DRIVVEN

Engine Position Tracking VI User's Manual

D000031
D000033
D000035
Rev B

October 2009

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

Contents

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. The VIs provide a standardized interface to higher level LabVIEW RT VIs as well as the FPGA fuel and spark VIs which they supervise. 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 performing their function 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 are provided with an installer package which may be downloaded from Drivven's Sharepoint website after obtaining login access from Drivven. User's may go to http://portal.drivven.com/SoftwareDownload and enter the provided username and password to gain access to the specific product installer packages which have been purchased. The installer packages are executables which should be run on the intended development computer, having LabVIEW development tools installed. After installing the package, a "Start->Programs->Drivven->ProductRelease" menu item will be added to the desktop. The specific product will have an example LabVIEW project appear under the "Examples" menu and the user manual will appear under the "Manuals" menu. Users may open the example projects to experiment with the module or open and save the project as a starting point for a new application. All software files are installed to the vi.lib->addons->DrivvenProductRelease.

When working with application code, user's will notice a "Drivven" function palette added to the standard LabVIEW palette, specific for the RT or FPGA target. VIs for a specific Drivven product will be categorized according to product name. Also, certain Drivven products will install miscellaneous VIs under a common palette which is intended to be used across multiple products.

The 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 is discussed in detail in the Pattern-Specific Information section at the end of this manual. Pattern types are sold and supported separately. EPT VIs are optimized for their particular pattern to consume as few FPGA resources as possible.

The Drivven VIs require:

- LabVIEW 8.5 or later
- LabVIEW RT Module 8.5. or later
- LabVIEW FPGA Module 8.5 or later
- NI-RIO 2.4 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. The installer will place this file in the LabVIEW addons directory with the FPGA VI.

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.

Table 1. Supported EPT Patterns

Pattern Type	EPT VI Names	Pattern Description
N - M	ept_nm_vte7_revb.vi	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 during every other crank tooth gap.
N + 1	ept_p1_vte9_revb	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 during every other Plus1 tooth.
Encoder	ept_enc_vte2_revb ept_enc_vte4_revb ept_enc_vte6_revb ept_enc_vte8_revb	Encoder patterns have N evenly spaced teeth on the crankshaft with a separately sensed single reference (index) tooth on the crankshaft or camshaft. The rising edge of the reference must occur between the same two crankshaft teeth rising edges each cycle. The EPT ENC VI can be configured to track 360 or 720 degrees. 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 either on the crankshaft or on the camshaft. When located on the crankshaft, then an additional cam phase signal must be present in order to mask (using an FPGA AND gate) every other crank reference, such that every other crank reference is allowed to pass through to the CamSig input of the EPT VI. This would cause the crank encoder pattern to effectively generate twice the teeth between each reference tooth seen by the EPT VI. 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. There are 360 CAD per two stroke cycle or one crankshaft rotation. There are 720 CAD per 4-stroke cycle, or two crankshaft rotations. In a 4-stroke engine, the camshaft completes exactly one rotation per two rotations of the crankshaft. There are two strokes of the piston (up and down) within the cylinder during a single rotation of the crankshaft. A single stroke of the piston covers 180 CAD.

EXTRAP: Level of EPT Position Extrapolation. This is a fixed power of two by which the angular resolution of tracked crankshaft position is improved over the physical teeth alone. For example, if the EPT VI has a fixed extrapolation level of 7, then the crank angle resolution between each physical tooth is improved by a factor of 2^7 = 128.

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

MAX_CAT: Maximum Crank Angle Ticks per engine cycle.

EPT VI Commonality

EPT Input Signals

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 Boolean crankshaft and camshaft input. These signals may be wired from any Boolean logic source, such as 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, as shown below in figure 1. 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 to operate with a single trigger wheel mounted to a camshaft. This would require routing the camshaft trigger wheel sensor signal into the crankshaft input of the EPT VI. Please refer to the Pattern-Specific Information section at the end of this manual for details or contact Drivven for assistance with these configurations.

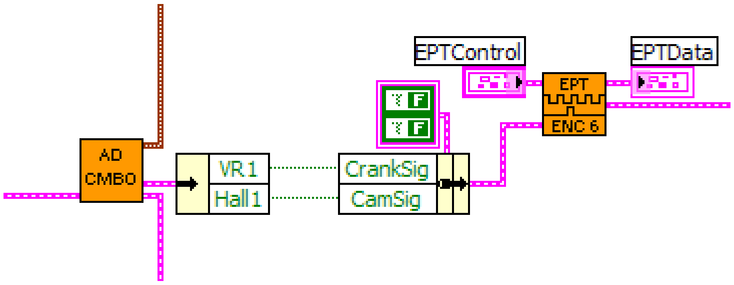


Figure 1. VR and Hall signals from the ADCombo wired to the CrankSig and CamSig inputs, using a LabVIEW Bundle By Name function.

Digital Signal Filtering

It is often necessary to implement filtering of the crank and/or cam signals in the FPGA prior to being wired to the EPT VI. Drivven provides an integrating filter VI which rejects positive and negative glitches in a Boolean signal having a pulse width less than the FilterValue, in terms of 40MHz clock ticks. The filt_int_reva.vi is described in detail in the EPT Extras section of this manual.

EPT Output Signals

During normal operation, all EPT VIs pass the CrankSig and CamSig from the EPTInSig cluster to the EPTOutSig cluster. These signals can be used in other FPGA processing or wired to FPGA output nodes to be monitored on an oscilloscope as shown below in figure 2.

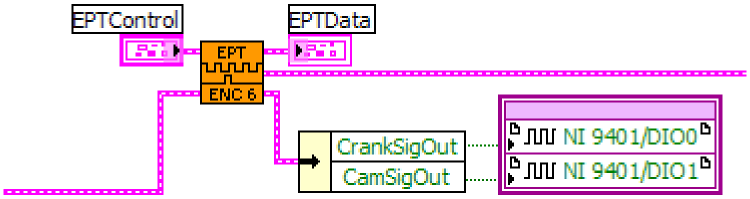


Figure 2. CrankSigOut and CamSigOut wire to NI 9401 digital output channels.

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 in FPGA clock ticks. These controls are found within the EPTControl cluster. The simulated crank and cam signals are output on the CrankSigOut and CamSigOut 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 internally 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. Figure 2 shows the simulated crank and cam signals wired to FPGA output nodes.

Watchdog

The WatchdogIn Boolean of the EPTControl cluster 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. The WatchdogIn can be toggled by a simple Boolean NOT function and feedback node within LabVIEW RT.

Achieving Sync

In order for position tracking, or sync, to start, a few conditions must be met. First, some pattern specific values must be set correctly within the EPTControl cluster, such as NumberOfCrankTeeth Stroke, CrankStartCount, CrankStallPeriod, etc. Next, the application must enable sync by setting SyncEnable to TRUE. Next, the crankshaft speed must surpass the stall speed. Technically, this means that the measured Period between crank teeth 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 CrankStalled 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 and the angular position of the crankshaft will be tracked. As sync is maintained, CrankCount and CurrentPosition of the EPTData cluster will show the latest crankshaft position with respect to absolute 0 position. Each EPT VI will have different rules for achieving and maintaining sync, as well as location for absolute position 0. This information is contained in the Pattern-Specific Information section at the end of this manual.

Period and PeriodAccum

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)

When calculating engine speed, be sure to check for Period=0 and PeriodAccum=0, when the crankshaft is stalled, otherwise a divide-by-zero condition will be introduced.

Drivven provides general RT sub 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 or PeriodAccum to RPM and internally accounts for Period=0. When converting Period to RPM, the PulsesPerRotation should be set to NumberOfCrankTeeth. When converting PeriodAccum to RPM, the PulsesPerRotation should be set to NumberOfCrankTeeth / PeriodAccumCount. For simulation purposes, the VI named speed2ticks.vi can be used to convert RPM set point to SimPeriod. Documentation for these sub VIs is provided in the General RT VIs user manual included with all Drivven software installations.

CrankCount

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 within the Pattern-Specific Information section at the end of this manual. 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 the pattern type is N-M, NumberOfCrankTeeth = 60 and Stroke = TRUE, then CrankCount will range from 0 to 119. CrankCount is not required to be wired to any other location. It is for monitoring purposes only, or may be used for addition processing beyond the scope of this document.

CurrentPosition

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 within the Pattern-Specific Information section at the end of this manual. The CurrentPosition 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 general RT sub VIs which can be implemented at the LabVIEW RT level for performing these calculations. The VI named CAT2CAD.vi can be used to convert CurrentPosition(CAT) to CAD. Documentation for these sub VIs is provided in the General RT VIs user manual included with all Drivven software installations.

CurrentPosition is not required to be wired to any other location. It is for monitoring purposes only, or may be used for addition processing beyond the scope of this document.

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, makes the EPT VIs a very robust position tracking algorithm.

EPT VI Selection

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 provided with four 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 EPT VI extrapolation value, shown in the VI icon, will affect the CAD resolution of CurrentPeriod. Drivven provides EPT VIs configured with extrapolation values so that approximately 0.1 CAD (or better) resolution is possible for each crank tooth count range. The worst case resolution is 0.35 CAD for very low encoder pattern 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 number of bits 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 CurrentPosition 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.

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

Pattern Type	NumberOfCrankTeeth	Stroke	Extrapolation	MAX_CAT	CAD / CAT
Encoder (max)	3600	4	2	28800	0.03
Encoder (min)	720	4	2	11520	0.13
Encoder (max)	720	4	4	46080	0.03
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 will lose sync under normal conditions if the crankshaft speed falls below the stall speed, as specified by the CrankStallPeriod input to the EPTControl cluster. The EPT VI will lose sync in error when missing or extra crank/cam pulses are encountered where unexpected. In all conditions, when sync is lost, CrankCount and CurrentPosition will be set to 0 and SyncStopped will be set to TRUE. A loss of sync due to an error condition will be further signaled by the appropriate pattern-specific error flag within the EPTData cluster. An error condition causing loss of sync will prevent any further sync until the error flag is cleared. 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. We will first discuss the two most common sources, which are electrical noise and mechanical trigger wheel imperfections. Then we will describe techniques to discover the trouble source.

Electrical Noise

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, provided with the EPT installer. The filter VI is called filt_int_reva.vi. The filter allows glitches up to a specified width to be rejected.

Mechanical Noise

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 arming 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.

Troubleshooting

Achieving reliable, clean crank and cam signals can sometimes be a trial and error process. It is Drivven's experience that reliable sync is often not achieved 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 signal margin for cam belt or chain stretch.

With these projects, it is always useful to have access to a multi-channel mixed signal oscilloscope to assist with locating the source of sync problems. Finding the source of the problem is most of the battle. If the EPT SyncStopped flag is routed out to external digital outputs, it can be used as an oscilloscope trigger 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 CrankSigOut, CamSigOut and SyncStopped signals out to digital outputs. These signals could be monitored on digital or analog scope channels. The raw external VR and/or hall-effect signals could also be monitored on the scope's analog channels. By triggering on the rising edge of SyncStopped, 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 CrankSigOut and CamSigOut. The resulting information can be used to determine the proper problem-solving action to take – whether it consists of external corrections in shielding or grounding, or internal filtering.

Sub VI Documentation

EPT VIs

The EPT 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.

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. The installer will place this file in the LabVIEW addons directory with the FPGA VI.

ept_nm_vte7_revb.vi

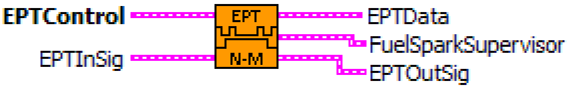
Engine Position Tracking VI

Pattern Type = N-M

Extrapolation Level = 7 Bits.

To be placed within a single cycle loop of an FPGA block diagram, executed at 40 MHz.
See the Pattern-Specific Information section in the EPT User Manual.

Connector Pane



Controls and Indicators

EPTControl Cluster of control parameters for configuring engine position tracking for a specific pattern type.

U16 **NumberOfCrankTeeth** Specifies the number of teeth on the crankshaft trigger wheel as if there were no missing or extra crank teeth. For example, a 60-2 crankshaft pattern would have NumberOfCrankTeeth = 60.

U8 **NumberOfMissingTeeth** 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.

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

U32 **CrankStallPeriod** 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. It should be noted that CrankStallPeriod must be able to account for the additional time created by the missing teeth. 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)

U16 **CrankCountStart** 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.

U16 **PeriodAccumCount** 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.

TF **AllowNoCamStart** 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.

TF **SyncEnable** 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.

TF **SimEnable** 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.

TF **MissedCrankFlagClr** When TRUE, the MissedCrankFlag within the EPTData cluster will be cleared.

TF **MissedGapFlagClr** When TRUE, the MissedGapFlag within the EPTData cluster will be cleared.

TF **NoCamFlagClr** When TRUE, the NoCamFlag within the EPTData cluster will be cleared.

U32 **SimPeriod** 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)

U16 **SimCrankSigDuration** 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 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. The WatchdogIn can be toggled by a simple NOT function and feedback node within LabVIEW RT.

EPTInSig Cluster of crank and cam input signals used by the EPT for tracking engine position.

CrankSig The digital input signal representing the crankshaft trigger wheel pulse train. This input should be wired from a digital input module with signal conditioning or a Drivven VR or Hall input channel of the AD Combo or VR/Hall Module Kit VI. Additional filtering or modification may be inserted before the signal is wired to the EPT VI. The EPT VI responds to the rising edge of CrankSig.

CamSig The digital input signal representing the camshaft trigger wheel pulse train. This input should be wired from a digital input module with signal conditioning or a Drivven VR or Hall input channel of the AD Combo or VR/Hall Module Kit VI. Additional filtering or modification may be inserted before the signal is wired to the EPT VI. The EPT Encoder VI responds to the rising edge of CamSig. The EPT N-M and N+1 VI test the level of CamSig.

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

EPTData Cluster of status information related to engine position tracking.

CrankSigOS 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 Set to TRUE when the engine speed is below stall speed.

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

MissedCrankFlag 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 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 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.

U32 **Period** 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)

U32 **PeriodAccum** 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 * NumberOfCrankTeeth / PeriodAccumCount)

U16 **CrankCount** The latest tooth count referenced to crank tooth 0. Tooth 0 is determined according to the pattern that is being tracked and is discussed within the pattern specific 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.

U16 **CurrentPosition** During sync, the EPT provides the latest crankshaft position 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 Information section at the end of the EPT user manual. The CurrentPosition 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.

EPTOutSig Cluster of crank and cam outputs. These signals are connected directly to the inputs during normal operation, and connected to the simulated signals while SimEnable is TRUE.

TF **CrankSigOut** Digital output signal originating from two different sources. If EPT simulation is disabled, then the crank and cam signal inputs are passed directly through to these outputs. If EPT simulation is enabled, then the simulated signals are output here.

TF **CamSigOut** Digital output signal originating from two different sources. If EPT simulation is disabled, then the crank and cam signal inputs are passed directly through to these outputs. If EPT simulation is enabled, then the simulated signals are output here.

ept_p1_vte9_revb.vi

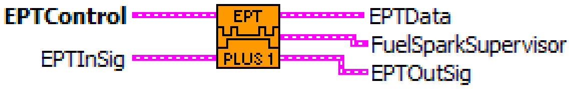
Engine Position Tracking VI

Pattern Type = N+1

Extrapolation Level = 9 Bits.

To be placed within a single cycle loop of an FPGA block diagram, executed at 40 MHz.
See the Pattern-Specific Information section in the EPT User Manual.

Connector Pane



Controls and Indicators

EPTControl Cluster of control parameters for configuring engine position tracking for a specific pattern type.

U16 **NumberOfCrankTeeth** Specifies the number of teeth on the crankshaft trigger wheel as if there were no missing or extra crank teeth. For example, a 60-2 crankshaft pattern would have NumberOfCrankTeeth = 60.

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

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

U32 **CrankStallPeriod** 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)

U16 **CrankCountStart** 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.

U16 **PeriodAccumCount** 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.

TF **AllowNoCamStart** 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.

TF **SyncEnable** 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.

TF **SimEnable** 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.

TF **MissedCrankFlagClr** When TRUE, the MissedCrankFlag within the EPTData cluster will be cleared.

TF **MissedPlus1FlagClr** When TRUE, the MissedPlus1Flag within the EPTData cluster will be cleared.

TF **NoCamFlagClr** When TRUE, the NoCamFlag within the EPTData cluster will be cleared.

U32 **SimPeriod** 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)

U16 **SimCrankSigDuration** 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)

TF **WatchdogIn** 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. The WatchdogIn can be toggled by a simple NOT function and feedback node within LabVIEW RT.

EPTInSig Cluster of crank and cam input signals used by the EPT for tracking engine position.

TF **CrankSig** The digital input signal representing the crankshaft trigger wheel pulse train. This input should be wired from a digital input module with signal conditioning or a Drivven VR or Hall input channel of the AD Combo or VR/Hall Module Kit VI. Additional filtering or modification may be inserted before the signal is wired to the EPT VI. The EPT VI responds to the rising edge of CrankSig.

TF **CamSig** The digital input signal representing the camshaft trigger wheel pulse train. This input should be wired from a digital input module with signal conditioning or a Drivven VR or Hall input channel of the AD Combo or VR/Hall Module Kit VI. Additional filtering or modification may be inserted before the signal is wired to the EPT VI. The EPT Encoder VI responds to the rising edge of CamSig. The EPT N-M and N+1 VI test the level of CamSig.

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

EPTData Cluster of status information related to engine position tracking.

TF **CrankSigOS** 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.

TF **CrankStalled** Set to TRUE when the engine speed is below stall speed.

TF **SyncStopped** Set to TRUE while position tracking is not taking place. In other words, no sync.

TF **MissedCrankFlag** 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.

TF **MissedPlus1Flag** 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.

TF **NoCamFlag** 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.

U32 **Period** 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)

U32 **PeriodAccum** 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 * NumberOfCrankTeeth / PeriodAccumCount)

U16 **CrankCount** The latest tooth count referenced to crank tooth 0. Tooth 0 is determined according to the pattern that is being tracked and is discussed within the pattern specific 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.

U16 **CurrentPosition** During sync, the EPT provides the latest crankshaft position 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 Information section at the end of the EPT user manual. The CurrentPosition 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.

EPTOutSig Cluster of crank and cam outputs. These signals are connected directly to the inputs during normal operation, and connected to the simulated signals while SimEnable is TRUE.

TF **CrankSigOut** Digital output signal originating from two different sources. If EPT simulation is disabled, then the crank and cam signal inputs are passed directly through to these outputs. If EPT simulation is enabled, then the simulated signals are output here.

TF **CamSigOut** Digital output signal originating from two different sources. If EPT simulation is disabled, then the crank and cam signal inputs are passed directly through to these outputs. If EPT simulation is enabled, then the simulated signals are output here.

ept_enc_vte2_revb.vi
ept_enc_vte4_revb.vi
ept_enc_vte6_revb.vi
ept_enc_vte8_revb.vi

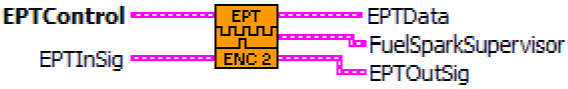
Engine Position Tracking VI

Pattern Type = Encoder

Extrapolation Level = 2, 4, 6, 8 Bits.

To be placed within a single cycle loop of an FPGA block diagram, executed at 40 MHz.
See the Pattern-Specific Information section in the EPT User Manual.

Connector Pane



Controls and Indicators

EPTControl Cluster of control parameters for configuring engine position tracking for a specific pattern type.

U16 **NumberOfCrankTeeth** Specifies the number of teeth on the crankshaft trigger wheel as if there were no missing or extra crank teeth. For example, a 60-2 crankshaft pattern would have NumberOfCrankTeeth = 60.

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

U32 **CrankStallPeriod** 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)

U16 **CrankCountStart** 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.

U16 **PeriodAccumCount** 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.

TF **SyncEnable** 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.

TF **SimEnable** 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.

MissedCrankFlagClr When TRUE, the MissedCrankFlag within the EPTData cluster will be cleared.

MissedCamFlagClr When TRUE, the MissedCamFlag within the EPTData cluster will be cleared.

SimPeriod 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 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 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. The WatchdogIn can be toggled by a simple NOT function and feedback node within LabVIEW RT.

EPTInSig Cluster of crank and cam input signals used by the EPT for tracking engine position.

CrankSig The digital input signal representing the crankshaft trigger wheel pulse train. This input should be wired from a digital input module with signal conditioning or a Drivven VR or Hall input channel of the AD Combo or VR/Hall Module Kit VI. Additional filtering or modification may be inserted before the signal is wired to the EPT VI. The EPT VI responds to the rising edge of CrankSig.

CamSig The digital input signal representing the camshaft trigger wheel pulse train. This input should be wired from a digital input module with signal conditioning or a Drivven VR or Hall input channel of the AD Combo or VR/Hall Module Kit VI. Additional filtering or modification may be inserted before the signal is wired to the EPT VI. The EPT Encoder VI responds to the rising edge of CamSig. The EPT N-M and N+1 VI test the level of CamSig.

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

EPTData Cluster of status information related to engine position tracking.

TF **CrankSigOS** 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.

TF **CrankStalled** Set to TRUE when the engine speed is below stall speed.

TF **SyncStopped** Set to TRUE while position tracking is not taking place. In other words, no sync.

TF **MissedCrankFlag** 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.

TF **MissedCamFlag** 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.

U32 **Period** 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)

U32 **PeriodAccum** 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 * NumberOfCrankTeeth / PeriodAccumCount)

U16 **CrankCount** The latest tooth count referenced to crank tooth 0. Tooth 0 is determined according to the pattern that is being tracked and is discussed within the pattern specific 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.

U16 **CurrentPosition** During sync, the EPT provides the latest crankshaft position 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 Information section at the end of the EPT user manual. The CurrentPosition 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.

EPTOutSig Cluster of crank and cam outputs. These signals are connected directly to the inputs during normal operation, and connected to the simulated signals while SimEnable is TRUE.

CrankSigOut Digital output signal originating from two different sources. If EPT simulation is disabled, then the crank and cam signal inputs are passed directly through to these outputs. If EPT simulation is enabled, then the simulated signals are output here.

CamSigOut Digital output signal originating from two different sources. If EPT simulation is disabled, then the crank and cam signal inputs are passed directly through to these outputs. If EPT simulation is enabled, then the simulated signals are output here.

EPT Extras

The EPT Extras 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 sub 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. The installer will place this file in the LabVIEW addons directory with the FPGA VI.

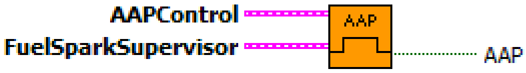
Each EPT installation includes a tool palette group called EPT Extras. This group includes several sub VIs which can assist with more complex engine control tasks, such as performing calculations only during certain crank-angle windows, or capturing the crank-angle position at which an event occurs. Below are descriptions of each sub VI.

aap_vt_reva.vi

Angle-Angle-Pulse. This VI accepts the FuelSparkSupervisor from the EPT VI. When enabled, it generates an output pulse according to a StartPosition match (rising edge) and an EndPosition match (falling edge). It is assumed that StartPosition and EndPosition will not change dynamically.

Depending on the value of CurrentPosition from the EPT VI, changing StartPosition or EndPosition while enabled and in sync could result in an AAP pulse being missed, a double pulse, or a pulse which remains high until the next cycle. Fuel and spark outputs handle dynamic timing gracefully but require more FPGA resources. The AAP VI is designed to require very little resources. If StartPosition or EndPosition must be changed dynamically and the above issues must be avoided, then StartPosition and EndPosition should be changed at a known safe value of CurrentPosition in the cycle, or the AAP VI should be temporarily disabled.

Connector Pane



Controls and Indicators

FuelSparkSupervisor This cluster input must be wired directly from the FuelSparkSupervisor cluster output of a Drivven EPT VI. The FuelSparkSupervisor cluster from a single EPT may be used to supervise any number of Drivven engine-synchronous output sub-VIs.

AAPControl Control cluster of parameters for configuring the angle-angle-pulse.

TF **AAPEnable** If AAPEnable is TRUE then AAP is enabled to operate according to the specified StartPosition and EndPosition. If AAPEnable is FALSE then AAP is forced low at all times.

U16 **StartPosition** Angle-angle-pulses are generated with a leading edge coinciding with StartPosition. The units of StartPosition are CAT. Drivven provides Offset2CAT.vi in the General RT VIs in the RT function palette. This VI can be implemented at the LabVIEW RT level for converting an angle in CAD, with respect to a cylinder TDC, to CAT.

U16 **EndPosition** Angle-angle-pulses are generated with a trailing edge coinciding with EndPosition. The units of EndPosition are CAT. Drivven provides Offset2CAT.vi in the General RT VIs in the RT function palette. This VI can be implemented at the LabVIEW RT level for converting an angle in CAD, with respect to a cylinder TDC, to CAT.

TF **AAP** AAP output is generated with a leading edge coinciding with StartPosition and a trailing edge coinciding with EndPosition. AAP defaults to FALSE when the function is disabled or when the EPT SyncStopped Boolean is TRUE.

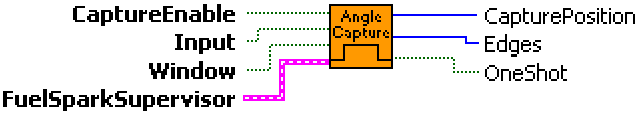
angle_capture_reva.vi

If CaptureEnable is TRUE and the EPT VI is in sync, then CapturePosition will be updated with EPT CurrentPosition upon the rising edge of Input while Window is TRUE. If CaptureEnable is FALSE or the EPT VI is not in sync, then CapturePosition will indicate 0.

Edges reports the number of rising edges captured while Window is TRUE. If CaptureEnable is FALSE or the EPT VI is not in sync, then Edges will indicate 0.

OneShot is a one-clock otuput pulse upon the rising edge of Input while Window is TRUE. OneShot can be used to trigger an interrupt or FIFO write.

Connector Pane



Controls and Indicators

CaptureEnable If CaptureEnable is TRUE and the EPT VI is in sync, then CapturePosition will be updated with EPT CurrentPosition upon the rising edge of Input while Window is TRUE. If CaptureEnable is FALSE or the EPT VI is not in sync, then CapturePosition will indicate 0.

Edges reports the number of rising edges captured while Window is TRUE. If CaptureEnable is FALSE or the EPT VI is not in sync, then Edges will indicate 0.

OneShot is a one-clock otuput pulse upon the rising edge of Input while Window is TRUE. OneShot can be used to trigger an interrupt or FIFO write.

Input If CaptureEnable is TRUE and the EPT VI is in sync, then CapturePosition will be updated with EPT CurrentPosition upon the rising edge of Input while Window is TRUE. If CaptureEnable is FALSE or the EPT VI is not in sync, then CapturePosition will indicate 0.

Edges reports the number of rising edges captured while Window is TRUE. If CaptureEnable is FALSE or the EPT VI is not in sync, then Edges will indicate 0.

OneShot is a one-clock otuput pulse upon the rising edge of Input while Window is TRUE. OneShot can be used to trigger an interrupt or FIFO write.

Window If CaptureEnable is TRUE and the EPT VI is in sync, then CapturePosition will be updated with EPT CurrentPosition upon the rising edge of Input while Window is TRUE. If CaptureEnable is FALSE or the EPT VI is not in sync, then CapturePosition will indicate 0.

Edges reports the number of rising edges captured while Window is TRUE. If CaptureEnable is FALSE or the EPT VI is not in sync, then Edges will indicate 0.

OneShot is a one-clock otuput pulse upon the rising edge of Input while Window is TRUE. OneShot can be used to trigger an interrupt or FIFO write.

FuelSparkSupervisor This cluster input must be wired directly from the FuelSparkSupervisor cluster output of a Drivven EPT VI. The FuelSparkSupervisor cluster from a single EPT may be used to supervise any number of Drivven engine-synchronous output sub-VIs.

U16 **CapturePosition** If CaptureEnable is TRUE and the EPT VI is in sync, then CapturePosition will be updated with EPT CurrentPosition upon the rising edge of Input while Window is TRUE. If CaptureEnable is FALSE or the EPT VI is not in sync, then CapturePosition will indicate 0.

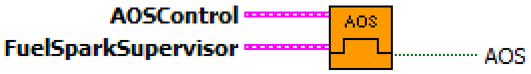
U8 **Edges** Edges reports the number of rising edges captured while Window is TRUE. If CaptureEnable is FALSE or the EPT VI is not in sync, then Edges will indicate 0.

TF **OneShot** OneShot is a one-clock otuput pulse upon the rising edge of Input while Window is TRUE. OneShot can be used to trigger an interrupt or FIFO write.

angle_one_shot_reva.vi

This VI accepts the FuelSparkSupervisor from the EPT VI. When enabled, it generates a one-clock one-shot pulse at the Position specified in terms Crank Angle Ticks (CAT).

Connector Pane



Controls and Indicators

FuelSparkSupervisor This cluster input must be wired directly from the FuelSparkSupervisor cluster output of a Drivven EPT VI. The FuelSparkSupervisor cluster from a single EPT may be used to supervise any number of Drivven engine-synchronous output sub-VIs.

AOSControl Control cluster of parameters for configuring the angle-one-shot.

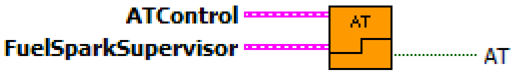
> **AOSEnable** If AOSEnable is TRUE then AOS is enabled to operate according to the specified Position. If AOSEnable is FALSE then AOS is forced low at all times.
>
> **Position** Angle-one-shots are generated with a leading edge coinciding with Position and having a duration of one clock tick. The units of Position are CAT. Drivven provides Offset2CAT.vi in the General RT VIs in the RT function palette. This VI can be implemented at the LabVIEW RT level for converting an angle in CAD, with respect to a cylinder TDC, to CAT.

AOS AOS output is generated with a leading edge coinciding with Position and a duration of one clock tick. AOS defaults to FALSE when the function is disabled or when the EPT SyncStopped Boolean is TRUE.

angle_toggle_reva.vi

This VI accepts the FuelSparkSupervisor from the EPT VI. When enabled, it toggles the output at the angular Position specified in Crank Angle Ticks (CAT).

Connector Pane



Controls and Indicators

FuelSparkSupervisor This cluster input must be wired directly from the FuelSparkSupervisor cluster output of a Drivven EPT VI. The FuelSparkSupervisor cluster from a single EPT may be used to supervise any number of Drivven engine-synchronous output sub-VIs.

ATControl Control cluster of parameters for configuring the angle-toggle.

TF **ATEnable** If ATEnable is TRUE then AT is enabled to toggle according to the specified Position. If ATEnable is FALSE then AT will remain at its previous state when the specified Position is reached.

U16 **Position** AT is toggled coinciding with Position. The units of Position are CAT. Drivven provides Offset2CAT.vi in the General RT VIs in the RT function palette. This VI can be implemented at the LabVIEW RT level for converting an angle in CAD, with respect to a cylinder TDC, to CAT.

TF **AT** AT output is toggled coinciding with Position. AT remains at its previous state when the function is disabled or when the EPT SyncStopped Boolean is TRUE. AT defaults to FALSE when the FPGA is reset.

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 during 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 perform diagnostics. 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) and CrankCount = NumberOfCrankTeeth.

It is allowed for CamSig to have more than one pulse per cycle, as long as the signal is high and low during the correct crank tooth gap for the correct phase of the engine. In some applications CamSig may only be a single short pulse which is not high during either crank tooth gaps. In this case, the Drivven signal_extend_reva.vi may be used to extend the CamSig for a specified number of crank teeth in order to cover one of the gaps. The signal extension VI is documented in the EPT Extras Sub VIs section of this manual.

Refer to the SubVI documentation for descriptions of the pattern-specific error flags.

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 during 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. As another example, if the Plus1 tooth is retarded, Stroke = FALSE and NumberOfCrankTeeth = 6, then CrankCount would increment from 0 to 5, such that tooth 5 would correspond to the tooth immediately after the Plus1 tooth and tooth 0 would correspond to the second tooth 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 NoCamFlag will be set to TRUE. The CPU is then able to perform diagnostics. 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 3 and 4 below.

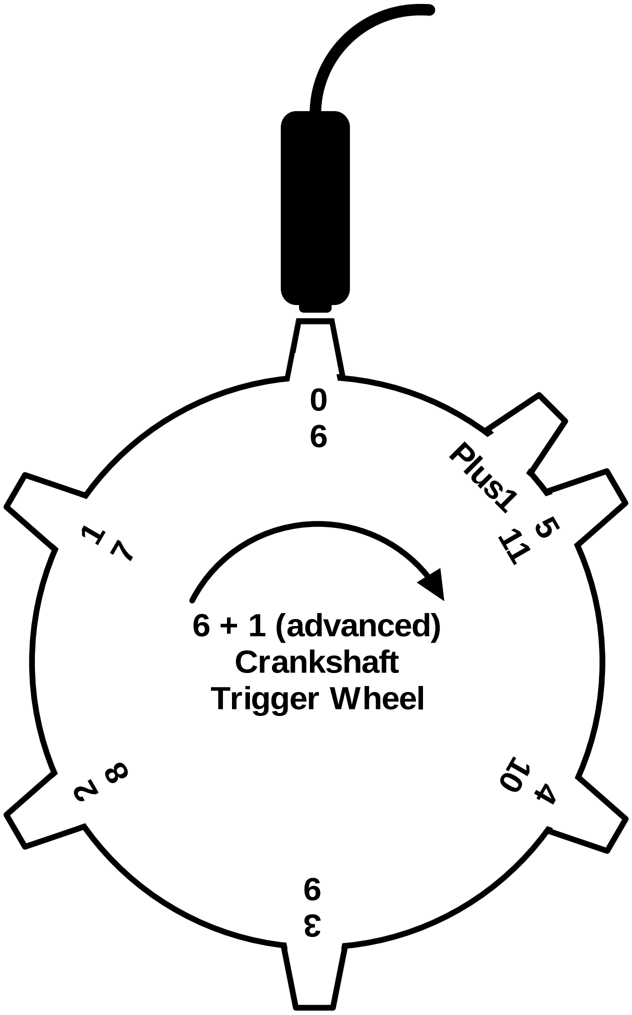


Figure 3. 6+1 (advanced) crankshaft trigger wheel tooth numbering

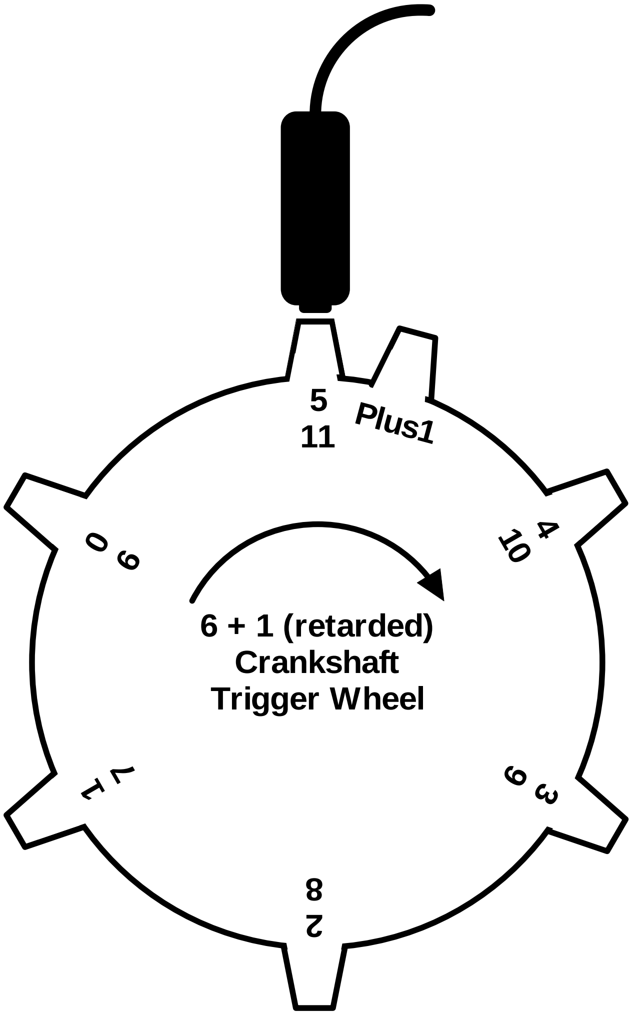


Figure 4. 6+1 (retarded) crankshaft trigger wheel tooth numbering

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) and CrankCount = NumberOfCrankTeeth.

Encoder Patterns

Pattern Description:
Encoder patterns have N evenly spaced teeth on the crankshaft with a separately sensed single reference (index) tooth on the crankshaft or camshaft. The rising edge of the reference must occur between the same two crankshaft teeth rising edges each cycle. The EPT ENC VI can be configured to track 360 or 720 degrees. 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 either on the crankshaft or on the camshaft. When located on the crankshaft, then an additional cam phase signal must be present in order to mask (using an FPGA AND gate) every other crank reference, such that every other crank reference is allowed to pass through to the CamSig input of the EPT VI. This would cause the crank encoder pattern to effectively generate twice the teeth between each reference tooth seen by the EPT VI. 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 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 raw 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 typically have 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, after masking by a cam phase signal. Encoders having pulse-counts of 360 to 3600 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, depending on the EPT extrapolation level.

If the optical encoder is mounted to the crankshaft, then NumberOfCrankTeeth should be set to the encoder pulse-count and Stroke should be set according to the engine type – 2-stroke (FALSE) or 4-stroke (TRUE).

If the engine is a 4-stroke, then a cam phase signal must be present in order to be logically gated (AND gate) with the Z pulse, within the FPGA, so that every other Z pulse is allowed to pass through. The result will be that the EPT VI will see two full rotations of A/B pulses for every Z pulse. The cam phase signal may be a complex pattern, as long as it provides a consistent TRUE state for one Z pulse and a FALSE state for the next, repeating this pattern. Drivven provides FPGA Vis for manipulating Boolean signals (extension, offset, delay, etc) in the EPT Extras tool palette in the FPGA. These can be used to adjust the cam phase signal in order to meet the proper gating requirement of the Z pulse. If the cam signal is too complex, then another EPT block may be required to track its position independently. Then a pseudo cam phase signal may be generated within the FPGA from matches to the cam position.

If the engine is a 2-stroke, then the Z pulse should be wired directly to the EPT CamSig input.

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 rising edge of the first CrankSig tooth following the rising edge of the CamSig input.

Figure 5 below shows the block diagram from the EPT Encoder example FPGA block diagram which implements cam phase signal extension and Z pulse offset operations from the sub VIs included in the EPT Extras. Figure 6 shows a plot of the relevant raw and modified signals. This block diagram and plot can be seen in detail within the EPT Encoder example project.

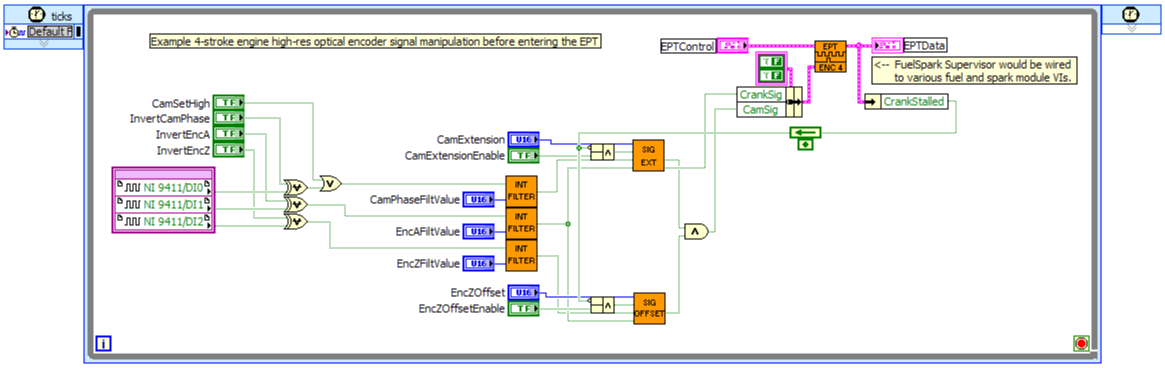


Figure 5. EPT Encoder example block diagram

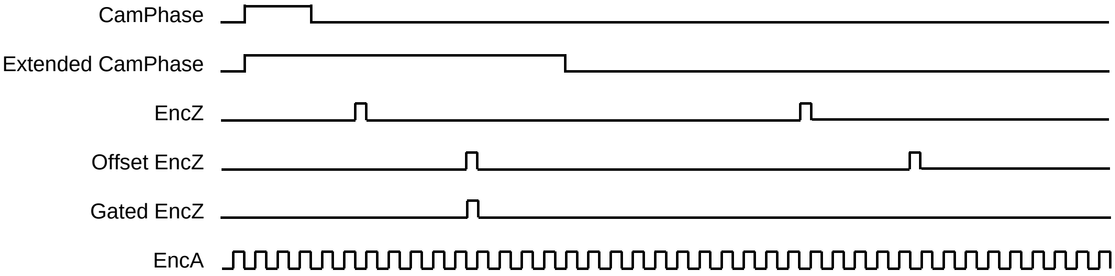


Figure 6. Raw CamPhase and EncZ, along with their modified versions with respect to EncA