Prices and availability vary.

Table 2. Standard Bosch DV-E5 throttle bodies available through Drivven

Throttle Body Diameter (mm)	Bosch Part Number
32	0 280 750 148
40	0 280 750 149
54	0 280 750 150
60	0 280 750 151
68	0 280 750 152

The following connector parts for standard Bosch throttle bodies are available through Drivven, According to the following part numbers list in Table 3. Prices and availability vary.

Table 3. Connector parts required for standard Bosch DV-E5 throttle bodies

Connector Part Description	AMP Part Number
Connector Housing	1-967616-1
Crimp Socket Contact (Tin)	965906-1
Wire Seal	967067-1

Software

The Electronic Throttle Driver Module Kit is provided with both a LabVIEW FPGA VI for interfacing directly to the module and a LabVIEW RT VI for interfacing with the FPGA VI and controlling throttle position.

Figure 4 shows the icon which represents FPGA throttle_revx.vi.

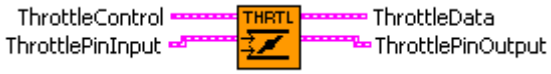


Figure 4. FPGA VI icon with leads.

Figure 5 shows the icon which represents throttle_rt_data_convert_revx.vi.

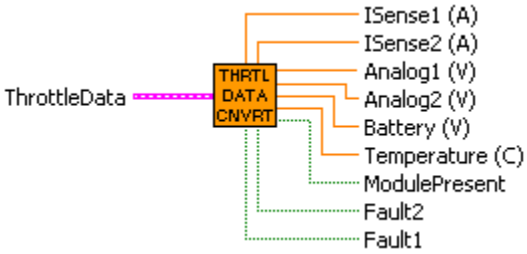


Figure 5. Throttle Data RT VI icon with leads.

Figure 6 shows the icon which represents throttle_rt_control.vi.

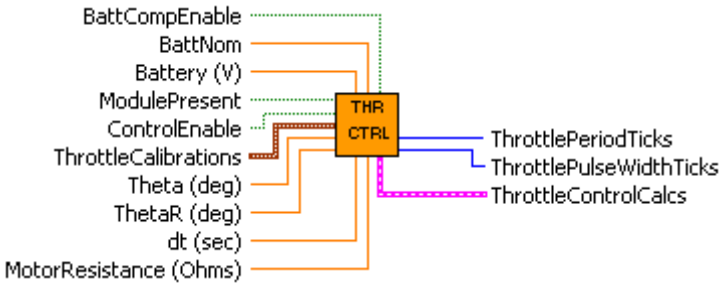


Figure 6. Throttle Control RT VI icon with leads.

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.

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 may be 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.

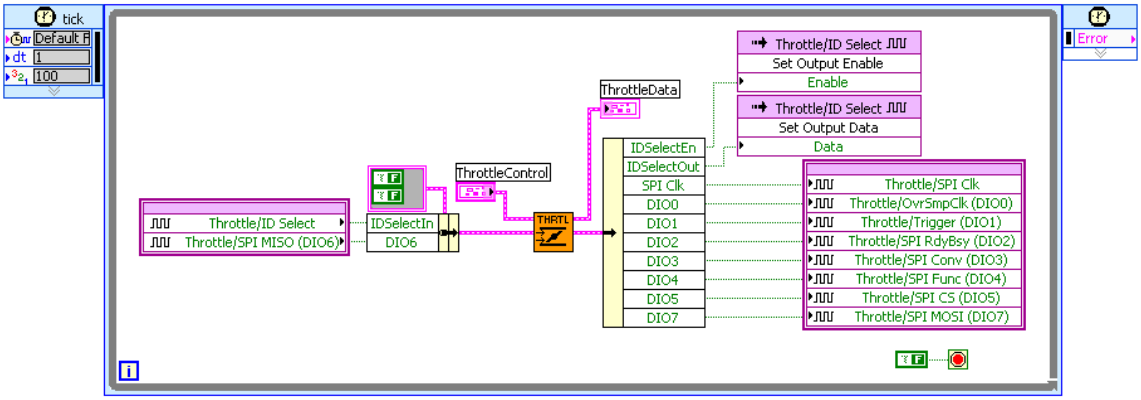


Figure 7. Example FPGA block diagram implementation of throttle_revx.vi.

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

ThrottlePinOutput (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 8a 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 8b 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.

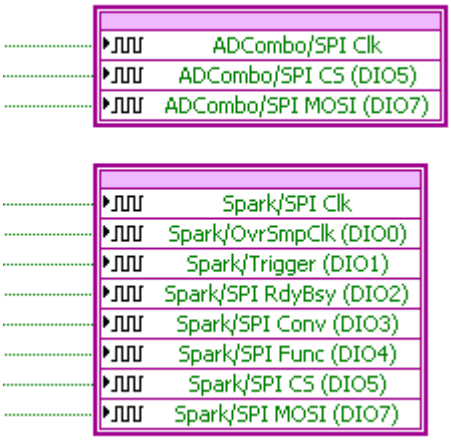


Figure 8a. Representative FPGA output nodes for ADCombo and Spark modules with correct output item selection.

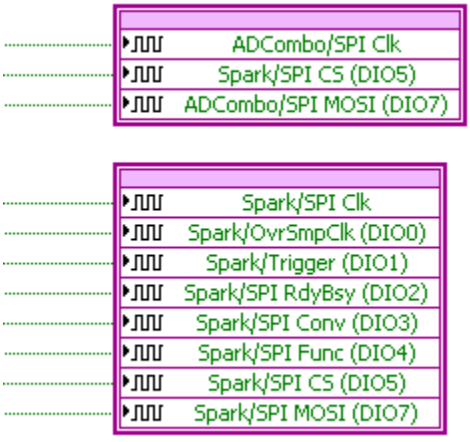


Figure 8b. Representative FPGA output nodes for ADCombo and Spark modules with incorrect output item selection within the ADCombo output node. The Spark (DIO5) output is selected in multiple nodes and therefore being driven by multiple sources. This will cause strange behavior or damage to the spark 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.

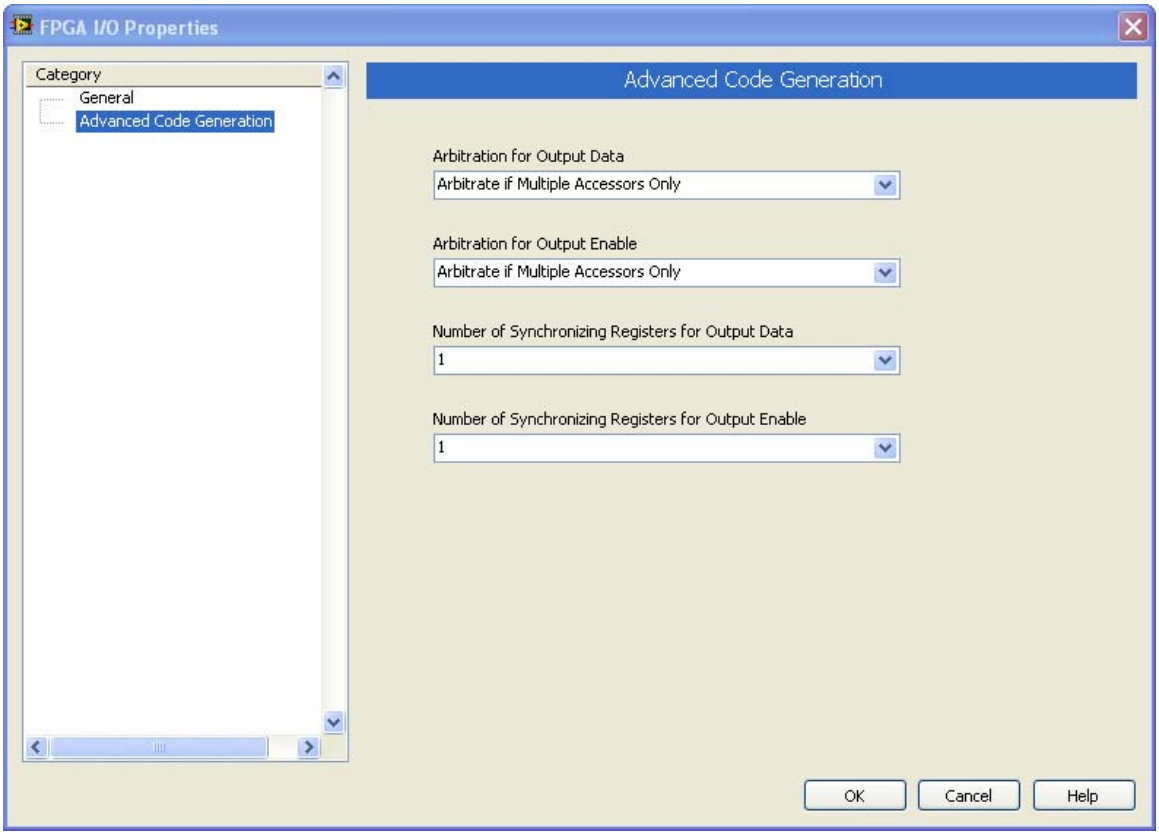
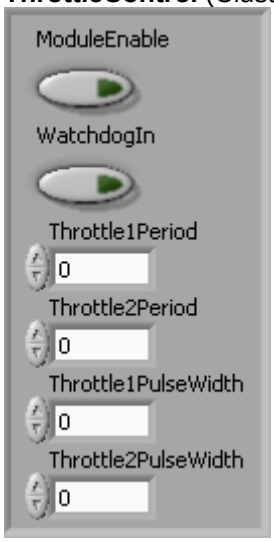


Figure 9. FPGA I/O Properties dialog configuration for cRIO modules.

ThrottleControl (Cluster)



ModuleEnable (boolean): If a throttle 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 ThrottleData cluster will be set to TRUE.

WatchdogIn (boolean): WatchdogIn must be toggled at a rate greater than or equal to 10Hz. 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.

Throttle1Period (uint16): The time period between leading edges of the PWM pulse train to h-bridge circuit 1. Throttle1Period is entered in terms of 4 MHz clock ticks. This provide a maximum period of 8.192 milliseconds or a minimum frequency of 122 Hz, and a resolution of 250 nsec.

Throttle1PulseWidth (int16): The time during each Throttle1Period in which the PWM pulse train to h-bridge circuit 1 is active. This value is signed, and represents direction of current flow through the h-bridge circuit. A positive value represents positive current flowing from terminal H1B to H1A. Throttle1PulseWidth is entered in terms of 4 MHz clock ticks. While Throttle1PulseWidth is 0, h-bridge circuit 1 will remain inactive. While Throttle1PulseWidth is greater than or equal to Throttle1Period, h-bridge circuit 1 will remain fully active. This condition should be avoided. Otherwise an over-current or over-temp fault will result.

Throttle2Period (uint16): The time period between leading edges of the PWM pulse train to h-bridge circuit 2. Throttle2Period is entered in terms of 4 MHz clock ticks. This provides a maximum period of 8.192 milliseconds or a minimum frequency of 122 Hz, and a resolution of 250 nsec.

Throttle2PulseWidth (int16): The time during each Throttle2Period in which the PWM pulse train to h-bridge circuit 2 is active. This value is signed, and represents direction of current flow through the h-bridge circuit. A positive value represents positive current flowing from terminal H2B to H2A. Throttle2PulseWidth is entered in terms of 4 MHz clock ticks. While Throttle2PulseWidth is 0, h-bridge circuit 2 will remain inactive. While Throttle2PulseWidth is greater than or equal to Throttle2Period, h-bridge circuit 2 will remain fully active. This condition should be avoided. Otherwise an over-current or over-temp fault will result.

Drivven provides a VI named throttle2ticks.vi, which can be implemented at the LabVIEW RT level for performing the conversion from frequency in hertz and signed duty cycle in percent to ThrottlePeriod ticks and ThrottlePulseWidth ticks. This VI icon is shown in Figure 10.

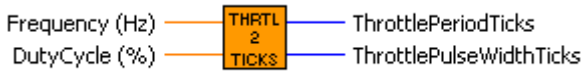


Figure 10. Throttle PWM conversion VI.

This VI is embedded within the throttle_rt_control.vi and does not need to be used directly by the programmer. Drivven recommends that the programmer always use the throttle_rt_control.vi for interfacing to the FPGA level.

ThrottleData (Cluster)
The ThrottleData Cluster should be terminated with an indicator cluster and made available as a complete cluster for interfacing to the RT VI. No FPGA code interface is required with any of the members of this cluster.

RT VI Implementation

The throttle_rt_control.vi (RT VI) and throttle_rt_data_convert_revx.vi must be placed within a while loop or timed loop and executed at a recommended rate of 200 Hz. This rate provides good throttle control capability. A reference must be opened to a LabVIEW FPGA application which implements the throttle_revx.vi. Also, FPGA read/write register functions must be placed within the RT loop to gain access to the ThrottleControl and ThrottleData clusters of the FPGA throttle_revx.vi. This is shown in Figure 11.

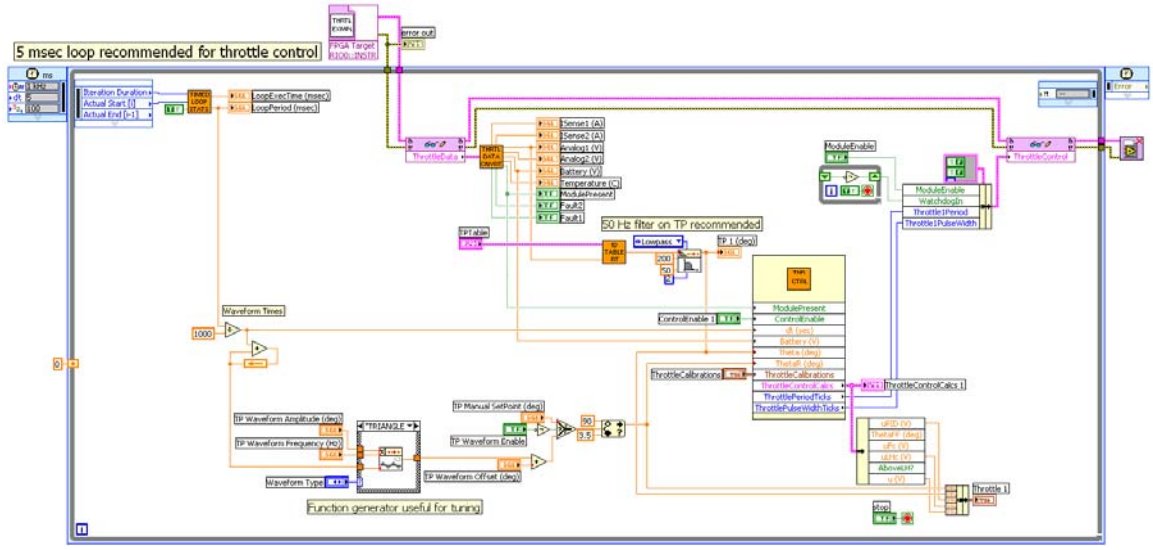
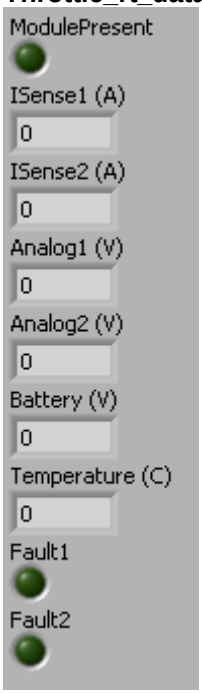


Figure 11. Example RT implementation of single throttle_rt_control.vi.

Throttle Data Conversion VI

ThrottleData (Cluster)
The ThrottleData Cluster input should be fed with the cluster wire from the ThrottleData output cluster of the FPGA. No other RT code interface is required with any of the members of this cluster.

Throttle_rt_data_convert_revx.vi output parameters



ModulePresent (boolean): When TRUE, then software has properly detected a Throttle Driver Module and can begin operation. When FALSE, then software has not yet detected the presence of a Throttle Driver Module. In order to be detected, the driver module must be properly inserted in its assigned 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 begin operating. 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 begin operating. ModulePresent should be wired directly to the ModulePresent input of the throttle_rt_control.vi.

ISense1 (A) (SGL): The load current of h-bridge circuit 1 in amps.

ISense2 (A) (SGL): The load current of h-bridge circuit 2 in amps.

Analog1 (V) (SGL): The raw voltage of external analog input 1.

Analog2 (V) (SGL): The raw voltage of external analog input 2.

Battery (V) (SGL): The battery voltage supplied to the module.

Temperature (C) (SGL): The internal module temperature.

Fault1 (boolean): When TRUE, then an over-current or over-temperature fault has occurred with h-bridge circuit 1 and operation will be interrupted until the fault is removed.

Fault2 (boolean): When TRUE, then an over-current or over-temperature fault has occurred with h-bridge circuit 2 and operation will be interrupted until the fault is removed.

The values given by the throttle_rt_data_convert_revx.vi may be used for monitoring or control

purposes. If the external analog inputs are used for position control feedback, then they must be converted to throttle angle by means of a transfer function or lookup table. Drivven provides a lookup table function in the rt_vi directory provided to all customers. The throttle angle in degrees should be input to the Theta (deg) input of the throttle_rt_control.vi.

It is recommended that the programmer use the module temperature to prevent the module operation at internal temperatures above 85C. Typical operating temperatures will be between 30C and 50C.

Throttle Control VI

Drivven provides a flexible throttle position control algorithm with the Throttle Driver Module Kit. Please follow the throttle control example provided for implementing throttle control in your engine control application. Drivven recommends performing throttle control at a rate of 200 Hz. Even if your engine control algorithms require execution at another rate, a separate timed loop can be created to implement throttle control. The throttle control VI uses position values in terms of degrees. This document does not go into detail about the procedures for tuning a typical PID loop. There are many texts available which cover that topic. It is expected that the user of this module kit be familiar with PID control concepts.

The throttle control algorithm calculates a final voltage to be applied to the throttle body DC motor. The control voltage may be compensated for actual battery voltage, according to BattCompEnable. The compensated voltage is converted to a signed PWM duty cycle at a fixed 500 Hz. The results which exit the throttle control VI are in terms of clock ticks to be wired to the FPGA period and pulse width parameters.

The throttle control algorithm involves 4 major functions:

1.) The angle setpoint, ThetaR is compensated by a user defined lead and lag time.
2.) A proportional, integral and derivative action is calculated based on two sets of PID gains, above or below the default limp-home region. The reason for two sets of gains is because electronic throttle bodies typically have stiffer spring return rates applied to angles below the limp-home region.
3.) A limp-home compensation value is added to the PID value to assist with travel through the limp-home region. This compensation can minimize the flat spot often found as the throttle plate moves through the limp-home region.
4.) A stiction compensation value is added to the PID value to assist with small error control.

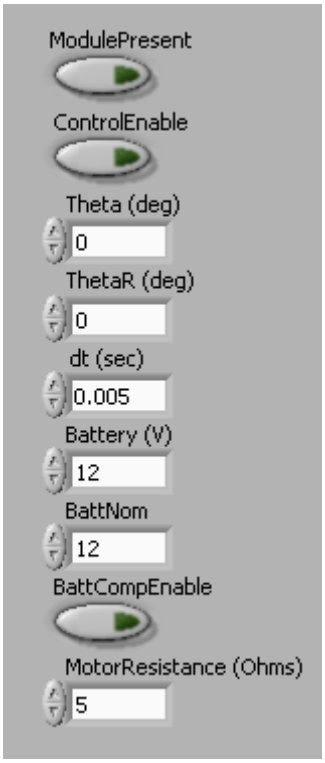
The first calibration which the user should tune is throttle angle versus sensor voltage. A linear equation can be used, as well as a two-point 1D lookup table. Manually close the throttle plate completely to determine the minimum sensor voltage. If the radii of the throttle body opening and the throttle plate are measured, then the minimum throttle angle can be calculated by using an inverse cosine calculation, or by approximating the small angle with sine. Manually open the throttle plate to wide-open-throttle, noting the maximum sensor voltage. Assume the wide-open-throttle angle to be 90 degrees. After calibrating for position versus voltage, enter the value for ThetaLH (default limp-home position).

It is recommended that a filter be implemented on throttle position. A second order, lowpass Butterworth filter with a cuttoff frequency of 50Hz is suggested. The filter is most effective during small-error control.

It is recommended that the user begin control calibration with zero for all Lead/Lag, PID and compensation calibration parameters. Begin tuning PID gains for angles above ThetaLH. Then introduce TLead and TLag values as necessary. Then begin tuning PID gains for angles below ThetaLH. ULH and US compensation should be tuned last. The default values saved for the example application were derived for a 68mm Bosch DV-E5 throttle body.

Warning: ThetaR should not be allowed to exceed the minimum and maximum angles of the throttle. This will force the controller to drive more current than the throttle or module can sustain. The force exerted may also mechanically damage the throttle body.

Throttle Control Parameters (Individual)



ModulePresent (boolean): Should be wired directly from the ModulePresent output of the throttle_rt_data_convert_revx.vi.

ControlEnable (boolean): When TRUE, the VI will generate active output values for the ThrottlePulseWidthTicks output. When FALSE, the VI will write 0 to the ThrottlePulseWidthTicks output.

Theta (deg) (SGL): The latest feedback throttle angle (degrees) of the throttle body under control.

ThetaR (deg) (SGL): The requested angle setpoint (degrees) of the throttle body under control.

dt (sec) (SGL): The period, in seconds, since the last call to the throttle control function. This value can be updated in real-time, or set to a constant.

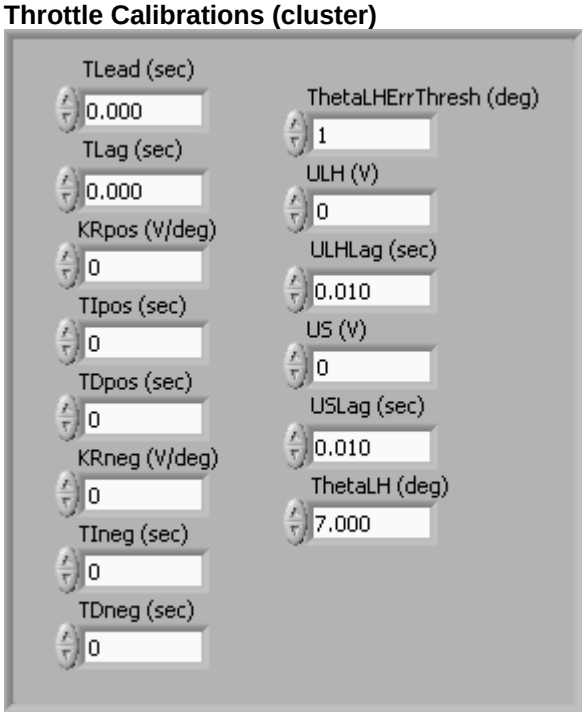
Battery (V) (SGL): The battery voltage supplied to the module, as calculated by the throttle_rt_data_convert_revx.vi.

BattNom (V) (SGL): The nominal battery voltage supplied to the module. Most automotive systems operate with 12V. This would be the nominal battery voltage. The actual voltage may fluctuate.

BattCompEnable (boolean): When TRUE, the VI will compensate the throttle output for the deviation of Battery compared to BattNom. When FALSE, the throttle output will not be compensated. Battery compensation has a small effect.

MotorResistance (Ohms) (SGL): The resistance of the motor winding must be entered here. A multi-meter can be used to measure this. This value is used to determine output ranges.

Throttle Calibrations (cluster)



TLead (sec) (SGL): Adjusts the lead compensation on the ThetaR value. A lead/lag compensation function is internally inserted in the ThetaR signal to the internal PID function. TLead can be increased to intensify the change in ThetaR.

TLag (sec) (SGL): Adjusts the lag compensation on the ThetaR value. A lead/lag compensation function is internally inserted in the ThetaR signal to the internal PID function. TLag can be increased to slow the change in ThetaR.

KRpos (V/deg) (SGL): Proportional gain for the throttle control PID function while Theta is greater than ThetaLH.

TIpos (sec) (SGL): Integral time for the throttle control PID function while Theta is greater than ThetaLH. Smaller times intensify the integral action. When TI is set to zero, the integral term is disabled.

TDpos (sec) (SGL): Derivative time for the throttle control PID function while Theta is greater than ThetaLH. Larger times intensify the derivative action. When TD is set to zero, the derivative term is disabled.

KRneg (V/deg) (SGL): Proportional gain for the throttle control PID function while Theta is less than or equal to ThetaLH.

TIneg (sec) (SGL): Integral time for the throttle control PID function while Theta is less than or equal to ThetaLH. Smaller times intensify the integral action. When TI is set to zero, the integral term is disabled.

TDneg (sec) (SGL): Derivative time for the throttle control PID function while Theta is less than or equal to ThetaLH. Larger times intensify the derivative action. When TD is set to zero, the derivative term is disabled.

ThetaLHErrThresh (deg) (SGL): Limp-home compensation (uLHc) is updated when ThetaR is within ThetaLHErrThresh degrees of ThetaLH (above or below).

ULH (V) (SGL): Voltage added or subtracted to the PID output when ThetaR is in the vicinity of ThetaLH. As Theta moves through the limp-home region, ULH is updated to a positive or negative value (negative going down, positive going up) and added to the PID output in order to assist throttle control through the region.

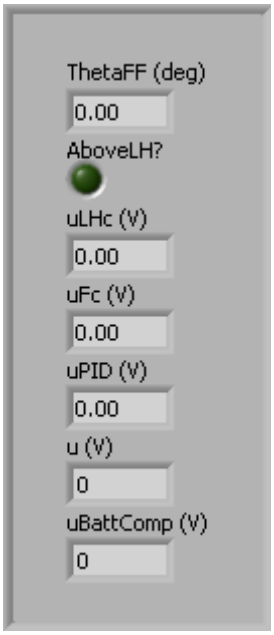
ULHLag (sec) (SGL): Lag filter time applied to the limp-home compensation. A lag time prevents uLHc from reversing directions too fast.

US (V) (SGL): Voltage added or subtracted to the PID output when Theta is outside the vicinity of ThetaLH. US is a stiction compensation value. This parameter assists throttle control during small Theta errors to overcome stiction. Parameter uFc is updated to a positive or negative value according to the sign of ThetaR-Theta error.

USLag (sec) (SGL): Lag filter time applied to the stiction compensation. A lag time prevents uFc from reversing directions too fast.

ThetaLH (deg) (SGL): Limp-home angle. This is the default throttle position which results from no power being applied to the motor.

Throttle Control Calculated Parameters (Cluster) (For Monitoring Purposes Only)



ThetaFF (deg) (SGL): Value calculated by the Lead/Lag compensator function on ThetaR.

AboveLH? (boolean): Indicates whether Theta is above ThetaLH.

uLHc (V) (SGL): Limp-home region compensation added to PID calculation.

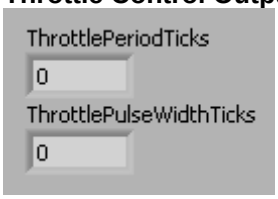
uFc (V) (SGL): Stiction compensation added to PID calculation.

uPID (V) (SGL): PID control result.

u (V) (SGL): Control value calculated from the sum of uPID, uLHc and uFc.

uBattComp (V) (SGL): Final control value compensated for actual battery voltage.

Throttle Control Output Parameters (Individual)



ThrottlePeriodTicks (uint16): Number of 4 MHz clock ticks required to achieve 500 Hz PWM frequency to the h-bridge circuit.

ThrottlePulseWidthTicks (int16): Signed number of 4 MHz clock ticks required to achieve PWM duty cycle calculated by the throttle_rt_control.vi.

Warning: ThetaR should not be allowed to exceed the minimum and maximum angles of the throttle. This will force the controller to drive more current than the throttle or module can sustain. The force exerted may also mechanically damage the throttle body.

Examples

The following screen captures show the implementation of the throttle_revx.vi and the throttle_rt_control.vi within the LabVIEW FPGA and LabVIEW RT environments, respectively. Both implementations are suggested for using the Throttle Driver Module Kit. You will find these implementations within the software bundle for the kit. The example project file is named ThrottleExample.lvproj and the top level FPGA VI is named ThrottleExample.vi. ThrottleExample.vi implements throttle_revx.vi. You will also find top level RT VIs named ThrottleExampleRT_1Chan.vi and ThrottleExampleRT_2Chan.vi which implement one and two throttle control, respectively, which implement throttle_rt_control.vi. The RT example VI shows how the interface takes place with the FPGA application. Most likely your application will require additional features and Drivven module implementations. At the FPGA level, you can include other module interface VIs within the same single cycle loop. Likewise, at the RT level you can include FPGA interface code for the additional modules within the same timed loop, or create multiple timed loops for different functions.

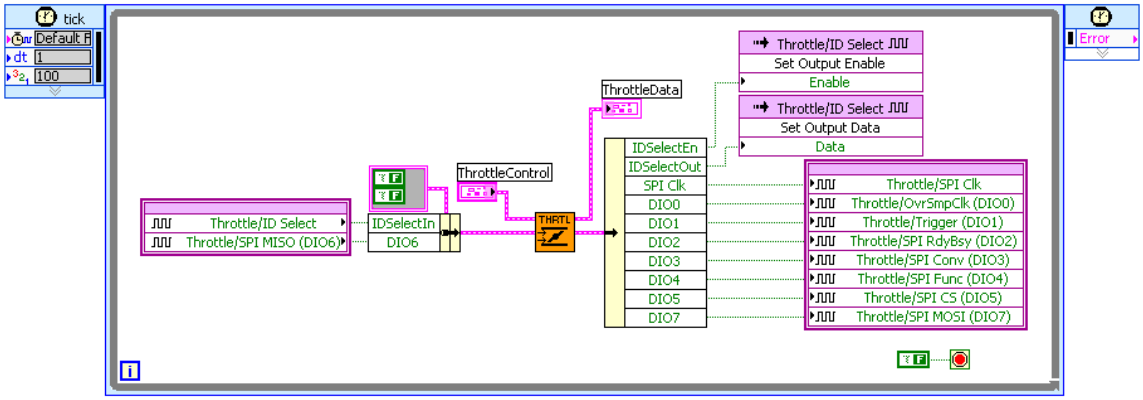


Figure 12. LabVIEW FPGA Block diagram example.

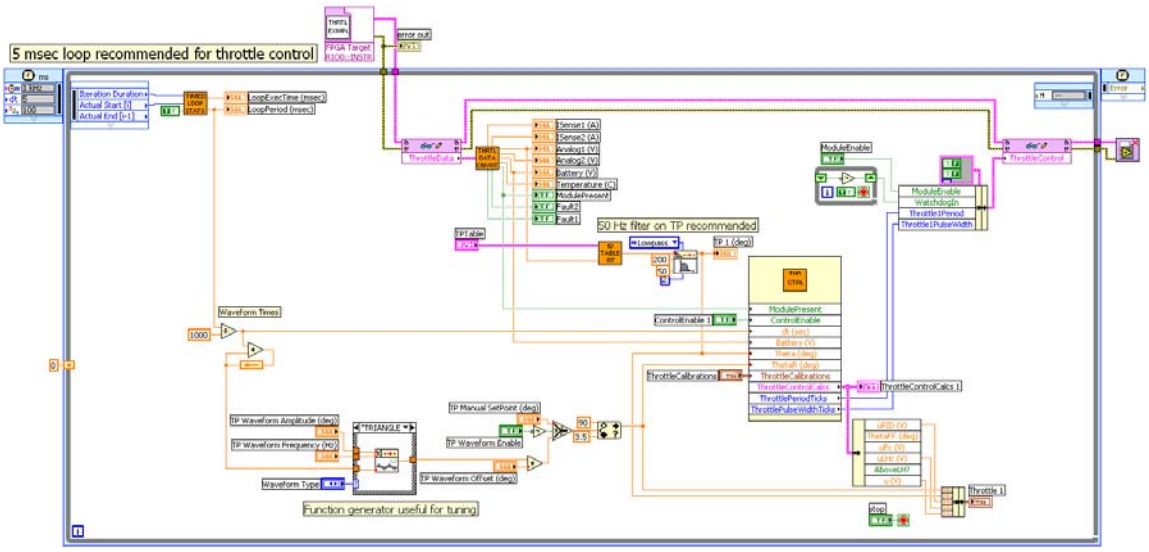


Figure 13. LabVIEW RT Block diagram example.