Application Report
SPRABX2A–August 2014–Revised October 2018

CW/CCW Support on the C2000™ eQEP Module

Zain Sheikh and Brett Larimore

ABSTRACT

This application report describes two methods of interfacing a pulse train output (PTO) of clockwise and counterclockwise (CW/CCW) signals with the C2000 enhanced quadrature encoder pulse (eQEP) module. The eQEP module can be used to count the number of pulses, note the direction of motion, and calculate other variables such as position and velocity using the input signals.

Contents

List of Figures

List of Tables

Trademarks

C2000 is a trademark of Texas Instruments.

1 Introduction

1.1 PTO Overview

A PTO is generally output by a programmable logic controller (PLC) in industrial automation for motion control. These outputs can be used in conjunction with stepper or servo drives for controlling motors and other motion-based hardware. The purpose of this document is to provide the means to support interfacing the C2000 eQEP module with clockwise and counterclockwise PTO signals.

1.2 CW/CCW Signals Overview

CW/CCW signals are often used to provide the position reference to a stepper/servo motor drive. At any given time, no more than one of these two signals should be pulsing. Any pulse received on the CW signal is interpreted as a command for a positive increment of the position reference by the drive. Similarly, any pulse received on the CCW signal is interpreted as a command for a negative increment of the position reference.

CW/CCW signals can either be active high or low. With active-high CW/CCW signals, the idle signal would be low when not pulsing (see Figure 1). Similarly, with active-low CW/CCW signals, the idle signal would be high when not pulsing (see Figure 2).

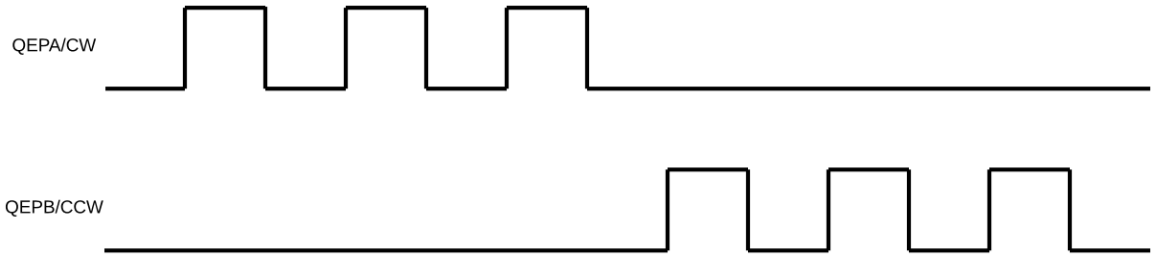

Figure 1. CW/CCW Active-High Signal Diagram

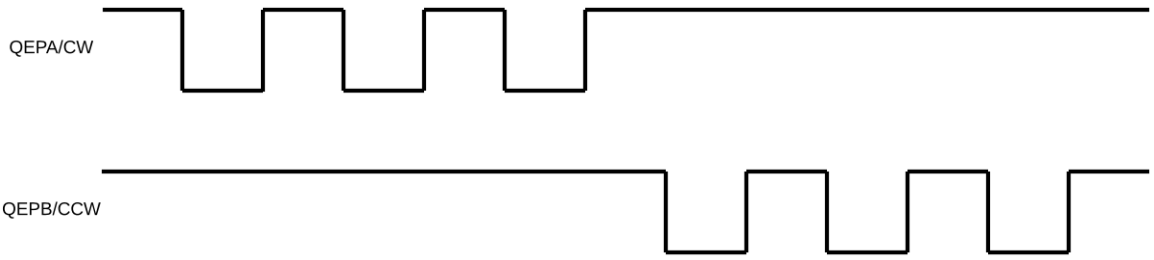

Figure 2. CW/CCW Active-Low Signal Diagram

2 eQEP Supported Counting Modes

All C2000 eQEP modules support two basic modes of operation. The Type-1 eQEP module added a new Quadrature Mode Adapter (QMA) module specifically designed to receive CW/CCW signals. The first operation mode is direction-count mode. In this mode, the accepted signals used as inputs are a clock signal and a direction signal. In this mode, the clock signal pulses while the direction signal indicates the direction of rotation. In many conventions, the clock signal is simply denoted as “pulse.”

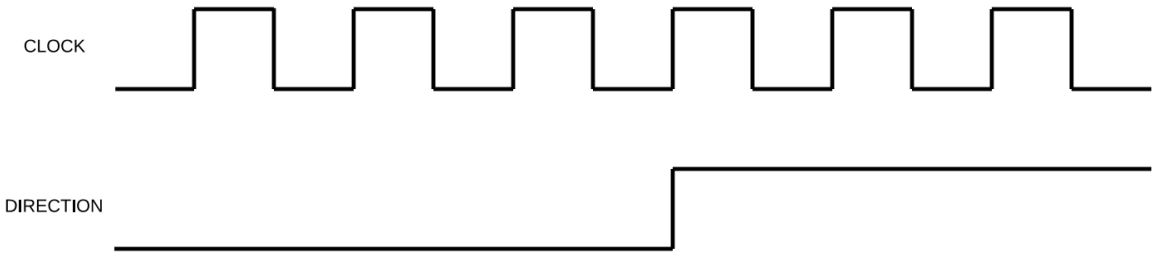

Figure 3. Direction-Count Diagram

The second operation mode is quadrature-clock mode, which is commonly used in interfacing with incremental encoders. In this mode, the accepted input signals used are two clock signals that are 90° out of phase. If these two signals are denoted as QEPA and QEPB, the interpreted clock (QCLK) is based off of the rising and falling edge of either QEPA or QEPB. The interpreted direction (QDIR) is generated by the relationship between the rising and falling edges of both signals. For example, if QEPA rises and then QEPB rises, the direction is clockwise.

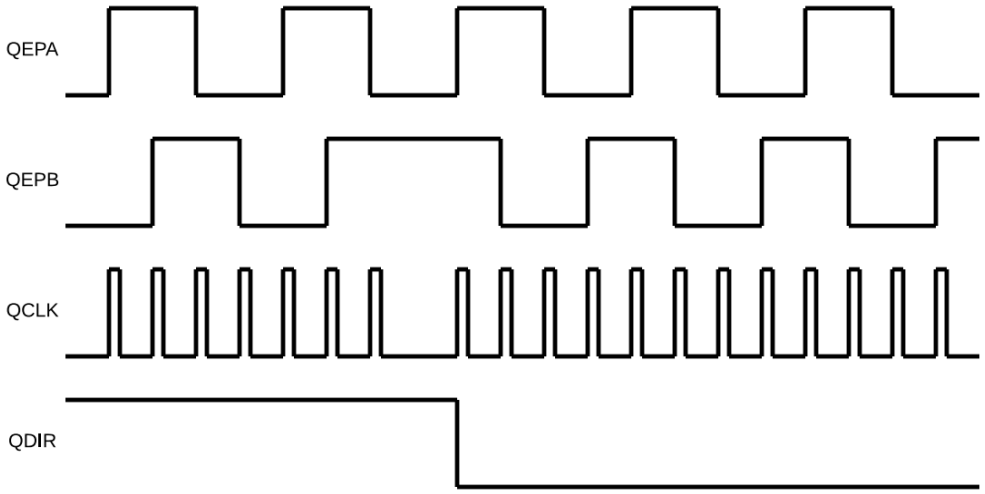

Figure 4. Quadrature-Clock Diagram

The third operation mode, introduced on the Type-1 eQEP module, is capable of directly receiving CW/CCW signals output by PLC machines. These signals were previously described and shown in Section 1.2 in Figure 1 and Figure 2.

3 External Logic Interfacing

One method of using CW/CCW signals with the eQEP module is by converting them into a clock signal and a direction signal. This can be achieved by interfacing the signals with some external logic. The external logic can be categorized into three separate pieces: clock generation, direction generation, and fault detection. With the clock and direction signals, the supported direction-count mode can be used.

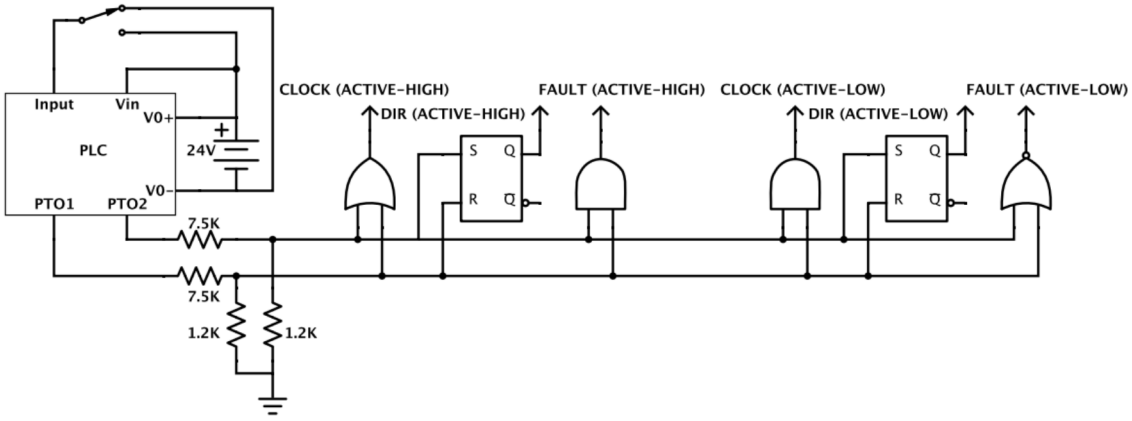

Figure 5. External Logic Interface Diagram

Figure 5 describes the configuration for the external logic. PTO outputs are generally 24 V. Because C2000 devices are 3.3 V logic, they must be scaled down to 3.3 V. Diodes could be added for further protection. After this, they are fed into the external logic where active-high and active-low clock, direction, and fault outputs are generated. These outputs are now ready to be interfaced with the eQEP module in direction-count mode.

3.1 Clock Generation

Clock generation logic uses two different types of gates depending on whether the clockwise and counterclockwise signals provided are active-high or active-low. For active-high signals an OR-gate should be used. The reasoning behind this is that anytime either of the signals is high, a pulse needs to be generated for a direction regardless of whether it is clockwise or counterclockwise.

For active-low signals an AND-gate is used. The reasoning behind this is anytime either of the signals is low, a pulse needs to be generated. This means that the only time the clock signal is high is if both the input signals are idle.

3.2 Direction Generation

The direction generation logic uses 3-state S-R latches. For active-high signals a NOR latch is used, and for active-low signals a NAND latch is used. This design choice has to do with maintaining your direction when no signal is being toggled. Although the latches do not provide for an explicit error, there is an output in both configurations that you can choose to avoid, which is denoted as an error state.

It is notable that the convention for clockwise and counterclockwise direction that the eQEP module uses is as follows: clockwise (up-count) is 1 and counterclockwise (down-count) is 0. This convention may be different for PLCs that generate the PTO signals.

Table 1. Active-High Direction Generation

	S (QEPB – CCW)	R (QEPA – CW)	Q (DIR Output)
Neither signal high	0	0	NO CHANGE
CW signal high	0	1	0
CCW signal high	1	0	1
Both signals high	1	1	ERROR

Table 2. Active-Low Direction Generation

	S (QEPB – CCW)	R (QEPA – CW)	Q (DIR Output)
Both signals low	0	0	ERROR
CCW signal low	0	1	1
CW signal low	1	0	0
Neither signal low	1	1	NO CHANGE

3.3 Fault Detection

Fault detection logic has to do with ensuring that both signals are never simultaneously high when using active-high signals or that both signals are never simultaneously low when using active-low signals. This implicitly prevents the states that have been denoted as ERROR in the direction generation logic from occurring. This is accomplished by using an AND-gate for active high signals, and a NOR-gate for active-low signals. The output of this logic could be used as a flag that is raised when an error occurs. This flag could then further be used as an external interrupt to alert the system that an error has occurred and halt eQEP counting.

3.4 Timing Issues

It is also notable that the propagation delay for an S-R latch may be larger than the propagation delay of an AND-gate or an OR-gate. For this reason, the clock can be generated prior to the direction being updated when a change occurs. In this case, it may be necessary to delay the clock output from the external logic in excess of 50 ns.

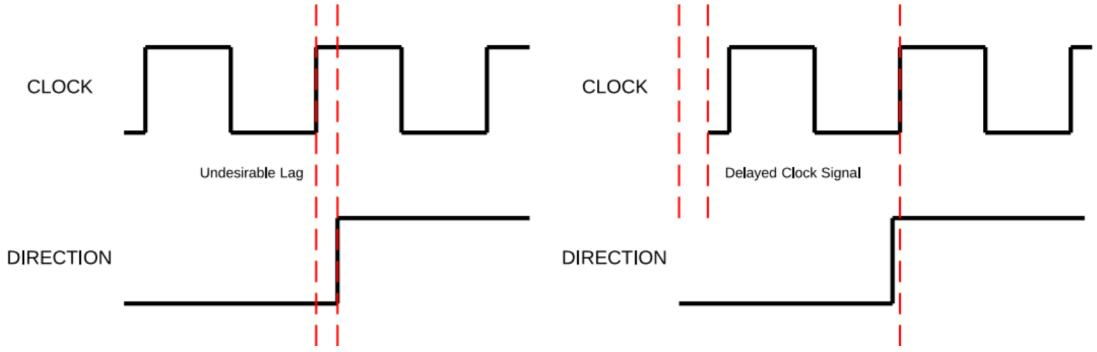

Figure 6. Delayed Clock Signal Diagram

3.5 Software Example

Configure the software project appropriately and initialize it with the proper settings in order to use the eQEP modules with this new interfacing method. The following configures and initializes one QEP for interfacing CW/CCW signals with the eQEP module using external logic. The following GPIO configuration code is applicable to F2837xD, F2837xS, and F2807x devices. Refer to the GPIO multiplexing options for your device if different GPIOs are to be used.

3.5.1 Module Configuration

It is necessary to disable the pull-up resistor connected to each pin:

```
GpioCtrlRegs.GPAPUD.bit.GPIO20 = 1;      // Disable GPIO20 pull-up (EQEP1A)
GpioCtrlRegs.GPAPUD.bit.GPIO21 = 1;      // Disable GPIO21 pull-up (EQEP1B)
GpioCtrlRegs.GPAPUD.bit.GPIO22 = 1;      // Disable GPIO22 pull-up (EQEP1S)
GpioCtrlRegs.GPAPUD.bit.GPIO23 = 1;      // Disable GPIO23 pull-up (EQEP1I)
```

The GPIOs have to be synchronized with the system clock:

```
GpioCtrlRegs.GPAQSEL2.bit.GPIO20 = 0;    // Sync GPIO20 to SYSCLK (EQEP1A)
GpioCtrlRegs.GPAQSEL2.bit.GPIO21 = 0;    // Sync GPIO21 to SYSCLK (EQEP1B)
GpioCtrlRegs.GPAQSEL2.bit.GPIO22 = 0;    // Sync GPIO22 to SYSCLK (EQEP1S)
GpioCtrlRegs.GPAQSEL2.bit.GPIO23 = 0;    // Sync GPIO23 to SYSCLK (EQEP1I)
```

Each GPIO that is going to be used has to be configured as an eQEP input:

```
GpioCtrlRegs.GPAGMUX2.bit.GPIO20 = 0;    // Configure GPIO20 as EQEP1A
GpioCtrlRegs.GPAGMUX2.bit.GPIO21 = 0;    // Configure GPIO21 as EQEP1B
GpioCtrlRegs.GPAGMUX2.bit.GPIO22 = 0;    // Configure GPIO22 as EQEP1S
GpioCtrlRegs.GPAGMUX2.bit.GPIO23 = 0;    // Configure GPIO23 as EQEP1I
GpioCtrlRegs.GPAMUX2.bit.GPIO20 = 1;     // Configure GPIO20 as EQEP1A
GpioCtrlRegs.GPAMUX2.bit.GPIO21 = 1;     // Configure GPIO21 as EQEP1B
GpioCtrlRegs.GPAMUX2.bit.GPIO22 = 1;     // Configure GPIO22 as EQEP1S
GpioCtrlRegs.GPAMUX2.bit.GPIO23 = 1;     // Configure GPIO23 as EQEP1I
```

Now that the eQEP module has been properly configured, move onto initializing the eQEP module.

3.5.2 Module Initialization

Do the following to initialize the eQEP module,:

```
void InitEQep1Example(void)
{

   EQep1Regs.QDECCTL.bit.QSRC=1;       // QEP direction count mode
   EQep1Regs.QDECCTL.bit.XCR=1;        // x1 resolution

   EQep1Regs.QEPCTL.bit.FREE_SOFT=2;

   EQep1Regs.QPOSMAX=5000;
   EQep1Regs.QEPCTL.bit.PCRM=1;        // PCRM=01 mode - QPOSCNT reset on
                                       // MAXPOS

   EQep1Regs.QEPCTL.bit.QPEN=1;        // QEP enable

}
```

In this specific routine, set the eQEP module in direction-count mode by setting the QSRC bit to 1. Because the signals coming in from the external logic match the signal descriptions for direction-count mode, the eQEP module is to be configured as such. The sampling resolution is then set to x1 by setting the XCR bit to 1. An x1 resolution means that only rising edges of pulses are counted while an x2 resolution counts rising and falling edges.

The position counter (QPOSCNT) should not be affected by emulation suspension, and this can be accounted for by setting the FREE_SOFT bit to 2. This also prevents emulation suspension from affecting the watchdog timer (QEDTMR), unit timer (QUTMR), and capture timer (QCTMR).

The QPOSMAX field should be set to the desired maximum count. By setting the PCRM bit, the position counter can be reset in four ways. Setting it to 0 would reset the counter on an index event. Setting it to 1 would reset the counter once it has reached the maximum value. Setting it to 2 would reset the counter on the first index event. And setting it to 3 would reset the counter on a unit time event.

Lastly, the eQEP can be enabled by setting the QPEN bit to 1. Similarly, clearing this bit would disable the module.

4 2QEP Interfacing

Another way of using CW/CCW signals with the eQEP module is to have them as inputs on two separate QEPs.

4.1 Signal Generation

Using two of the QEPs in direction-count mode with active-high signals, the clockwise signal can be connected to one of the QEP's clock input and directly force its direction signal low. Similarly, the counterclockwise signal can be connected to the other QEP's clock input and force its direction signal low as well (this is to achieve a positive count for the counterclockwise pulses).

Now, there is one QEP that is constantly counting the clockwise pulses and another QEP that is constantly counting the counterclockwise pulses. If the counterclockwise pulses are subtracted from the clockwise pulses, a net count of the pulses is determined. If given active-low signals, this result can be replicated by simply inverting both the clock inputs using the GPIO inversion functionality in the GPIO logic. If needed, the direction information could be obtained by observing which QEP's POSCNT value is being modified.

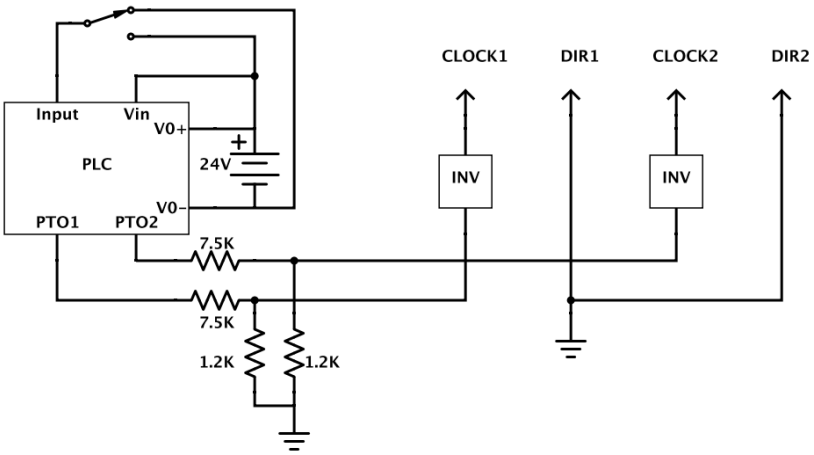

Figure 7. 2QEP Interface Diagram

Figure 7 describes the configuration for the 2QEP interface. PTO outputs are generally 24 V. Because C2000 devices are 3.3 V logic, they must be scaled down to 3.3 V. Diodes could be added for further protection. After this, they are fed directly into the eQEP module.

4.2 Software Example

To use the 2QEP interfacing method, two QEPs must be configured and initialized independently. The first QEP can be configured and initialized as described in Section 3.5. The second QEP can be configured and initialized by repeating the same procedure. The following code configures and initializes the second QEP for use with the 2QEP interfacing method. The following GPIO configuration code is applicable to F2837xD, F2837xS, and F2807x devices. Refer to the GPIO multiplexing options for your device if different GPIOs are to be used.

4.2.1 Module Configuration

```
GpioCtrlRegs.GPAPUD.bit.GPIO24 = 1;      // Disable GPIO24 pull-up (EQEP2A)
GpioCtrlRegs.GPAPUD.bit.GPIO25 = 1;      // Disable GPIO25 pull-up (EQEP2B)
GpioCtrlRegs.GPAPUD.bit.GPIO26 = 1;      // Disable GPIO26 pull-up (EQEP2S)
GpioCtrlRegs.GPAPUD.bit.GPIO27 = 1;      // Disable GPIO27 pull-up (EQEP2I)
GpioCtrlRegs.GPAQSEL2.bit.GPIO24 = 0;    // Sync GPIO24 to SYSCLK (EQEP2A)
GpioCtrlRegs.GPAQSEL2.bit.GPIO25 = 0;    // Sync GPIO25 to SYSCLK (EQEP2B)
GpioCtrlRegs.GPAQSEL2.bit.GPIO26 = 0;    // Sync GPIO26 to SYSCLK (EQEP2S)
GpioCtrlRegs.GPAQSEL2.bit.GPIO27 = 0;    // Sync GPIO27 to SYSCLK (EQEP2I)
GpioCtrlRegs.GPAGMUX2.bit.GPIO24 = 1;    // Configure GPIO24 as EQEP2A
GpioCtrlRegs.GPAGMUX2.bit.GPIO25 = 1;    // Configure GPIO25 as EQEP2B
GpioCtrlRegs.GPAGMUX2.bit.GPIO26 = 1;    // Configure GPIO26 as EQEP2S
GpioCtrlRegs.GPAGMUX2.bit.GPIO27 = 1;    // Configure GPIO27 as EQEP2I
GpioCtrlRegs.GPAMUX2.bit.GPIO24 = 1;     // Configure GPIO24 as EQEP2A
GpioCtrlRegs.GPAMUX2.bit.GPIO25 = 1;     // Configure GPIO25 as EQEP2B
GpioCtrlRegs.GPAMUX2.bit.GPIO26 = 1;     // Configure GPIO26 as EQEP2S
GpioCtrlRegs.GPAMUX2.bit.GPIO27 = 1;     // Configure GPIO27 as EQEP2I
```

4.2.2 Module Initialization

```
void InitEQep2Example(void)
{

   EQep2Regs.QDECCTL.bit.QSRC=1;            // QEP direction count mode
   EQep2Regs.QDECCTL.bit.XCR=1;             // x1 resolution

   EQep2Regs.QEPCTL.bit.FREE_SOFT=2;

   EQep2Regs.QPOSMAX=5000;
   EQep2Regs.QEPCTL.bit.PCRM=1;             // PCRM=01 mode - QPOSCNT reset on
                                            // MAXPOS

   EQep2Regs.QEPCTL.bit.QPEN=1;             // QEP enable

}
```

5 QMA Module Interfacing

Introduced on the Type-1 eQEP module, an additional Quadrature Mode Adapter (QMA) module was designed to extend the C2000 eQEP module capabilities to support direct interfacing with the CW/CCW signals output by PLC modules.

The QMA module receives the CW/CCW signals and decodes the incoming signal to the basic direction-count mode that is interpreted by the eQEP module. Essentially, the QMA module is the same as the External Logic Interfacing method described above, but with the components pulled inside of the MCU, simplifying board complexity and cost. The QMA module also incorporates error detection logic to detect illegal transitions on the EQEPA and EQEPB inputs. This interrupt is added to the standard EQEP interrupt vector and can be handled appropriately.

The QMA module is enabled by setting the QMACTRL.MODE bit as required. Setting QMACTRL.MODE to 1 will configure the eQEP to receive active Low CW/CCW Signals. Setting QMACTRL.MODE to 2 will configure the eQEP to receive active-high CW/CCW signals. Setting QMACTRL.MODE to 0 will bypass the QMA module enhancements; this is the default configuration. When the QMA module is enabled, the EQEP must be placed in direction-count mode as in the other modes described in this document. Configure the GPIOs similarly as well.

To determine if your device has the Type-1 eQEP module with the added QMA module, see the your device-specific data sheet or the *C2000 Real-Time Control Peripherals Reference Guide*.

6 Design Verification

These two new methods of interfacing CW/CCW signals were verified by using a Schneider programmable logic controller to generate clockwise and counterclockwise signals. With these signals, the external logic design was completed using TI digital logic DIPs. The results were generated in a lab under normal operating conditions. The part numbers for the design are as follows:

Part Number	Use Case
•TI TMS320F28377D	C2000 MCU
•Schneider TM238LFDC24DT	PLC with PTO Generation Capability
•TI CD4070BE (XOR Gate)	Active-High Pulse Generation
•TI CD4043BE (NOR S-R Latch)	Active-High Direction Generation
•TI CD4081BE (AND Gate)	Active-High Fault Detection, Active-Low Pulse Generation
•TI CD4044BE (NAND S-R Latch)	Active-Low Direction Generation
•TI CD4001UBE (NOR Gate)	Active-Low fault Detection

6.1 External Logic Verification

Figure 8 and Figure 9 show scope outputs of the External Logic Interfacing method for active-high and active-low signals, respectively. In both figures, signal 1 is QEPA (CW), signal 2 is QEPB (CCW), signal 3 is the generated clock signal, and signal 4 is the generated direction signal. The verification procedure did not make use of index or strobe signals taken as inputs for the eQEP module.

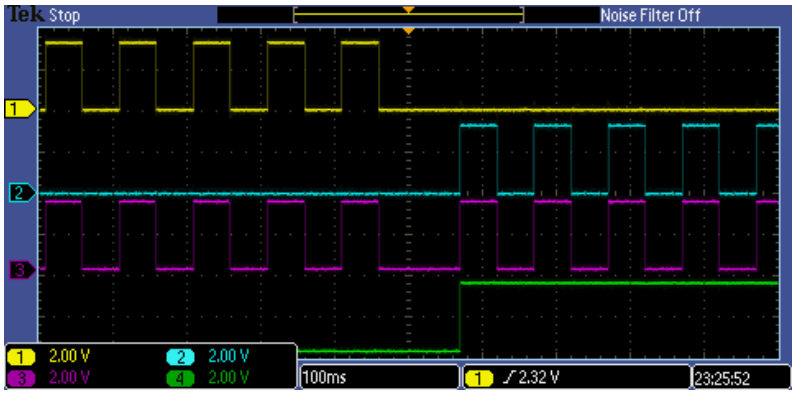

Figure 8. Active-High External Logic Interfacing Results

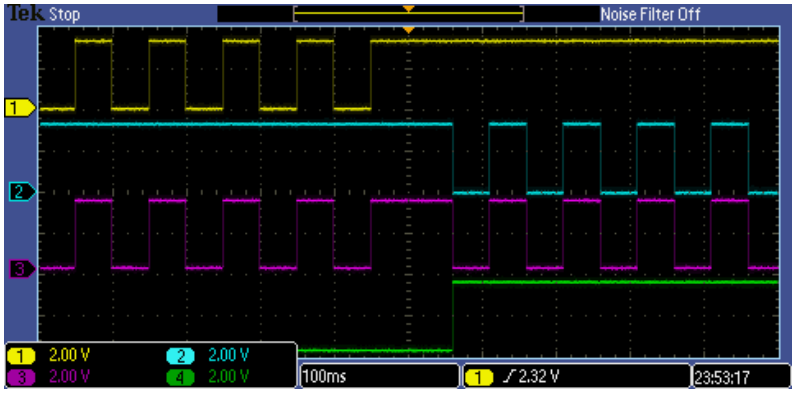

Figure 9. Active-Low External Logic Interfacing Results

6.2 2QEP Verification

Figure 10 and Figure 11 show the expressions windows for the integrated development environments (IDEs) of both the programmable logic controller and the microcontroller. The first expression window shows that the Pulse Train Output module of the PLC is instructed to output a pulse train of 4000 pulses in the positive (CW) direction. The actual command given to the PLC was to first output 4000 pulses CW, then 4000 pulses CCW, and lastly 4000 more pulses CW. The second expression window shows that the eQEP module of the MCU has counted 4000 pulses, verifying the interface. It is notable that the CW count on QEP1 is 8000 pulses and the CCW count on QEP2 is 4000 pulses, resulting in a net total of 4000 pulses.

Expression	Type	Value
M238_Controller_PTO.Application.PLC_PRG_CFC.v_xPto00_CmdMoveRelative	BOOL	FALSE
M238_Controller_PTO.Application.PLC_PRG_CFC.v_dwPto00_MRelative_Dir	PTO_DIRECTION	PTO_POSITIVE
M238_Controller_PTO.Application.PLC_PRG_CFC.v_dwPto00_MRelative_Vel	DWORD	100000
M238_Controller_PTO.Application.PLC_PRG_CFC.v_dwPto00_MRelative_Dist	DWORD	4000
M238_Controller_PTO.Application.PLC_PRG_CFC.length_dur	TIME	T#40ms

Figure 10. PLC Command for 4000 Pulses at 100 kHz

Expression	Value
EQep1Regs.QPOSCNT	8000
EQep2Regs.QPOSCNT	4000
total_count	4000
Add new expression	

Figure 11. eQEP QPOSCNT of 4000 Pulses

6.3 Frequency Test

The eQEP module was also verified to function for high frequencies with both methods of interfacing CW/CCW signals. The PLC used in the verification was capable of producing a PTO at a frequency of 100 kHZ. It was estimated that if 4000 pulses were produced at 100 kHz, all the pulses should be counted in 40 ms. This result was replicated as shown in Figure 10 and Figure 11.

7 References

- *TMS320F2837xD Dual-Core Delfino™ Microcontrollers Data Manual*
- *TMS320F28377D, TMS320F28376D, TMS320F28375D, TMS320F28374D Delfino Microcontrollers Silicon Errata*
- *TMS320F2837xD Delfino Microcontrollers Technical Reference Manual*
- *C2000 Real-Time Control Peripherals Reference Guide*

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Original (August 2014) to A Revision	Page
• Replaced 'TMS320F28377D' with 'C2000' throughout the document.	1
• Update was made in Section 1.1	2
• Update regarding added capabilities on Type-1 eQEP in Section 2	3
• Update was made in Section 3.	4
• Clarified GPIO configuration in Section 3.5.	6
• Update was made in Section 4.	7
• Clarified second eQEP configuration in Section 4.2.	7
• Clarified GPIO configuration in Section 4.2.	7
• Added new Section 5.	8

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265