DRIVVEN

Engine Position Tracking VI User's Manual

D000031
D000033
D000035
Rev B

January 2, 2007

Drivven, Inc. • 14027 N. Hills Village • San Antonio, Texas 78249 • USA
Phone : 210.248.9308
Web : www.drivven.com , E-mail : info@drivven.com

Introduction

The Drivven family of Engine Position Tracking (EPT) VIs tracks the angular position of a wide variety of crankshaft trigger patterns within the LabVIEW FPGA environment. Each VI provides a standardized interface to higher level CPU algorithms and also with other fuel and spark LabVIEW FPGA VIs which it supervises. The EPT VIs are designed to track engine position and supervise fuel and spark output VIs so that the CPU is required to respond to little or no engine synchronous events. Flags are provided for errors. Once initialized, the EPT VIs, combined with any number of fuel and/or spark VIs, are capable of operating fully autonomously.

Features:

- Angular crankshaft position tracking to approximately 0.1 crank angle degree resolution
- Supports N-M, N+1 and encoder style patterns
- Supports 2-stroke and 4-stroke engine configurations
- Built in crankshaft and camshaft pattern simulation for bench testing applications without a running engine
- Supervises any number of fuel and spark control VIs from Drivven
- Standardized interfaces to reduce high level CPU code modifications when changing engine configurations
- Provides outputs for speed, position, tooth count, tooth edge, sync and errors

Software

The Drivven Engine Position Tracking (EPT) VIs track angular position over 2-stroke or 4-stroke engine cycles of a wide variety of crankshaft and camshaft pattern types. The EPT LabVIEW FPGA VIs are configured programmatically by the user for a specific stroke and tooth count. Currently, Drivven supports the pattern types shown in table 1. Each pattern type may have a range of acceptable tooth count configurations or a fixed tooth count due to the nature of the pattern. Pattern types are supported separately by EPT VIs so that each VI can be optimized for its particular pattern to consume as few FPGA resources as possible.

VERY IMPORTANT NOTES:

The FPGA VIs require:

- LabVIEW 8.2 or later
- LabVIEW RT Module 8.2. or later
- LabVIEW FPGA Module 8.2 or later
- NI-RIO 2.2 or later

The FPGA VIs 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 VIs 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.

This documentation covers the use of the EPT VIs. However, the following note is important for the use of accompanying CompactRIO modules from Drivven.
Drivven CompactRIO module FPGA VIs require 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.

Table 1. Supported EPT Patterns

EPT VI Directory Name Format	Pattern Name	Pattern Description
ept_nm_vte7_revb	N-M	N-M patterns have N evenly spaced teeth and M missing adjacent teeth (gap) on the crankshaft trigger wheel. M is limited to 1 or 2 missing teeth. N is stated in terms of evenly spaced teeth, as if there were no missing teeth. For example, a 60-2 pattern would consist of 58 physical teeth followed by two missing teeth. EPT N-M VIs can be configured to track 360 or 720 degrees depending upon an optional camshaft input. When tracking 360 degrees for 2-stroke engines, the camshaft input is ignored. When tracking 720 degrees for 4-stroke engines, the camshaft input signal must be TRUE at every other crank tooth gap.

Table 1. Supported EPT Patterns (continued)

ept_p1_vte9_revb	N+1	N+1 patterns have N evenly spaced teeth and a single additional tooth (Plus1), placed between two evenly spaced teeth, on the crankshaft trigger wheel. The Plus1 tooth can be located at one of two different positions between two evenly spaced teeth – advanced or retarded. N is stated in terms of evenly spaced teeth, as if there were no additional teeth. For example, a 6+1 pattern would consist of 6 physical evenly spaced teeth and an additional physical tooth placed between two evenly spaced teeth. EPT N+1 VIs can be configured to track 360 or 720 degrees depending upon an optional camshaft input. When tracking 360 degrees for 2-stroke engines, the camshaft input is ignored. When tracking 720 degrees for 4-stroke engines, the camshaft input signal must be TRUE at every other Plus1 tooth.
ept_enc_vte4_revb ept_enc_vte6_revb ept_enc_vte8_revb	Encoder	Encoder patterns have N evenly spaced teeth on the crankshaft with a single reference tooth on the camshaft. The rising or falling edge of the reference tooth must occur between the same two crankshaft teeth each cycle. The EPT ENC VI can be configured to track 360 or 720 degrees depending upon the location of the reference tooth. When tracking 360 degrees, the reference tooth is located on the crankshaft, yet is wired to the CamSig input of the EPT VI. When tracking 720 degrees, the reference tooth is located on the camshaft. Encoder style patterns are a very common form of engine position tracking in both production and research engine control systems.

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.

Figure 1 shows the icons which represent the current EPT VIs, having similar clustered terminals.

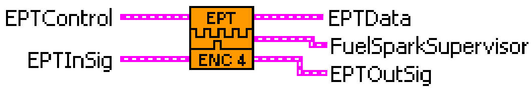


Figure 1a. Encoder EPT VI icon with leads (4-bit extrapolation).

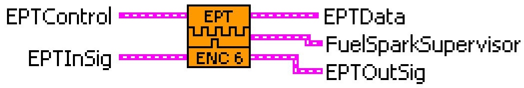


Figure 1b. Encoder EPT VI icon with leads (6-bit extrapolation).

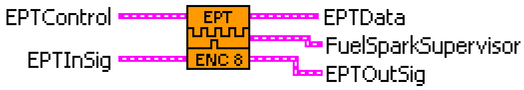


Figure 1c. Encoder EPT VI icon with leads (8-bit extrapolation).

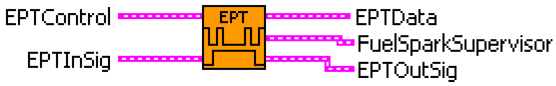


Figure 1d. N-M EPT VI icon with leads (7-bit extrapolation).

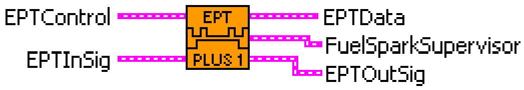


Figure 1e. Plus1 EPT VI icon with leads (9-bit extrapolation).

The EPT VIs are required for supervising all Drivven fuel and spark VIs, which are included with the fuel and spark cRIO module kits. The EPT VI provides the necessary FuelSparkSupervisor output cluster to be wired to the similarly named input clusters of the fuel and spark VIs.

If a pattern is encountered which is not supported, please contact Drivven with the desired pattern specifications. Drivven can quickly provide custom pattern position tracking.

EPT VI Commonality

All EPT VIs have a common internal core that tracks angular position. The supporting internal logic around the core interprets the incoming crank and cam position pulses according to an expected pattern. Each EPT VI has a crankshaft and camshaft input. These signals should be wired from the VR or Hall outputs from the ad_combo.vi or vr_hall.vi provided with Drivven's AD Combo or VR/Hall cRIO Module Kits. The pattern input signals are named according to the most common engine applications, such that there is a crankshaft trigger wheel and an optional 4-stroke camshaft trigger wheel. However, it is also possible for some VI pattern types to operate with a single trigger wheel pattern mounted to a camshaft. This would require routing the camshaft trigger wheel sensor signal into the crankshaft input of the EPT VI. The EPT VI however may need to be configured as a two-stroke application. Please contact Drivven for assistance with these configurations.

EPT Pattern Simulation

All EPT VIs contain a built in pattern simulation tool specific to the pattern for which the EPT VI was configured. Pattern simulation is turned on simply by setting SimEnable to TRUE and setting the SimPeriod to an appropriate value for the simulated speed. An RT VI is provided by Drivven to convert from engine speed in RPM to SimPeriod. These controls are found within the EPTControl cluster. The simulated crank and cam signals are output on the SimCrankSig and SimCamSig indicators within the EPTOutSig cluster. When simulation is turned on, the CrankSig and CamSig inputs to the EPT VI are internally disconnected and replaced by the simulated counterparts. The simulation feature allows the designer to bench test the engine control system functionality at any stage of the development without the need for external simulation hardware. The simulated crank and cam signals can be monitored on an oscilloscope while also monitoring the fuel and/or spark actuators to verify the control strategy.

Starting Position Tracking (Sync)

In order for position tracking, or sync, to start, a few conditions must be met. First, the CPU must enable sync by setting SyncEnable to TRUE. Then the crankshaft speed must surpass the stall speed. Technically, this means that the measured Period output must be less than the CrankStallPeriod input. The CrankStallPeriod may be configured to the EPT VI in terms of RPM by using the RT VI named speed2ticks.vi. When the stall speed (RPM) is exceeded, the Stalled output within the EPTData cluster will be set to FALSE and the EPT VI begins counting the number of crank teeth specified by the CrankCountStart control within the EPTControl cluster. After CrankCountStart is satisfied, the EPT VI begins looking for the first occurrence of a unique feature of the crankshaft and camshaft pattern which identifies an absolute position. For example, if an N-M pattern is being used, the VI searches for the occurrence of the M-tooth gap. This is the moment of achieving sync. When sync is achieved, the SyncStopped output within the EPTData cluster is set to FALSE. Each EPT VI will have different rules for achieving and maintaining sync. This information is contained in the documentation below, specific for each pattern type.

Period, PeriodAccum, CrankCount and CurrentPosition
During sync, the EPT provides the latest period between trigger teeth at the Period output within the EPTData cluster in terms of 40 MHz clock ticks. Period is reported as though there were no missing or extra teeth. Therefore the CPU may use this value for calculating the latest instantaneous engine speed. Another output value named PeriodAccum within the EPTData cluster contains an accumulation of Period values over the number of teeth specified by PeriodAccumCount. The PeriodAccum value is updated each time PeriodAccumCount teeth have passed. PeriodAccum may be used to easily calculate an average crankshaft speed over a complete rotation. Crankshaft speed may be calculated in RPM according to the following:

EngineSpeedInstant(RPM) = (60 * 4E7) / (Period * NumberOfCrankTeeth)

EngineSpeedAve(RPM) = (60 * 4E7) / (PeriodAccum * NumberOfCrankTeeth / PeriodAccumCount)

Drivven provides utility VIs which can be implemented at the LabVIEW RT level for performing these calculations. The VI named ticks2speed.vi can be used to convert Period to RPM. For simulation purposes, the VI named speed2ticks.vi can be used to convert RPM set point to SimPeriod. The icons for these VIs are shown in Figure 4.

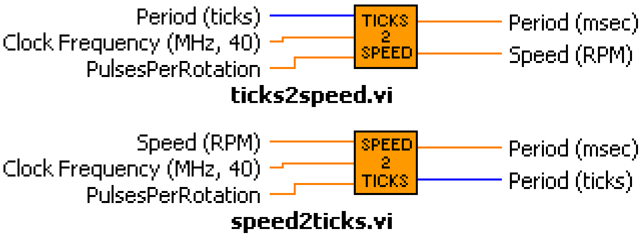


Figure 4. Period / Speed conversion VIs.

During sync, the EPT provides the latest crank tooth count at the CrankCount output within the EPTData cluster, referenced to tooth 0. Tooth 0 is determined according to the pattern that is being tracked and is discussed below within the pattern specific documentation. CrankCount is provided in terms of a complete engine cycle and is reported as if there were no missing or extra teeth in the pattern. For example, if PATTERN = N-M, NumberOfCrankTeeth = 60 and Stroke = TRUE, then CrankCount will range from 0 to 119.

During sync, the EPT provides the latest crankshaft position at the CurrentPosition output within the EPTData cluster in terms of absolute Crank Angle Ticks (CAT) over a complete engine cycle, referenced to position 0. Position 0 is determined according to the pattern that is being tracked and is discussed below within the pattern specific documentation. Drivven provides each EPT pattern type with one or more optimized VIs for a particular range of number of crank teeth. For example, the encoder pattern EPT is provide with three VIs, each optimized for low, medium and high crank tooth counts. The suggested tooth count ranges are shown below in Table 2 for each pattern type. The extrapolation value will affect the CAD resolution of CurrentPeriod. Drivven provides EPT VIs configured with extrapolation values so that 0.1 CAD (or better) resolution is achieved for most crank tooth count values. The resolution grows up to 0.35 CAD for very low tooth counts.

For example, if using a 4-stroke 36-1 pattern, then the suggested EPT VI is the ept_nm_vte7_revb.vi along with ept_nm_vte7_rev.ngc. The “e7” component of the VI name refers to the amount of binary extrapolation of crank angle between each crank tooth. In this case, CurrentPosition would increment by 2^7, or 128 CAT between each crank tooth. The range of CurrentPeriod would be from 0 to MAX_CAT = 2*36*128 = 9216. This is the total number of angular positional ticks that are tracked by the EPT over a complete 4-stroke engine cycle. Therefore CurrentPosition will track from 0 to 9216 and roll over to 0 at the occurrence of tooth 0.

This provides an angular resolution of 0.078 CAD/CAT. The angular position in CAD can be calculated according to the following:

CurrentPosition(CAD) = [CurrentPosition(CAT) * Stroke * 360] / MAX_CAT

Where Stroke = 2 if 4-stroke and Stroke = 1 if 2-stroke.

Drivven provides a utility VI which can be implemented at the LabVIEW RT level for performing this calculation. The VI named CAT2CAD.vi can be used to convert CAT to CAD. This VI icon is shown in Figure 5.

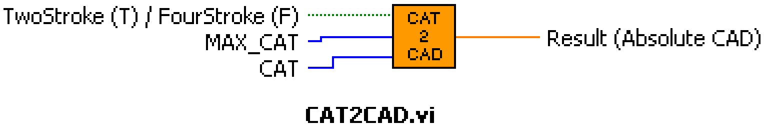


Figure 5. CAT / CAD conversion VI.

Table 2. EPT VI Extrapolation Values and Suggestion Tooth Count Ranges

Pattern Type	NumberOfCrankTeeth	Stroke	Extrapolation	MAX_CAT	CAD / CAT
Encoder (max)	1440	4	4	46080	0.02
Encoder (min)	120	4	4	3840	0.19
Encoder (max)	120	4	6	15360	0.05
Encoder (min)	60	4	6	7680	0.09
Encoder (max)	60	4	8	30720	0.02
Encoder (min)	4	4	8	2048	0.35
N-M (max)	120	4	7	30720	0.02
N-M (min)	12	4	7	3072	0.23
N+1 (max)	24	4	9	24576	0.03
N+1 (min)	4	4	9	4096	0.18

Loss of Sync

The EPT VI will lose sync when missing or extra crank/cam pulses are encountered where unexpected. A loss of sync will be signaled by the appropriate pattern-specific error flag within the EPTData cluster. An obvious possible reason for such errors would be that the crank or cam trigger wheel pattern does not match the configuration of the EPT VI. If this is the case, the EPT will either not sync at all or will lose sync almost immediately after initial sync. Assuming that the trigger pattern is correct, there are a few other common sources of losing sync. The two most common sources are electrical noise and mechanical trigger wheel imperfections.

Electrical noise can be picked up and manifested in a wide variety of ways. The Drivven AD_Combo and VR/Hall Modules' adaptive VR sensor inputs and hall-effect sensor inputs perform very well at rejecting electrical noise. The documentation for the AD_Combo Module Kit should be consulted for more information about VR and hall-effect sensor input channel configuration. However, there are a few guidelines that should be followed for ensuring appropriate immunization to electrical noise. It is unlikely that noise will be introduced on the crank or cam signals between the cRIO module and the main RIO FPGA. It is most likely that electrical noise will be picked up between the sensor and the external connection to the module. If possible, the wiring for VR sensors should be a twisted shielded pair and as short as possible. Hall-effect sensor wiring should also preferably be shielded and as short as possible. There should be few (or no) electrical connections between the sensor and the module. The signals should be routed away, if possible, from ignition primary wires or ignition coils. Efforts should focus on external wiring. However, if noise spikes are still causing false triggering of the VR or hall-effect sensor circuit, digital filtering may be implemented at the FPGA level using Drivven's integrating digital filter, offered with the ADCombo and VR/Hall Module Kits. The filter VI is called filt_int_reva.vi. The filter allows glitches up to a specified width to be rejected. The output of Drivven's hall-effect sensor circuit follows the input with a small analog filter delay. It also inverts the signal. The output of Drivven's VR sensor circuit has a rising edge at the center of the sensed tooth or slot.

Mechanical trigger wheel imperfections, if sensed by a VR sensor, will usually manifest themselves as significant VR pulses at higher engine speeds. Trigger wheels are machined in two varieties: positive teeth or negative teeth (slots). Sometimes positive teeth will have distinct edges left behind by the machining process at the rising or falling edge of the mechanical tooth. At low speeds it may be difficult to see a resulting VR pulses on an oscilloscope, and the EPT will sync just fine. However, at higher speeds the generated pulse from the imperfection may become significant enough to be accepted by the VR input circuit. Drivven's VR circuits are adaptive and will reject a pulse that is approximately 50% amplitude (or lower) of the pulse before it, if the imperfection is immediately following a real tooth. As the imperfection becomes farther away from a real tooth, the triggering threshold becomes lower. Negative trigger teeth, or slotted wheels can also have imperfections along the flat surfaces where no slots are present. Since the sensor is mounted very close to these flat areas, they will be sensitive to imperfections in the surface. Therefore it may be required to increase the sensor gap or remove the offending imperfections if they cause false outputs to the RIO FPGA.

Achieving reliable, clean crank and cam signals can sometimes be a trial and error process. It is Drivven's experience that this setup process usually does not work perfectly with the first attempt at running the engine over the entire speed and load range. Even though sync may occur reliable at low engine speeds does not mean that there will not be signal issues at high speeds or loads due to trigger wheel imperfections, ignition system noise, or camshaft position lash with respect to the crankshaft. The system designer should take care to check that external signal shielding and routing are optimal and that there is plenty of margin for cam belt or chain stretch. It is always useful to have access to a good triggering multi-channel mixed signal oscilloscope. Drivven strives to make the engine control prototyping process as rapid as possible with modular products that are quickly configured. However, patience and perseverance are still required as always.

With that said, there are some tricks the designer can employ with the use of an oscilloscope to locate the source of sync problems. After all, finding the source of a problem is most of the battle. If the EPT error flags are routed out to external digital outputs via unused cRIO slot pins, they can be used as triggers to capture crank and cam signal problems. This is where a multi-channel mixed signal oscilloscope is so useful. For example, the designer could route the CrankSig, CamSig and ErrorFlag signals out to unused cRIO slot pins. These signals could be monitored on digital or analog scope channels. The external VR and/or hall-effect signals could also be monitored on the scope's analog channels. By triggering on the rising edge of one of the error flags, it would be possible to see the occurrence of signal imperfections on the external signals. It would also be possible to see the correlation between the external signals and the signals presented to the CrankSig and CamSig inputs. This information can be used to determine the proper problem-solving action to take.

The EPT will also lose sync if the crankshaft speed falls below the stall speed, as specified with the CrankStallPeriod input to the EPT VI. When sync is lost, CrankCount and CurrentPosition will be set to 0 and SyncStopped will be set to TRUE. When crank speed falls below the stall speed, Period and PeriodAccum will be set to 0 and Stalled will be set to TRUE. When calculating engine speed from Period, be sure to check for this zero condition, otherwise a divide-by-zero condition will be encountered. The ticks2speed.vi accounts for this.

Position Tracking

Between each crank tooth, CurrentPosition is incremented smoothly at a rate proportional to the previous crank tooth period. In the presence of crankshaft acceleration or deceleration, sync is still properly maintained. If a new crank tooth is received before CurrentPosition has incremented to the correct position, then CurrentPosition is incremented at a faster rate until proper synchronization is achieved. CurrentPosition is then further incremented at a rate proportional to the previous crank tooth period. If CurrentPosition reaches the correct position expected for the next tooth before the tooth is received, then CurrentPosition holds its value until the next crank tooth. This mechanism, along with the aid of other unique pattern specific features, make the EPT VIs a very robust position tracking algorithm.

Block Diagram Design

The EPT VIs 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.

Each EPT 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 VI.

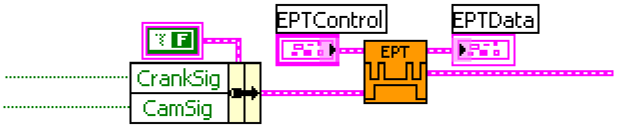


Figure 6. Example block diagram implementation of EPT VIs.

EPTInSig (Cluster)

CrankSig (boolean): The digital input signal representing the crankshaft trigger wheel pulse train. This input should be wired directly from a VR or Hall output of the AD Combo or VR/Hall Module Kit VI. The EPT VI responds to the rising edge of CrankSig.

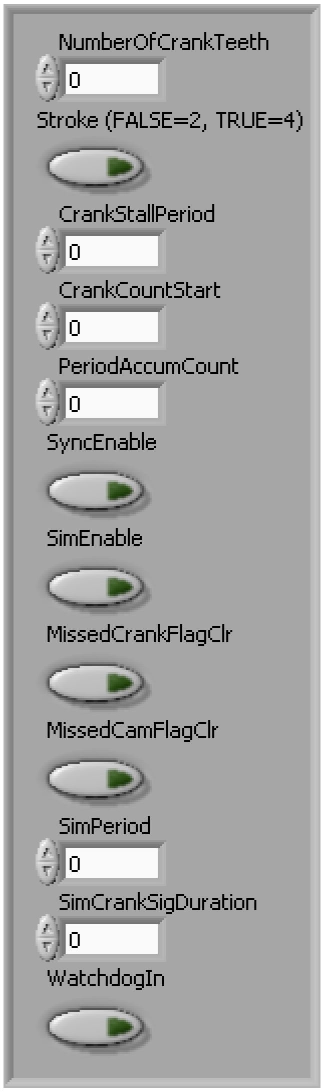
CamSig (boolean): The digital input signal representing the camshaft trigger wheel pulse train. This input should be wired directly from a VR or Hall output of the AD Combo or VR/Hall Module Kit VI. For some pattern types, the EPT VI responds to the rising edge of CamSig. For other pattern types, the EPT VI tests the level of CamSig.

EPTOutSig (Cluster)

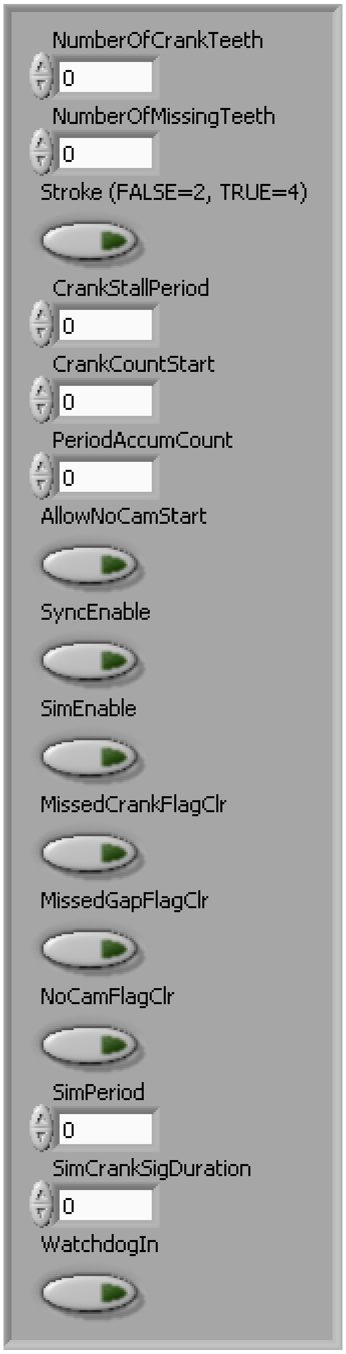
SimCrankSig (boolean): Digital output signal generated by the built-in pattern simulation tool to simulate the crankshaft pattern for which the EPT VI is configured.

SimCamSig (boolean): Digital output signal generated by the built-in pattern simulation tool to simulate the camshaft pattern for which the EPT VI is configured.

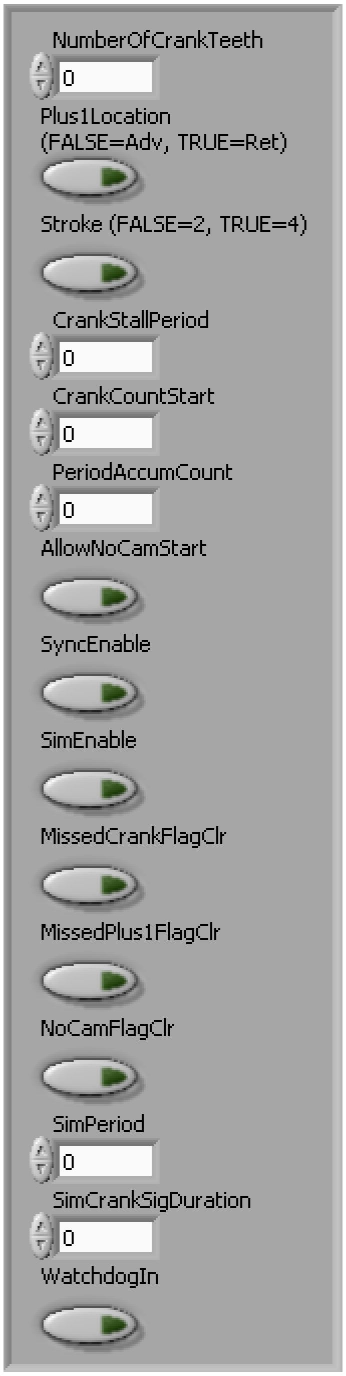
EPTControl (Cluster)


NumberOfCrankTeeth
0
Stroke (FALSE=2, TRUE=4)
CrankStallPeriod
0
CrankCountStart
0
PeriodAccumCount
0
SyncEnable
SimEnable
MissedCrankFlagClr
MissedCamFlagClr
SimPeriod
0
SimCrankSigDuration
0
WatchdogIn

EPT Encoder


NumberOfCrankTeeth
0
NumberOfMissingTeeth
0
Stroke (FALSE=2, TRUE=4)
CrankStallPeriod
0
CrankCountStart
0
PeriodAccumCount
0
AllowNoCamStart
SyncEnable
SimEnable
MissedCrankFlagClr
MissedGapFlagClr
NoCamFlagClr
SimPeriod
0
SimCrankSigDuration
0
WatchdogIn

EPT N-M


NumberOfCrankTeeth
0
Plus1Location
(FALSE=Adv, TRUE=Ret)
Stroke (FALSE=2, TRUE=4)
CrankStallPeriod
0
CrankCountStart
0
PeriodAccumCount
0
AllowNoCamStart
SyncEnable
SimEnable
MissedCrankFlagClr
MissedPlus1FlagClr
NoCamFlagClr
SimPeriod
0
SimCrankSigDuration
0
WatchdogIn

EPT N+1

NumberOfCrankTeeth (uint16): Specifies the number of teeth on the crankshaft trigger wheel as if there were no missing or extra crank teeth. For example, a 60-2 crankshaft pattern would have NumberOfCrankTeeth = 60.

NumberOfMissingTeeth (uint16): Specifies the number of missing adjacent teeth on the crankshaft trigger wheel for an N-M pattern type. For example, a 60-2 crankshaft pattern would have NumberOfMissingTeeth = 2. This parameter is only available for N-M pattern types.

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

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

CrankStallPeriod (uint32): The maximum crank tooth period between evenly spaced crank teeth in terms of 40 MHz clock ticks. A measured crank tooth period greater than CrankStallPeriod causes the Period output of the EPT VI to report as zero and the CrankStalled output to report as TRUE. Engine speed in RPM can be converted to CrankStallPeriod in 40 MHz clock ticks according to the following equation:

CrankStallPeriod(clock ticks) = (60 * 4E7) / (EngineSpeed(RPM) * NumberOfCrankTeeth)

CrankCountStart (uint16): Specifies the number of teeth on the crankshaft trigger wheel to count before looking for the pattern-specific unique feature indicating tooth 0. This can help with waiting for a cam shaft VR signal to gain recognition, or it can help reject a number of false triggers during the initial engagement of a starter motor. The value must be set with a minimum of 2.

PeriodAccumCount (uint16): Specifies the number of crank teeth to accumulate Period in the PeriodAccum output of the EPT VI. PeriodAccum can be used to measure an average crankshaft speed over multiple crank teeth.

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

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

SimEnable (boolean): When TRUE, CrankSig and CamSig inputs within the EPTInSig cluster are internally disconnected from the position tracking core and simulated signals are connected in their place.

ErrorFlagClr (boolean): The names and number of error flags will vary depending on EPT pattern type. When TRUE, the corresponding error flag output within the EPTData cluster will be cleared.

SimPeriod (uint32): The simulated period between evenly spaced crank teeth in terms of 40 MHz clock ticks. SimPeriod is only effective when it is set to a non-zero value while SimEnable is TRUE. The EPT will not sync to the simulator unless the effective engine speed is greater than the stall speed. Engine speed in RPM can be converted to SimPeriod in 40 MHz clock ticks according to the following equation:

SimPeriod(clock ticks) = (60 * 4E7) / (EngineSpeed(RPM) * NumberOfCrankTeeth)

SimCrankSigDuration (uint32): The pulse width of simulated crank teeth in terms of 40 MHz clock ticks. SimCrankSigDuration in milliseconds can be converted to SimCrankSigDuration in 40 MHz clock ticks according to the following equation:

SimCrankSigDuration(clock ticks) = 4E4 * SimCrankSigDuration(msec)

WatchdogIn (boolean): The WatchdogIn boolean signal must be toggled at a rate greater than 4 Hz. The purpose of the watchdog is to shutdown engine position tracking (as well as any engine synchronous outputs) in the case of a loss of LabVIEW RT communication with the FPGA. For example, the LabVIEW RT application may fail due to coding errors without properly closing the FPGA application. It would not be desirable for the EPT to continue tracking engine position and supervising engine synchronous outputs according to the last command received from the LabVIEW RT application. Figure 7 shows an example of toggling the WatchdogIn signal with a simple while loop structure within LabVIEW RT. The toggled signal is wired to the WatchdogIn element of a Cluster Bundle By Name function.

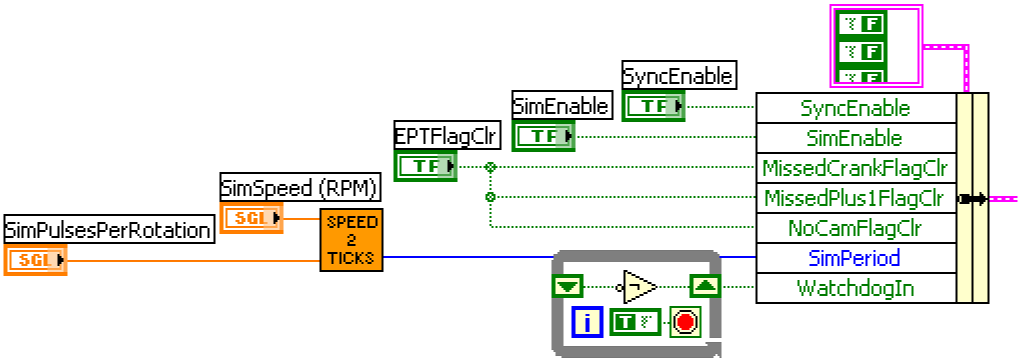
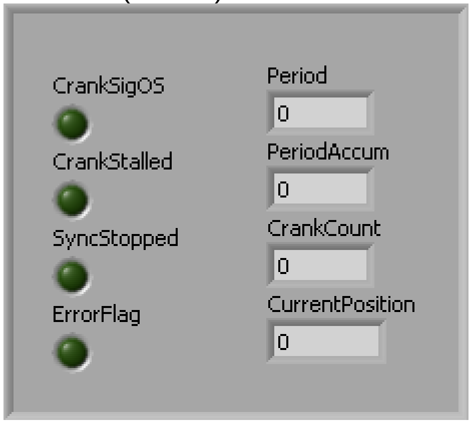


Figure 7. Example of toggling the WatchdogIn.

EPTData (Cluster)



CrankSigOS (boolean): A 40 MHz one-clock one-shot generated at the rising edge of each received or simulated CrankSig input. This signal can be used in conjunction with CrankCount for any purpose within LabVIEW FPGA logic to perform engine synchronous tasks.

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

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

ErrorFlag (boolean): The names and number of error flags will vary depending on EPT pattern type. When TRUE, the EPT VI has detected an error while tracking crankshaft position. The resulting actions are pattern-specific. Error flags are cleared by setting the corresponding flag within the EPTControl cluster to TRUE.

Period (uint32): The period between the last two crank trigger teeth, as if there were no missing or extra teeth, in terms of 40 MHz clock ticks. Period will be set to 0 while Stalled is TRUE. Period in clock ticks can be converted to instantaneous engine speed in RPM according to the following equation:

EngineSpeedInstant(RPM) = (60 * 4E7) / (Period * NumberOfCrankTeeth)

PeriodAccum (uint32): The accumulated period over PeriodAccumCount crank teeth in terms of 40 MHz clock ticks. PeriodAccum will be set to 0 while Stalled is TRUE. PeriodAccum in clock ticks can be converted to average engine speed in RPM according to the following equation:

EngineSpeedAve(RPM) = 60 * 4E7) / PeriodAccum

CrankCount (uint16): The latest tooth count referenced to crank tooth 0. Tooth 0 is determined according to the pattern that is being tracked and is discussed within the pattern specific documentation. CrankCount is provided in terms of a complete engine cycle and is reported as if there were no missing or extra teeth in the pattern. For example, if PATTERN = N-M, NumberOfCrankTeeth = 60 and Stroke = TRUE, then CrankCount will range from 0 to 119.

CurrentPosition (uint16): The current angular position of the crankshaft in terms of absolute Crank Angle Ticks (CAT) over a complete engine cycle, referenced to position 0. Position 0 is determined according to the pattern that is being tracked and is discussed within the pattern specific documentation.

Drivven provides each EPT pattern type with one or more optimized VIs for a particular range of number of crank teeth.

For example, the encoder pattern EPT is provide with three VIs, each optimized for low, medium and high crank tooth counts. The suggested tooth count ranges are shown in Table 2 for each pattern type. The extrapolation value will affect the CAD resolution of CurrentPeriod. Drivven provides EPT VIs configured with extrapolation values so that 0.1 CAD (or better) resolution is achieved for most crank tooth count values. The resolution grows up to 0.35 CAD for very low tooth counts.

For example, if using a 4-stroke 36-1 pattern, then the suggested EPT VI is the ept_nm_vte7_revb.vi along with ept_nm_vte7_rev.ngc. The “e7” component of the VI name refers to the amount of binary extrapolation of crank angle between each crank tooth. In this case, CurrentPosition would increment by 2^7, or 128 CAT between each crank tooth. The range of CurrentPeriod would be from 0 to MAX_CAT = 2*36*128 = 9216. This is the total number of angular positional ticks that are tracked by the EPT over a complete 4-stroke engine cycle. Therefore CurrentPosition will track from 0 to 9216 and roll over to 0 at the occurrence of tooth 0. This provides an angular resolution of 0.078 CAD/CAT. The angular position in CAD can be calculated according to the following:

CurrentPosition(CAD) = [CurrentPosition(CAT) * Stroke * 360] / MAX_CAT

Where Stroke = 2 if 4-stroke and Stroke = 1 if 2-stroke.

FuelSparkSupervisor (Cluster)
This cluster output must be wired directly to the FuelSparkSupervisor cluster inputs of Drivven fuel and/or spark VIs.

Pattern Specific Information

N-M Patterns

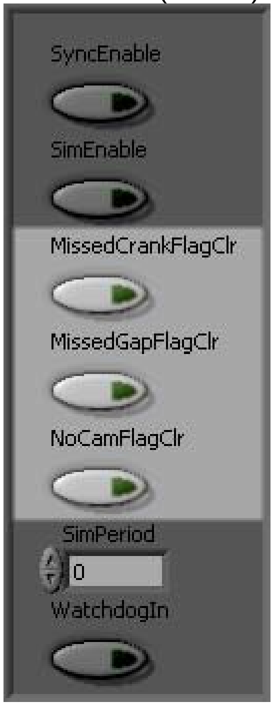
Pattern Description:
N-M patterns have N evenly spaced teeth and M missing adjacent teeth (gap) on the crankshaft trigger wheel. M is limited to 1 or 2 missing teeth. N is stated in terms of evenly spaced teeth, as if there were no missing teeth. For example, a 60-2 pattern would consist of 58 physical teeth followed by two missing teeth. EPT N-M VIs can be configured to track 360 or 720 degrees depending upon an optional camshaft input. When tracking 360 degrees for 2-stroke engines, the camshaft input is ignored. This is designated by configuring the VI with Stroke = FALSE. When tracking 720 degrees for 4-stroke engines, the camshaft input signal must be TRUE at every other crank tooth gap. This is designated by configuring the VI with Stroke = TRUE.

2-Stroke Applications (Stroke = FALSE):
Position 0 and tooth 0 coincide with the first tooth following each crank tooth gap. The CamSig input is ignored. CrankCount will increment from 0 to NumberOfCrankTeeth - 1. During crank tooth gaps, CrankCount will continue to increment at the location of each missing tooth. For example, if Stroke = FALSE, NumberOfCrankTeeth (N) = 60 and NumberOfMissingTeeth (M) = 2, then CrankCount would increment from 0 to 59, such that teeth 58 and 59 would correspond to the two missing teeth and tooth 0 would correspond to the tooth following missing tooth 59.

4-Stroke Applications (Stroke = TRUE):
Position 0 and tooth 0 coincide with the first tooth following the crank tooth gap while the CamSig input is TRUE. The CamSig is actually only used for achieving initial sync. Once sync is achieved, it is maintained according to the location of each gap. However, CamSig is still monitored to make sure that its value is as expected during each gap. If CamSig is not at the expected level during each gap, then the NoCamFlag will be set to TRUE. The CPU is then able to make an appropriate decision. CrankCount will increment from 0 to 2*NumberOfCrankTeeth - 1. During crank tooth gaps, CrankCount will continue to increment at the location of each missing tooth. For example, if Stroke = 4, NumberOfCrankTeeth (N) = 60 and NumberOfMissingTeeth (M) = 2, then CrankCount would increment from 0 to 119, such that teeth 58 and 59 would correspond to the two missing teeth while CamSig is FALSE. Teeth 118 and 119 would correspond to the two missing teeth while CamSig is TRUE. Tooth 0 would correspond to the tooth following missing tooth 119.

If AllowNoCamStart is TRUE, the EPT N-M VI can achieve initial sync upon the first detected gap, whether CamSig is TRUE or FALSE. If CamSig is TRUE, sync will begin at position 0. If CamSig is FALSE, sync will begin at the mid-cycle position of CurrentPosition = (MAX_CAT / 2) or CrankCount = NumberOfCrankTeeth.

EPTControl (Cluster)

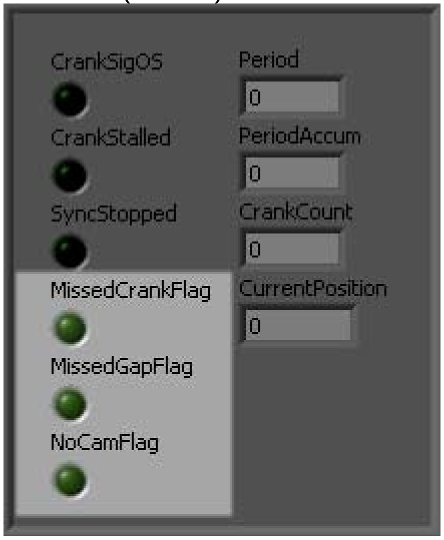


MissedCrankFlagClr (boolean): When TRUE, the MissedCrankFlag within the EPTData cluster will be cleared.

MissedGapFlagClr (boolean): When TRUE, the MissedGapFlag within the EPTData cluster will be cleared.

NoCamFlagClr (boolean): When TRUE, the NoCamFlag within the EPTData cluster will be cleared.

EPTData (Cluster)



Period
0
PeriodAccum
0
CrankCount
0
CurrentPosition
0

MissedCrankFlag (boolean): Set to TRUE upon the detection of a crank tooth gap while the CrankCount value has not yet reached the expected value. In other words, a gap has been detected sooner than expected. This condition causes a loss of sync. Re-sync is not allowed until the flag is cleared by setting MissedCrankFlagClr to TRUE.

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

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

N+1 Patterns

Pattern Description:
N+1 patterns have N evenly spaced teeth and a single additional tooth (Plus1), placed between two evenly spaced teeth, on the crankshaft trigger wheel. There are two different sub-configurations within this pattern type. An advanced Plus1 tooth must be positioned such that it is advanced from the midpoint between two evenly spaced teeth. The amount of advance must be at least ¼ of the normal tooth spacing. A retarded Plus1 tooth must be positioned such that it is retarded from the midpoint between two evenly spaced teeth. The amount of retard must be at least ¼ of the normal tooth spacing. N is stated in terms of evenly spaced teeth, as if there were no additional teeth. For example, a 6+1 pattern would consist of 6 physical evenly spaced teeth and an additional physical tooth placed between two evenly spaced teeth. EPT N+1 VIs can be configured to track 360 or 720 degrees depending upon an optional camshaft input. When tracking 360 degrees for 2-stroke engines, the camshaft input is ignored. This is designated by configuring the VI with Stroke = FALSE. When tracking 720 degrees for 4-stroke engines, the camshaft input signal must be TRUE at every other Plus1 tooth. This is designated by configuring the VI with Stroke = TRUE.

2-Stroke Applications (Stroke = FALSE):
For advanced Plus1 tooth configurations, position 0 and tooth 0 coincide with the first tooth following the Plus1 tooth. For retarded Plus1 tooth configurations, position 0 and tooth 0 coincide with the second tooth following the Plus1 tooth. The CamSig input is ignored. CrankCount will increment from 0 to NumberOfCrankTeeth - 1. CrankCount will not be incremented at the location of the Plus1 tooth. For example, if the Plus1 tooth is advanced, Stroke = FALSE and NumberOfCrankTeeth = 6, then CrankCount would increment from 0 to 5, such that tooth 5 would correspond to the tooth immediately before the Plus1 tooth and tooth 0 would correspond to the tooth immediately following the Plus1 tooth.

4-Stroke Applications (Stroke = TRUE):
For advanced Plus1 tooth configurations, position 0 and tooth 0 coincides with the first tooth following the Plus1 tooth while the CamSig input is TRUE. For retarded Plus1 tooth configurations, position 0 and tooth 0 coincide with the second tooth following the Plus1 tooth while the CamSig input is TRUE. The CamSig is actually only used for achieving initial sync. Once sync is achieved, it is maintained according to the location of each Plus1 tooth. However, CamSig is still monitored to make sure that its value is as expected during each Plus1 tooth. If CamSig is not at the expected level during each Plus1 tooth, then the NoCamFlag will be set to TRUE. The CPU is then able to make an appropriate decision. CrankCount will increment from 0 to 2*NumberOfCrankTeeth - 1. CrankCount will not increment at the Plus1 tooth. For example, if the Plus1 tooth is advanced, Stroke = TRUE and NumberOfCrankTeeth = 6, then CrankCount would increment from 0 to 11, such that tooth 5 and tooth 11 would correspond to the tooth immediately before the Plus1 tooth while CamSig is FALSE and CamSig is TRUE, respectively. Tooth 6 and tooth 0 would correspond to the tooth immediately following the Plus1 tooth while CamSig is FALSE and CamSig is TRUE, respectively.

Tooth numbering for the two different Plus1 tooth location configurations is shown in figures 8 and 9 below.

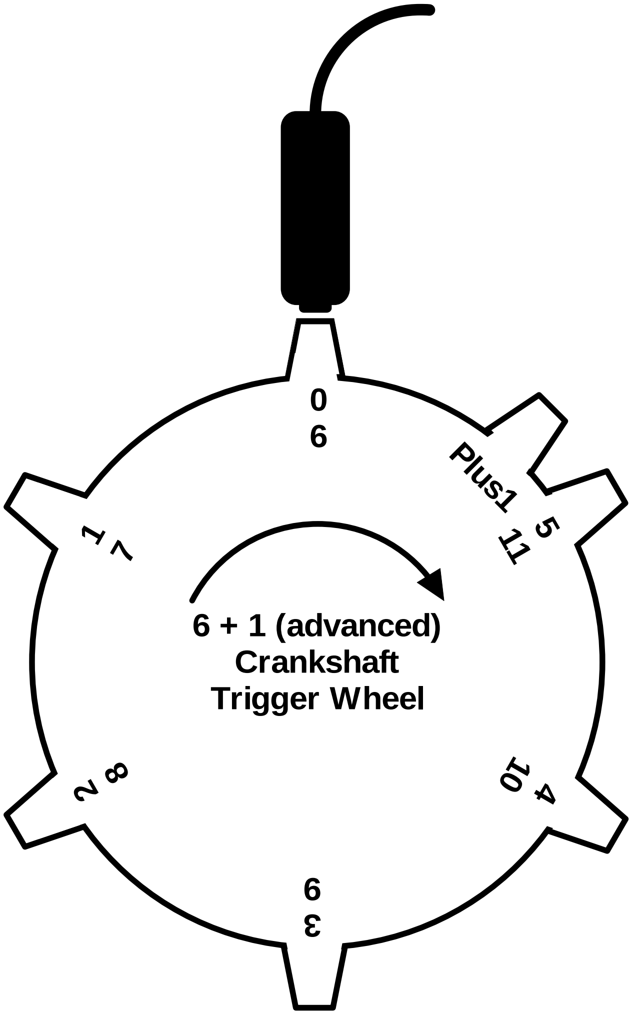


Figure 8. 6+1 (advanced) crankshaft trigger wheel tooth numbering

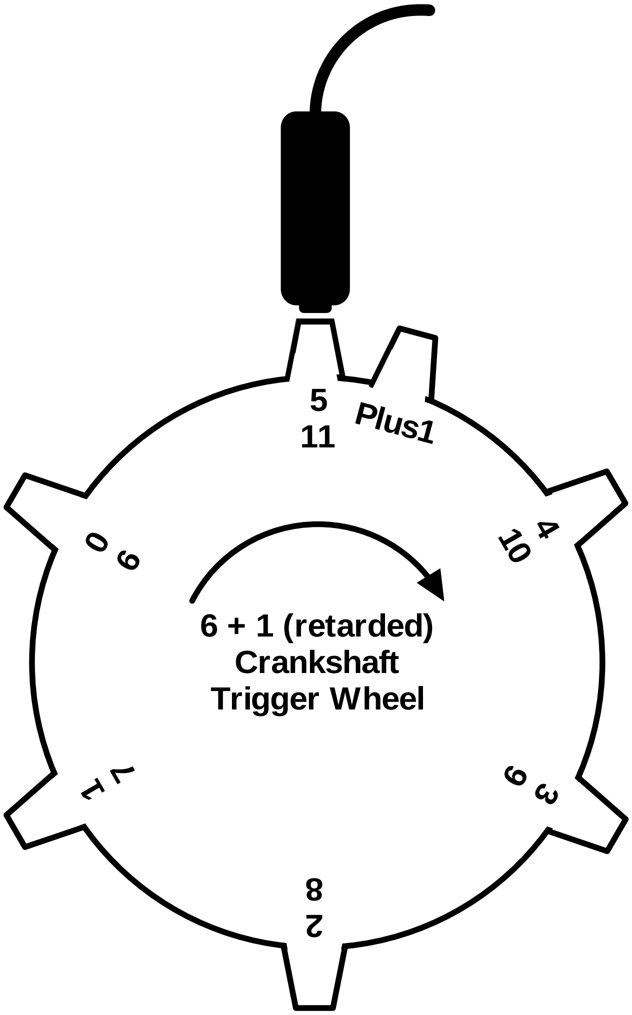
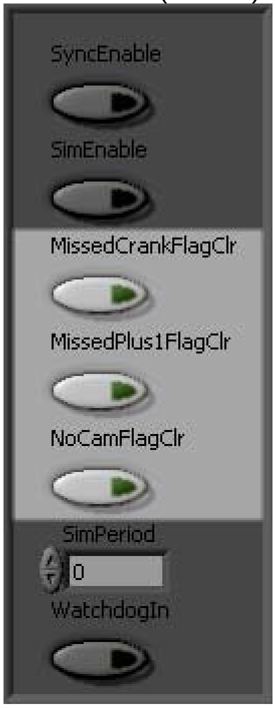


Figure 9. 6+1 (retarded) crankshaft trigger wheel tooth numbering

If AllowNoCamStart is TRUE, the EPT N+1 VI can achieve initial sync upon the first detected Plus1 tooth, whether CamSig is TRUE or FALSE. If CamSig is TRUE, sync will begin at position 0. If CamSig is FALSE, sync will begin at the mid-cycle position of CurrentPosition = (MAX_CAT / 2) or CrankCount = NumberOfCrankTeeth.

EPTControl (Cluster)

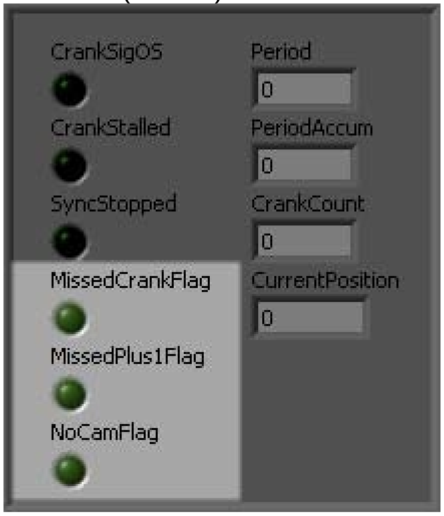


MissedCrankFlagClr (boolean): When TRUE, the MissedCrankFlag within the EPTData cluster will be cleared.

MissedPlus1FlagClr (boolean): When TRUE, the MissedPlus1Flag within the EPTData cluster will be cleared.

NoCamFlagClr (boolean): When TRUE, the NoCamFlag within the EPTData cluster will be cleared.

EPTData (Cluster)



Period
0
PeriodAccum
0
CrankCount
0
CurrentPosition
0

MissedCrankFlag (boolean): Set to TRUE upon the detection of a Plus1 tooth while the CrankCount value has not yet reached the expected value. In other words, a Plus1 tooth has been detected sooner than expected. This condition causes a loss of sync. Re-sync is not allowed until the flag is cleared by setting MissedCrankFlagClr to TRUE.

MissedPlus1Flag (boolean): Set to TRUE upon CrankCount being incremented past the expected value for the location of a Plus1 tooth before the Plus1 tooth is detected. In other words, evenly spaced teeth are received at the location where a Plus1 tooth is expected. This condition causes a loss of sync. Re-sync is not allowed until the flag is cleared by setting MissedPlus1FlagClr to TRUE.

NoCamFlag (boolean): Set to TRUE upon detecting 2 consecutive Plus1 teeth without detecting a TRUE CamSig (Stroke = TRUE). CamSig is only sampled at Plus1 tooth detection. This condition does not cause a loss of sync. NoCamFlag is cleared by setting NoCamFlagClr to TRUE.

Encoder Patterns

Pattern Description:
Encoder patterns have N evenly spaced teeth on the crankshaft with a single reference tooth on the camshaft. The rising or falling edge of the reference tooth must occur between the same two crankshaft teeth each cycle. The EPT ENC VI can be configured to track 360 or 720 degrees depending upon the location of the reference tooth. When tracking 360 degrees, the reference tooth is located on the crankshaft, yet is wired to the CamSig input of the EPT VI. When tracking 720 degrees, the reference tooth is located on the camshaft. Encoder style patterns are a very common form of engine position tracking in both production and research engine control systems.

In production systems, an encoder pattern will typically consist of a VR sensor looking at a relatively low resolution crankshaft trigger wheel of 4 to 16 teeth. The camshaft sensor may be a VR sensor or hall-effect sensor looking at a single tooth on the cam. For this setup, the rising or falling edge of the cam signal should occur as close as possible to the midway point between two crank signals. This ensures that the cam signal edge will repeatedly occur between the same two crank signals in the presence of crankshaft acceleration. The system designer should also ensure that the camshaft lash due to stretching of the camshaft belt or chain during acceleration or deceleration will not cause the cam signal to move with respect to the crank signals by more than half of the crank tooth spacing. For this reason, production systems will use a low resolution crank trigger wheel. The rising or falling edge of the cam sensor signal may be used as long as the signal is inverted properly within the LabVIEW FPGA block diagram to present a rising edge to the EPT VI CamSig input.

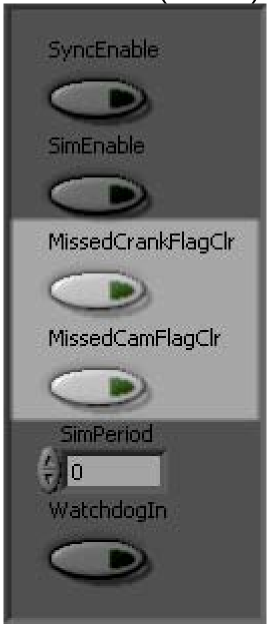
Optical encoders are a common source of signals for this pattern type in research applications. A single optical encoder will have at least three different outputs named A, B and Z. Signals A and B are quadrature encoded 90 degrees out of phase, while Z is a single reference pulse per 360 degrees rotation of the encoder shaft. Signal A or B may be wired to the CrankSig input of the EPT VI while signal Z must be wired to the CamSig input. The designer must remember that signals A and B are equally phased. One or the other must be selected to align with a known location of the crankshaft or camshaft. Encoders having pulse-counts of 360 or 720 pulses per rotation are common for research engine control applications. Although any pulse count may be used. The EPT ENC VI will then extrapolate this to 0.1 CAD or better.

If the optical encoder is mounted to the crankshaft, then NumberOfCrankTeeth should equal the encoder pulse-count and Stroke should be set to FALSE (2-stroke). This setup will track 360 degrees. Four-stroke engines may still be controlled under this setup; however, wasted spark ignition must be implemented as well as batch fueling. The highest possible crankshaft position accuracy will be achieved, since the encoder is attached directly to the crankshaft.

If the optical encoder is mounted to the camshaft, then NumberOfCrankTeeth should equal the encoder pulse-count divided by two, and Stroke should set to TRUE (4-stroke). This setup will track 720 degrees. With this setup, crankshaft positional accuracy may suffer under engine acceleration and deceleration due to camshaft belt or chain stretching. Nonetheless, it is still a popular method for research applications, especially if engine operation will primarily be steady state.

With encoder patterns, position 0 and tooth 0 correspond to the first tooth following the rising edge to the CamSig input.

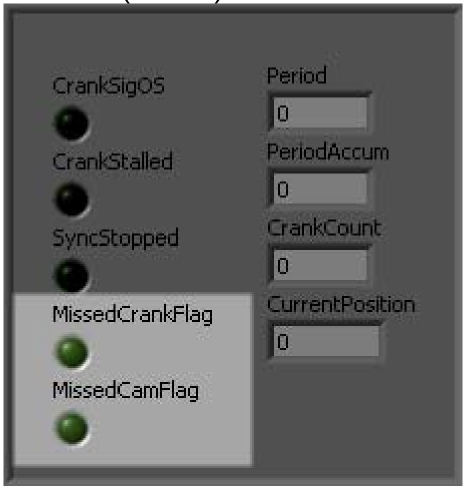
EPTControl (Cluster)



MissedCrankFlagClr (boolean): When TRUE, the MissedCrankFlag within the EPTData cluster will be cleared.

MissedCamFlagClr (boolean): When TRUE, the MissedCamFlag within the EPTData cluster will be cleared.

EPTData (Cluster)



Period: 0
PeriodAccum: 0
CrankCount: 0
CurrentPosition: 0

MissedCrankFlag (boolean): Set to TRUE upon the detection of a CamSig rising edge while the CrankCount value has not yet reached the expected value. In other words, the Cam tooth has been detected sooner than expected. This condition causes a loss of sync. Re-sync is not allowed until the flag is cleared by setting MissedCrankFlagClr to TRUE.

MissedCamFlag (boolean): Set to TRUE upon CrankCount being incremented past the expected value for the location of a CamSig rising edge before the CamSig rising edge is detected. In other words, the cam tooth was not received when expected. This condition causes a loss of sync. Re-sync is not allowed until the flag is cleared by setting MissedCamFlagClr to TRUE.

Examples

The following screen capture in Figure 10 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.

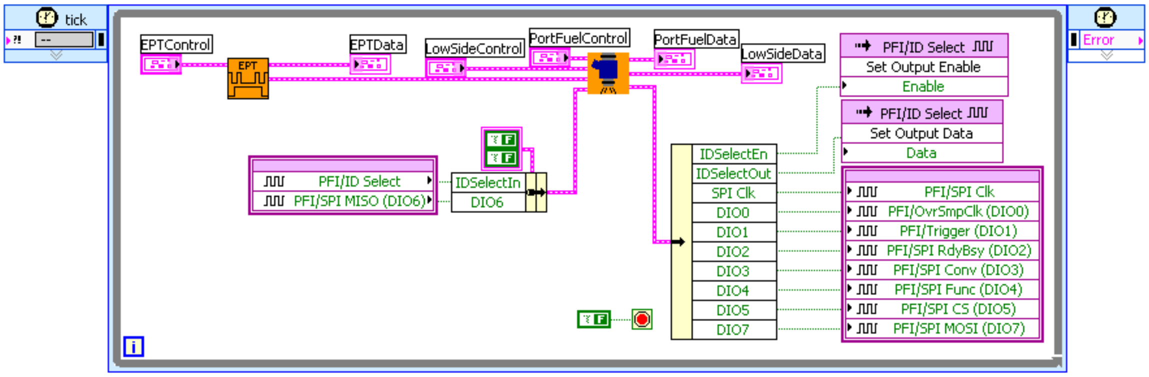


Figure 10. LabVIEW FPGA Block diagram example.