Quadrature Encoder Interface (QEI)

HIGHLIGHTS

This section of the manual contains the following major topics:

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Introduction</td>
<td>1-2</td>
</tr>
<tr>
<td>2.0</td>
<td>Control and Status Register Map</td>
<td>1-5</td>
</tr>
<tr>
<td>3.0</td>
<td>Module Description</td>
<td>1-20</td>
</tr>
<tr>
<td>4.0</td>
<td>QEI Operation in Power-Saving Modes</td>
<td>1-30</td>
</tr>
<tr>
<td>5.0</td>
<td>Effects of a Reset</td>
<td>1-30</td>
</tr>
<tr>
<td>6.0</td>
<td>Design Tips</td>
<td>1-31</td>
</tr>
<tr>
<td>7.0</td>
<td>Related Application Notes</td>
<td>1-32</td>
</tr>
<tr>
<td>8.0</td>
<td>Revision History</td>
<td>1-33</td>
</tr>
</tbody>
</table>
1.0 INTRODUCTION

The Quadrature Encoder Interface (QEI) module provides the interface to incremental encoders for obtaining mechanical position data. Quadrature encoders, also known as incremental encoders or optical encoders, detect position and speed of rotating motion systems. Quadrature encoders enable closed-loop control of motor control applications, such as Switched Reluctance (SR) and AC Induction Motors (ACIM).

A typical quadrature encoder includes a slotted wheel attached to the shaft of the motor and an emitter/detector module that senses the slots in the wheel. Typically, three output channels, Phase A (QEAx), Phase B (QEBx) and Index (INDx) provide information on the movement of the motor shaft, including distance and direction.

The two channels, Phase A (QEA) and Phase B (QEB), are typically 90° out of phase with respect to each other. The Phase A and Phase B channels have a unique relationship. If Phase A leads Phase B, the direction of the motor is deemed positive or forward. If Phase A lags Phase B, the direction of the motor is deemed negative or reverse. The Index pulse occurs once per mechanical revolution and is used as a reference to indicate an absolute position. Figure 1-1 illustrates the Quadrature Encoder Interface signals.

The quadrature signals from the encoder can have four unique states (‘01’, ‘00’, ‘10’ and ‘11’) that reflect the relationship between QEA and QEB. Figure 1-1 illustrates these states for one count cycle. The order of the states get reversed when the direction of travel changes.

The quadrature decoder increments or decrements the 32-bit up/down Position Counter (POSxCNT) for each Change-of-State (COS). The counter increments when QEA leads QEB and decrements when QEB leads QEA.
Table 1-1 shows the truth table that describes how the quadrature signals are decoded.

<table>
<thead>
<tr>
<th>Current Quadrature State</th>
<th>Previous Quadrature State</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>QEA</td>
<td>QEB</td>
<td>QEA</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 1-2 illustrates the simplified block diagram of the QEI module. The QEI module consists of decoder logic to interpret the Phase A (QEA) and Phase B (QEB) signals, and an Up/Down Counter to accumulate the count. The counter pulses are generated when the quadrature state changes. The count direction information must be maintained in a register until a direction change is detected. The module also includes digital noise filters, which condition the input signal.

The QEI module consists of the following major features:

- Four input pins: two phase signals, an Index pulse and a Home pulse
- Programmable digital noise filters on inputs
- Quadrature decoder providing counter pulses and count direction
- Count direction status
- $x4$ count resolution
- Index (INDXx) pulse to reset the Position Counter
- General purpose 32-bit Timer/Counter mode
- Interrupts generated by QEI or counter events
- Up to 32-bit Velocity Counter
- 32-bit Position Counter
- 32-bit Index Pulse Counter
- 32-bit Interval Timer
- 32-bit position Initialization/Capture/Compare High Word register
- 32-bit position Initialization/Capture/Compare Low Word register
- 4X Quadrature Count mode
- External Up/Down Count mode
- External Gated Count mode
- External Gated Timer mode
- Interval Timer mode
Figure 1-2: Quadrature Encoder Interface (QEI) Module Block Diagram

Note 1: These registers map to the same memory location.
## 2.0 CONTROL AND STATUS REGISTER MAP

A summary of the registers associated with the dsPIC33/PIC24 Family Reference Manual’s Quadrature Encoder Interface (QEI) module is provided in Table 2-1.

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 15</th>
<th>Bit 14</th>
<th>Bit 13</th>
<th>Bit 12</th>
<th>Bit 11</th>
<th>Bit 10</th>
<th>Bit 9</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>All</th>
<th>Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>QEIXCON</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>QEIXIOC</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>QEIXIOCH</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>QEIXSTAT</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>POSxCNTRL</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>POSxCNTH</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>POSxHLD</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>VELxCNTRL</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>VELxCNTH</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>VELxHLD</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>INTxTMRL</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>INTxTMRH</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>INTxHLD</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>INDXxCNTRL</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>INDXxCNTH</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>INDXxHLD</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>QEIXCL</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>QEIXCH</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>QEIXECL</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>QEIXECH</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0x0000</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- `x` = unknown value on Reset; `—` = unimplemented, read as ‘0’. Reset values are shown in hexadecimal.
- **Note 1:** These registers are not available on all devices. Refer to the device-specific data sheet for availability.
### Register 2-1: QEIxCON: QEIx Control Register

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>QEIEN: Quadrature Encoder Interface Module Counter Enable bit</td>
</tr>
<tr>
<td></td>
<td>1 = Module counters are enabled</td>
</tr>
<tr>
<td></td>
<td>0 = Module counters are disabled, but SFRs can be read or written</td>
</tr>
<tr>
<td>14</td>
<td>Unimplemented: Read as ‘0’</td>
</tr>
<tr>
<td>13</td>
<td>QEISIDL: QEI Stop in Idle Mode bit</td>
</tr>
<tr>
<td></td>
<td>1 = Discontinues module operation when device enters Idle mode</td>
</tr>
<tr>
<td></td>
<td>0 = Continues module operation in Idle mode</td>
</tr>
<tr>
<td>12-10</td>
<td>PIMOD[2:0]: Position Counter Initialization Mode Select bits&lt;sup&gt;1,5&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>111 = Modulo Count mode for Position Counter and every Index input event resets the Position Counter&lt;sup&gt;4&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>110 = Modulo Count mode for Position Counter</td>
</tr>
<tr>
<td></td>
<td>101 = Resets the Position Counter when the Position Counter equals the QEIxGEC register</td>
</tr>
<tr>
<td></td>
<td>100 = Second Index event after Home event initializes Position Counter with contents of QEIxC register</td>
</tr>
<tr>
<td></td>
<td>011 = First Index event after Home event initializes Position Counter with contents of QEIxC register</td>
</tr>
<tr>
<td></td>
<td>010 = Next Index input event initializes the Position Counter with contents of QEIxC register</td>
</tr>
<tr>
<td></td>
<td>001 = Every Index input event resets the Position Counter</td>
</tr>
<tr>
<td></td>
<td>000 = Index input event does not affect Position Counter</td>
</tr>
<tr>
<td>9-8</td>
<td>IMV[1:0]: Index Match Value bits&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>11 = Index match occurs when QEB = 1 and QEA = 1</td>
</tr>
<tr>
<td></td>
<td>10 = Index match occurs when QEB = 1 and QEA = 0</td>
</tr>
<tr>
<td></td>
<td>01 = Index match occurs when QEB = 0 and QEA = 1</td>
</tr>
<tr>
<td></td>
<td>00 = Index match occurs when QEB = 0 and QEA = 0</td>
</tr>
<tr>
<td>7</td>
<td>Unimplemented: Read as ‘0’</td>
</tr>
</tbody>
</table>

#### Legend:
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- -n = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown

**Note:**
1. When CCMx = 10 or CCMx = 11, all of the QEI Counters operate as timers and the PIMOD[2:0] bits are ignored.
2. When CCMx = 00, and QEA and QEB values match the Index Match Value (IMV), the POSxCNTL and POSxCNTL registers are reset.
3. The selected clock rate should be at least twice the expected maximum quadrature count rate.
4. Not all devices support this mode. Refer to the device-specific data sheet for availability.
5. The QCAPEN and HCAPEN bits must be cleared during PIMODx Modes 2 through 7 to ensure proper functionality. Not all devices support HCAPEN. Refer to the device-specific data sheet for availability.
Register 2-1: REIxCON: QEIx Control Register (Continued)

bit 6-4  INTDIV[2:0]: Timer Input Clock Prescale Select bits(3)
(Interval Timer, Main Timer (Position Counter), Velocity Counter and Index Counter internal clock divider select)
111 = 1:256 prescale value
110 = 1:64 prescale value
101 = 1:32 prescale value
100 = 1:16 prescale value
011 = 1:8 prescale value
010 = 1:4 prescale value
001 = 1:2 prescale value
000 = 1:1 prescale value

bit 3  CNTPOL: Position and Index Counter/Timer Direction Select bit
1 = Counter direction is negative unless modified by external up/down signal
0 = Counter direction is positive unless modified by external up/down signal

bit 2  GATEN: External Count Gate Enable bit
1 = External gate signal controls Position Counter operation
0 = External gate signal does not affect Position Counter/timer operation

bit 1-0  CCM[1:0]: Counter Control Mode Selection bits
11 = Internal Timer mode
10 = External Clock Count with External Gate mode
01 = External Clock Count with External Up/Down mode
00 = Quadrature Encoder mode

Note 1: When CCMx = 10 or CCMx = 11, all of the QEI Counters operate as timers and the PIMOD[2:0] bits are ignored.
2: When CCMx = 00, and QEA and QEB values match the Index Match Value (IMV), the POSxCNTH and POSxCNTL registers are reset.
3: The selected clock rate should be at least twice the expected maximum quadrature count rate.
4: Not all devices support this mode. Refer to the device-specific data sheet for availability.
5: The QCAPEN and HCAPEN bits must be cleared during PIMODx Modes 2 through 7 to ensure proper functionality. Not all devices support HCAPEN. Refer to the device-specific data sheet for availability.
Register 2-2: QEIxIOC: QEIx I/O Control Register

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 14</th>
<th>bit 13-11</th>
<th>bit 10-9</th>
<th>bit 8</th>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5</th>
<th>bit 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>QCAPEN</td>
<td>FLTREN</td>
<td>QFDIV[2:0]</td>
<td>OUTFNC[1:0]</td>
<td>SWPAB</td>
<td>HOMPOL</td>
<td>IDXPOL</td>
<td>QEBPOL</td>
<td>QEAPOL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R-x</td>
<td>R-x</td>
<td>R-x</td>
<td>R-x</td>
<td>R-x</td>
</tr>
</tbody>
</table>

Legend:
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- -n = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown

- **QCAPEN**: Position Counter Input Capture by Index Match Event Enable bit
  - 1 = Index match event (positive edge) triggers a position capture event (HCAPEN must be cleared)
  - 0 = Index match event (positive edge) does not trigger a position capture event

- **FLTREN**: QEAx/QEBx/INDXx/HOMEx Digital Filter Enable bit
  - 1 = Input pin digital filter is enabled
  - 0 = Input pin digital filter is disabled (bypassed)

- **QFDIV[2:0]**: QEAx/QEBx/INDXx/HOMEx Digital Input Filter Clock Divide Select bits
  - 111 = 1:256 clock divide
  - 110 = 1:64 clock divide
  - 101 = 1:32 clock divide
  - 100 = 1:16 clock divide
  - 011 = 1:8 clock divide
  - 010 = 1:4 clock divide
  - 001 = 1:2 clock divide
  - 000 = 1:1 clock divide

- **OUTFNC[1:0]**: QEI Module Output Function Mode Select bits
  - 11 = The CNTCMPx pin goes high when POSSxCNT ≤ QEIxLEC or POSSxCNT ≥ QEIxGEC
  - 10 = The CNTCMPx pin goes high when POSSxCNT ≤ QEIxLEC
  - 01 = The CNTCMPx pin goes high when POSSxCNT ≥ QEIxGEC
  - 00 = Output is disabled

- **SWPAB**: Swap QEAx and QEBx Inputs bit
  - 1 = QEAx and QEBx are swapped prior to quadrature decoder logic
  - 0 = QEAx and QEBx are not swapped

- **HOMPOL**: HOMEx Input Polarity Select bit
  - 1 = Input is inverted
  - 0 = Input is not inverted

- **IDXPOL**: INDXx Input Polarity Select bit
  - 1 = Input is inverted
  - 0 = Input is not inverted

- **QEBPOL**: QEBx Input Polarity Select bit
  - 1 = Input is inverted
  - 0 = Input is not inverted

- **QEAPOL**: QEAx Input Polarity Select bit
  - 1 = Input is inverted
  - 0 = Input is not inverted
Quadrature Encoder Interface (QEI)

Register 2-2: QEIxIOC: QEIx I/O Control Register (Continued)

bit 3  HOME: Status of HOMEx Input Pin After Polarity Control bit (read-only)
1 = Pin is at logic '1', if HOMPOL bit is set to '0'
   Pin is at logic '0', if HOMPOL bit is set to '1'
0 = Pin is at logic '0', if HOMPOL bit is set to '0'
   Pin is at logic '1', if HOMPOL bit is set to '1'

bit 2  INDEX: Status of INDXx Input Pin After Polarity Control bit (read-only)
1 = Pin is at logic '1', if IDXPOL bit is set to '0'
   Pin is at logic '0', if IDXPOL bit is set to '1'
0 = Pin is at logic '0', if IDXPOL bit is set to '0'
   Pin is at logic '1', if IDXPOL bit is set to '1'

bit 1  QEB: Status of QEBx Input Pin After Polarity Control and SWPAB Pin Swapping bit (read-only)
1 = Physical pin QEBx is at logic '1', if QEBPOL bit is set to '0' and SWPAB bit is set to '0'
Physical pin QEBx is at logic '0', if QEBPOL bit is set to '1' and SWPAB bit is set to '0'
Physical pin QEAx is at logic '1', if QEBPOL bit is set to '0' and SWPAB bit is set to '1'
Physical pin QEAx is at logic '0', if QEBPOL bit is set to '1' and SWPAB bit is set to '1'
0 = Physical pin QEBx is at logic '0', if QEBPOL bit is set to '0' and SWPAB bit is set to '0'
Physical pin QEBx is at logic '1', if QEBPOL bit is set to '1' and SWPAB bit is set to '0'
Physical pin QEAx is at logic '0', if QEBPOL bit is set to '0' and SWPAB bit is set to '1'
Physical pin QEAx is at logic '1', if QEBPOL bit is set to '1' and SWPAB bit is set to '1'

bit 0  QEA: Status of QEAx Input Pin After Polarity Control and SWPAB Pin Swapping bit (read-only)
1 = Physical pin QEAx is at logic '1', if QEAPOL bit is set to '0' and SWPAB bit is set to '0'
Physical pin QEAx is at logic '0', if QEAPOL bit is set to '1' and SWPAB bit is set to '0'
Physical pin QEBx is at logic '1', if QEAPOL bit is set to '0' and SWPAB bit is set to '1'
Physical pin QEBx is at logic '0', if QEAPOL bit is set to '1' and SWPAB bit is set to '1'
0 = Physical pin QEAx is at logic '0', if QEAPOL bit is set to '0' and SWPAB bit is set to '0'
Physical pin QEAx is at logic '1', if QEAPOL bit is set to '1' and SWPAB bit is set to '0'
Physical pin QEBx is at logic '0', if QEAPOL bit is set to '0' and SWPAB bit is set to '1'
Physical pin QEBx is at logic '1', if QEAPOL bit is set to '1' and SWPAB bit is set to '1'

Register 2-3: QEIxIOCH: QEIx I/O Control High Register(1)

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

bit 15 bit 8

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HCAPEN</td>
</tr>
</tbody>
</table>

bit 7 bit 0

Legend:
R = Readable bit  W = Writable bit  U = Unimplemented bit, read as '0'
-n = Value at POR  '1' = Bit is set  '0' = Bit is cleared  x = Bit is unknown

bit 15-1  Unimplemented: Read as '0'
bit 0  HCAPEN: Position Counter Input Capture by Home Event Enable bit
1 = HOMEx input event (positive edge) triggers a position capture event (QCAPEN must be cleared)
0 = HOMEx input event (positive edge) does not trigger a position capture event

Note 1: This register is not present on all devices. Refer to the device-specific data sheet for availability.
Register 2-4: **QEIxSTAT: QEIx Status Register**

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>HS/R/C-0</th>
<th>R/W-0</th>
<th>HS/R/C-0</th>
<th>R/W-0</th>
<th>HS/R/C-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 15</td>
<td>PCHEQIRQ</td>
<td>PCHEQIEN</td>
<td>PCLEQIRQ</td>
<td>PCLEQIEN</td>
<td>POSOVIRQ</td>
<td>POSOVIEN</td>
<td></td>
</tr>
</tbody>
</table>

Legend:
- C = Clearable bit
- HS = Hardware Settable bit
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- -n = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown

- **bit 15**
  - **PCHEQIRQ**: Position Counter Greater Than Compare Status bit
    - 1 = POSxCNT > QEIxGEC
    - 0 = POSxCNT < QEIxGEC

- **bit 12**
  - **PCHEQIEN**: Position Counter Greater Than Compare Interrupt Enable bit
    - 1 = Interrupt is enabled
    - 0 = Interrupt is disabled

- **bit 11**
  - **PCLEQIRQ**: Position Counter Less Than Compare Status bit
    - 1 = POSxCNT < QEIxLEC
    - 0 = POSxCNT > QEIxLEC

- **bit 10**
  - **PCLEQIEN**: Position Counter Less Than Compare Interrupt Enable bit
    - 1 = Interrupt is enabled
    - 0 = Interrupt is disabled

- **bit 9**
  - **POSOVIRQ**: Position Counter Overflow Status bit
    - 1 = Overflow has occurred
    - 0 = No overflow has occurred

- **bit 8**
  - **POSOVIEN**: Position Counter Overflow Interrupt Enable bit
    - 1 = Interrupt is enabled
    - 0 = Interrupt is disabled

- **bit 7**
  - **PCIIRQ**: Position Counter (Homing) Initialization Process Complete Status bit
    - 1 = POSxCNT was reinitialized
    - 0 = POSxCNT was not reinitialized

- **bit 6**
  - **PCIEN**: Position Counter (Homing) Initialization Process Complete Interrupt Enable bit
    - 1 = Interrupt is enabled
    - 0 = Interrupt is disabled

- **bit 5**
  - **VELOVIRQ**: Velocity Counter Overflow Status bit
    - 1 = Overflow has occurred
    - 0 = No overflow has occurred

- **bit 4**
  - **VELOVIEN**: Velocity Counter Overflow Interrupt Enable bit
    - 1 = Interrupt is enabled
    - 0 = Interrupt is disabled

- **bit 3**
  - **HOMIRQ**: Status Flag for Home Event Status bit
    - 1 = Home event has occurred
    - 0 = No Home event has occurred

**Note 1**: This status bit is only applicable to PIMOD[2:0] modes, ‘011’ and ‘100’.
Register 2-4: QEIxSTAT: QEIx Status Register (Continued)

bit 2  **HOMIEN**: Home Input Event Interrupt Enable bit
   1 = Interrupt is enabled
   0 = Interrupt is disabled

bit 1  **IDXIRQ**: Status Flag for Index Event Status bit
   1 = Index event has occurred
   0 = No Index event has occurred

bit 0  **IDXIEN**: Index Input Event Interrupt Enable bit
   1 = Interrupt is enabled
   0 = Interrupt is disabled

**Note 1**: This status bit is only applicable to PIMOD[2:0] modes, '011' and '100'.
Register 2-5: POSxCNTL: Position x Counter Low Word Register

<table>
<thead>
<tr>
<th>Bit 15</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
</tbody>
</table>

**Legend:**
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- -n = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown

**bit 15-0 POSCNT[15:0]:** Low Word Used to Form 32-Bit Position Counter Register (POSxCNT) bits

Register 2-6: POSxCNTH: Position x Counter High Word Register

<table>
<thead>
<tr>
<th>Bit 15</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
</tbody>
</table>

**Legend:**
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- -n = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown

**bit 15-0 POSCNT[31:16]:** High Word Used to Form 32-Bit Position Counter Register (POSxCNT) bits

Register 2-7: POSxHLD: Position x Counter Hold Register

<table>
<thead>
<tr>
<th>Bit 15</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
</tbody>
</table>

**Legend:**
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- -n = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown

**bit 15-0 POSHLD[15:0]:** Hold Register for Reading and Writing Position Counter High Word Register (POSxCNTH) bits
Quadrature Encoder Interface (QEI)

Register 2-8: VELxCNTL: Velocity x Counter Low Word Register

<table>
<thead>
<tr>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

VELCNT[15:8]

bit 15 - R/W

Legend:
R = Readable bit  W = Writable bit  U = Unimplemented bit, read as ‘0’
-n = Value at POR  ‘1’ = Bit is set  ‘0’ = Bit is cleared  x = Bit is unknown

bit 0 - R/W

VELCNT[7:0]

bit 7 - R/W

Note 1: This register is not present on all devices. Refer to the device-specific data sheet for availability.

Register 2-9: VELxCNTH: Velocity x Counter High Word Register

<table>
<thead>
<tr>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

VELCNT[31:24]

bit 15 - R/W

Legend:
R = Readable bit  W = Writable bit  U = Unimplemented bit, read as ‘0’
-n = Value at POR  ‘1’ = Bit is set  ‘0’ = Bit is cleared  x = Bit is unknown

bit 0 - R/W

VELCNT[23:16]

bit 7 - R/W

Note 1: This register is not present on all devices. Refer to the device-specific data sheet for availability.

Register 2-10: VELxHLD: Velocity x Counter Hold Register

<table>
<thead>
<tr>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

VELHLD[15:8]

bit 15 - R/W

Legend:
R = Readable bit  W = Writable bit  U = Unimplemented bit, read as ‘0’
-n = Value at POR  ‘1’ = Bit is set  ‘0’ = Bit is cleared  x = Bit is unknown

bit 0 - R/W

VELHLD[7:0]

bit 7 - R/W

Note 1: This register is not present on all devices. Refer to the device-specific data sheet for availability.
Register 2-11: INTxTMRL: Interval x Timer Low Word Register

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTTMR[15:8]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>bit 8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTTMR[7:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>bit 0</td>
</tr>
</tbody>
</table>

Legend:
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as '0'
- -n = Value at POR
- ’1’ = Bit is set
- ’0’ = Bit is cleared
- x = Bit is unknown

bit 15-0 INTTMR[15:0]: Low Word Used to Form 32-Bit Interval Timer Register (INTxTMR) bits

Register 2-12: INTxTMRH: Interval x Timer High Word Register

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTTMR[31:24]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>bit 8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTTMR[23:16]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>bit 0</td>
</tr>
</tbody>
</table>

Legend:
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- -n = Value at POR
- ’1’ = Bit is set
- ’0’ = Bit is cleared
- x = Bit is unknown

bit 15-0 INTTMR[31:16]: High Word Used to Form 32-Bit Interval Timer Register (INTxTMR) bits
Quadrature Encoder Interface (QEI)

Register 2-13: INTxHLDL: Interval x Timer Hold Low Word Register

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTHLD[15:8]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 15</td>
<td>bit 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:
R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
-n = Value at POR
‘1’ = Bit is set
‘0’ = Bit is cleared
x = Bit is unknown

bit 15-0 INTHLD[15:0]: Low Word Used to Form 32-Bit Interval Timer Hold Register (INTxHLD) bits

Register 2-14: INTxHLDH: Interval x Timer Hold High Word Register

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTHLD[31:24]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 15</td>
<td>bit 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTHLD[23:16]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 7</td>
<td>bit 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:
R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
-n = Value at POR
‘1’ = Bit is set
‘0’ = Bit is cleared
x = Bit is unknown

bit 15-0 INTHLD[31:16]: High Word Used to Form 32-Bit Interval Timer Hold Register (INTxHLD) bits
Register 2-15: INDXxCNTL: Index x Counter Low Word Register

<table>
<thead>
<tr>
<th></th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:
R = Readable bit  W = Writable bit  U = Unimplemented bit, read as '0'
-n = Value at POR  '1' = Bit is set  '0' = Bit is cleared  x = Bit is unknown

bit 15-0 INDXCNT[15:0]: Low Word Used to Form 32-Bit Index x Counter Register (INDXxCNT) bits

Register 2-16: INDXxCNTH: Index x Counter High Word Register

<table>
<thead>
<tr>
<th></th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:
R = Readable bit  W = Writable bit  U = Unimplemented bit, read as '0'
-n = Value at POR  '1' = Bit is set  '0' = Bit is cleared  x = Bit is unknown

bit 15-0 INDXCNT[31:16]: High Word Used to Form 32-Bit Index x Counter Register (INDXxCNT) bits

Register 2-17: INDXxHLD: Index x Counter Hold Register

<table>
<thead>
<tr>
<th></th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:
R = Readable bit  W = Writable bit  U = Unimplemented bit, read as '0'
-n = Value at POR  '1' = Bit is set  '0' = Bit is cleared  x = Bit is unknown

bit 15-0 INDXHLD[15:0]: Hold Register for Reading/Writing Index x Counter High Word Register (INDXxCNTH) bits
### Quadrature Encoder Interface (QEI)

#### Register 2-18: QEIxCL: QEIx Initialization/Capture Low Word Register

<table>
<thead>
<tr>
<th>Bit 15</th>
<th>Bit 14</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>QEIIC[15:8]</td>
<td>bit 15</td>
<td>bit 8</td>
</tr>
</tbody>
</table>

#### Register 2-19: QEIxCH: QEIx Initialization/Capture High Word Register

<table>
<thead>
<tr>
<th>Bit 15</th>
<th>Bit 14</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>QEIIC[31:24]</td>
<td>bit 15</td>
<td>bit 8</td>
</tr>
</tbody>
</table>

### Legend:
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- -n = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- **x** = Bit is unknown

#### Bit 15-0 QEIIC[15:0]: Low Word Used to Form 32-Bit Initialization/Capture Register (QEIxIC) bits

#### Register 2-19: QEIxCH: QEIx Initialization/Capture High Word Register

<table>
<thead>
<tr>
<th>Bit 15</th>
<th>Bit 14</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>QEIIC[23:16]</td>
<td>bit 7</td>
<td>bit 0</td>
</tr>
</tbody>
</table>

### Legend:
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- -n = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- **x** = Bit is unknown

#### Bit 15-0 QEIIC[31:16]: High Word Used to Form 32-Bit Initialization/Capture Register (QEIxIC) bits
Register 2-20:  QEILxLECL: QEIX Less Than or Equal Compare Low Word Register

<table>
<thead>
<tr>
<th></th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

QEILEC[15:8]

Legend:
R = Readable bit   W = Writable bit   U = Unimplemented bit, read as '0'
-n = Value at POR   '1' = Bit is set   '0' = Bit is cleared   x = Bit is unknown

bit 15-0  QEILEC[15:0]: Low Word Used to Form 32-Bit Less Than or Equal Compare Register (QEILxLEC) bits

Register 2-21:  QEILxLECH: QEIX Less Than or Equal Compare High Word Register

<table>
<thead>
<tr>
<th></th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

QEILEC[31:24]

Legend:
R = Readable bit   W = Writable bit   U = Unimplemented bit, read as '0'
-n = Value at POR   '1' = Bit is set   '0' = Bit is cleared   x = Bit is unknown

bit 15-0  QEILEC[31:16]: High Word Used to Form 32-Bit Less Than or Equal Compare Register (QEILxLEC) bits
### Quadrature Encoder Interface (QEI)

**Register 2-22: QEIxGECL: QEIx Greater Than or Equal Compare Low Word Register**

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 14</th>
<th>bit 13</th>
<th>bit 12</th>
<th>bit 11</th>
<th>bit 10</th>
<th>bit 9</th>
<th>bit 8</th>
</tr>
</thead>
</table>

QEIGEC[15:8]

Legend:

- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown

- Por: Value at POR

- bit 15-0 **QEIGEC[15:0]:** Low Word Used to Form 32-Bit Greater Than or Equal Compare Register (QEIxGEC) bits

**Register 2-23: QEIxGECH: QEIx Greater Than or Equal Compare High Word Register**

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 14</th>
<th>bit 13</th>
<th>bit 12</th>
<th>bit 11</th>
<th>bit 10</th>
<th>bit 9</th>
<th>bit 8</th>
</tr>
</thead>
</table>

QEIGEC[31:24]

Legend:

- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown

- Por: Value at POR

- bit 15-0 **QEIGEC[31:16]:** High Word Used to Form 32-Bit Greater Than or Equal Compare Register (QEIxGEC) bits
3.0 MODULE DESCRIPTION

3.1 Position Counter

The Position Counter is 32 bits wide and is contained in two separate 16-bit registers: POSxCNTL and POSxCNTH. The counter counts the number of pulses generated by an encoder.

To read the counter during the counter operation, the user application should first read the least significant word (lsw) of the counter value from the POSxCNTL register. When the lsw is read first, the contents of POSxCNTH are automatically transferred into a hold register, POSxHLD. The user application can then read the POSxHLD register to get the most significant word (msw) of the counter value. The Position x Counter Hold register (POSxHLD) ensures that the occurrence of a carry or borrow between the read operation does not affect the reading of a coherent 32-bit value.

To write a value to the POSxCNTL:POSxCNTH register pair, the user application should first write the msw to the POSxHLD register. When the lsw of the timer value is written to the POSxCNTL register, the contents of POSxHLD are automatically transferred to the POSxCNTH register. Thus, a coherent 32-bit value can be loaded into the Position Counter in a single clock cycle.

If the POSOVIEN bit in the QEIx Status register (QEIxSTAT[8]) is set, and the Position Counter rolls over from 0x7FFFFFFF to 0x80000000, or from 0x80000000 to 0x7FFFFFFF, an interrupt will be generated.

The operating mode of the Position Counter is controlled by the CCM[1:0] bits in the QEIx Control register (QEIxCON[1:0]). The Position Counter supports the following operating modes:

- Quadrature Count Mode
- External Count with External Up/Down Mode
- External Count with External Gate Mode
- Internal Timer Mode

3.1.1 QUADRATURE COUNT MODE

In this mode, the QEA/EXTCNT and QEB/DIR/GATE inputs are decoded to generate count pulses and direction information to control the POSxCNT and VELxCNT registers. The INDXxCNT register counts when a valid edge is detected on the INDXx input. Figure 1-1 illustrates the timing diagram of the Quadrature Count mode operation.
### 3.1.2 EXTERNAL COUNT WITH EXTERNAL UP/DOWN MODE

In this mode, the QEAx/EXTCNT input is considered as an external count signal and the QEBx/DIR/GATE input provides the count direction information. The count direction is positive unless overridden by the CNTPOL bit in the QEix Control register (QEixCON[3]). Figure 3-1 illustrates the timing diagram of an External Count with External Up/Down mode operation.

**Figure 3-1: External Count with External Up/Down Mode**

<table>
<thead>
<tr>
<th>QEAx</th>
<th>QEBx</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>POSxCNT when CNTPOL = 0</td>
<td>1 76 77 78 79 78 77 76</td>
</tr>
<tr>
<td>POSxCNT when CNTPOL = 1</td>
<td>1 76 75 74 73 74 75 76</td>
</tr>
</tbody>
</table>
3.1.3 EXTERNAL COUNT WITH EXTERNAL GATE MODE

In this mode, the QEAx/EXTCNT input is considered as an external count signal. If the GATEN bit in the QEIx Control register (QEIxCON[2]) is set, and QEBx/DIR/GATE = 0, the QEBx/DIR/GATE input will inhibit the counter signal. If the GATEN bit is cleared, the gate signal does not affect the counter operation. The default count direction is positive. If the CNTPOL bit in the QEIx Control register (QEIxCON[3]) is set, the count direction is negative. Figure 3-2 illustrates the timing diagram of an External Count with External Gate mode operation.

Figure 3-2: External Count with External Gate Mode

3.1.4 INTERNAL TIMER MODE

In this mode, the velocity, index and interval center of the Position Counter uses an internal clock as the count source. The internal clock is divided by the clock divider using the INTDIV[2:0] bits in the QEIx Control register (QEIxCON[6:4]). If the GATEN bit in the QEIx Control register (QEIxCON[2]) is set and QEBx/DIR/GATE = 0, the QEBx/DIR/GATE input will inhibit the counter signal. If the GATEN bit is cleared, the gate signal does not affect the operation of the counter. The default count direction is positive. If the CNTPOL bit in the QEIx Control register (QEIxCON[3]) is set, the count direction is negative. Figure 3-3 illustrates the timing diagram of an Internal Timer mode operation.

Note: Although the POSxCNT register allows byte accesses, reading from or writing to the POSxCNT register in Byte mode gives unpredictable results. As the hold register (POSxHLD) is only 16 bits wide, the operation in Byte mode is not recommended.
Quadrature Encoder Interface (QEI)

Figure 3-3: Internal Timer Mode

- **Fcy**
- **QEBx**
- **POSxCNT when GATEN = 1 and CNTPOL = 0**
- **POSxCNT when GATEN = 1 and CNTPOL = 1**
- **POSxCNT when GATEN = 0 and CNTPOL = 0**
- **POSxCNT when GATEN = 0 and CNTPOL = 1**
- **VELxCNT when CNTPOL = 0**
- **VELxCNT when CNTPOL = 1**
- **INDXx**
- **INDXxCNT when CNTPOL = 0**
- **INDXxCNT when CNTPOL = 1**
- **HOMEx**
- **INTxTMR**
3.2 Velocity Counter

The Velocity Counter (VELxCNT) is a register that is up to 32 bits wide and increments or decrements based on the signal from the quadrature decoder logic. Refer to the device-specific data sheet for register width. Reading this register results in a counter Reset. The Index input or any of the modes specified by the PIMOD[2:0] bits in the QEIx Control register (QEIxCON[12:10]) does not affect the operation of the Velocity Counter. If the Velocity Counter rolls over from 0x7FFF to 0x8000, or from 0x8000 to 0x7FFF, and the VELOVIEN bit in the QEIx Status register (QEIxSTAT[4]) is set, an interrupt will be generated. Figure 3-4 illustrates the timing diagram of the Velocity Counter operation.

For devices that implement 32-bit Velocity registers, reads to the Velocity Counter during operation should first read the least significant word (lsw) of the counter value from the VELxCNTL register. When the lsw is read first, the contents of VELxCNTH are automatically transferred into a hold register (VELxHLD). The user application can then read the VELxHLD register to get the most significant word (msw) of the counter value. The hold register (VELxHLD) ensures that the occurrence of a carry or borrow between the read operation does not affect the reading of a coherent 32-bit value.

**Note:** The Velocity Counter specifies the distance traveled between the time interval of each sample. Reading the VELxCNT register results in a counter Reset. The user application should read the Velocity Counter at a rate of 1-4 kHz.

![Velocity Counter Diagram](image-url)
3.3 Index Counter

The Index Counter (INDXxCNT) is 32 bits wide and is contained in two separate 16-bit registers: INDXxCNTH and INDXxCNTL. It counts the Index events and is incremented or decremented based on the direction output of the quadrature logic decoder (see Figure 1-2). For more information, refer to Section 3.7 “Index Event”.

To read the Index Counter during the counter operation, the user application should first read the lsw of the counter value from the INDXxCNTL register. When the lsw is read first, the contents of the INDXxCNTH register are automatically transferred into a hold register, INDXxHLD. The user application can then read the INDXxHLD register to get the msw of the counter value.

To write a value to the INDXxCNTH:INDXxCNTL register pair, the user application should first write the msw to the INDXxHLD register. When the lsw of the Index value is written to the INDXxCNTL register, the contents of the INDXxHLD register is automatically transferred to the INDXxCNTH register. Thus, a coherent 32-bit value can be loaded into the Index Counter in a single clock cycle.

**Note:** Although, the INDXxCNT register allows byte accesses, reading from or writing to the POSxCNT register in Byte mode gives unpredictable results. As the hold register (POSxHLD) is only 16 bits wide, the operation in Byte mode is not recommended.

3.4 Interval Timer

When a motor runs at a very low speed, the encoder does not generate enough pulses for accurate speed measurement. Therefore, instead of counting the number of pulses, the pulse duration can be measured. The 32-bit Interval Timer (INTxTMR) is used to measure the time interval between each decoded quadrature count pulse when the motor operates at a very low speed. The timer counts at a rate specified by the INTDIV[2:0] bits in the QEIx Control register (QEIxCON[6:4]). The Interval Timer is cleared when the first count pulse is detected. When the next count pulse is detected, the current contents of the Interval Timer are transferred to the Interval Hold registers (INTxHLDH and INTxHLDL), the Interval Timer is cleared and then the process repeats. The lower word, INTxHLDL, has to be read first. This would lock Interval Hold registers from being updated until INTxHLDH is read. The Interval Hold registers always contain the most recent completed timing measurements. Figure 3-5 illustrates the timing diagram of the Interval Timer operation.

**Note:** If the INTxHLD register is read when a new position count pulse is detected, the contents of the INTxHLD register are not updated to avoid incoherent data reading.
3.5 Initialization/Capture Register

The 32-bit QEIx Initialization/Capture register (QEIxC) is a general purpose register that can be used to perform the following functions:

- Initialize the Position Counter
- Capture the contents of the Position Counter

The QEIxC register can perform only one of these tasks at a time, but the mode of operation can be changed during the operation. The selection is done by the PIMOD[2:0] bits of the QEIx Control register (QEIxCON[12:10]). To initialize the Position Counter mode, the contents of the QEIxC register are loaded into the POSxCNT register, based on the condition set by the PIMOD[2:0] bits.

In Capture mode, the input signal is used to capture the contents of the position register into the QEIxC register. When used for position capture, an Index match event (QCAPEN = 1) or a Home event (HCAPEN = 1) can cause the QEIxC register to take a copy of the current Position Counter contents.

3.6 Position Comparator

The 32-bit Compare registers (QEIxGEC and QEIxLEC) and associated comparator allow the user application to compare the contents of the Position Counter to a specified value. The QEIxGEC and QEIxLEC registers can be configured to define abounds of travel beyond which a Fault is generated. When used in Compare mode, the Compare register values are continually compared to the Position Counter. The comparator provides two outputs: greater than or equal, and less than or equal. When a suitable condition is met, the comparator generates and sets the PCHEQIRQ or PCLEQIRQ bit in the QEIx Status register (QEIxSTAT[13] and QEIxSTAT[11], respectively). If the interrupt enable bit, PCHEQIEN or PCLEQIEN, is set, an interrupt is generated. The comparator output is available on the CNTCMPx pin. The selection of a condition is made by the OUTFNC[1:0] bits of the QEIx I/O Control register (QEIxIOC[10:9]). The comparator can also be used to reset the Position Counter when a match is detected. The selection is made by the PIMOD[2:0] bits of the QEIx Control register (QEIxCON[12:10]). Figure 3-6 illustrates the Index Reset Position Counter operation.
### 3.7 Index Event

The IMV[1:0] bits in the QEI Control register (QEIxCON[9:8]) specify the state of the QEAx and QEBx input signals required to Acknowledge an Index event. An Index event is accepted when an Index pulse occurs while the value of the QEAx and QEBx inputs match the condition set in the IMV[1:0] bits. This prevents further Index events from being accepted until the Index input signal is deasserted, and ensures that only one Index event occurs for each Index input pulse. Figure 3-6 illustrates the Index Reset Position Counter operation.

---

**Figure 3-6: Index Reset Position Counter Operation**

- **Forward Direction (QEAx leads QEBx)**
- **Reverse Direction (QEBx leads QEAx)**

<table>
<thead>
<tr>
<th>QEAx</th>
<th>QEBx</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Count**
  - +1 +1 +1 +1 +1 +1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

- **POSxCNT(1,2)**
  - 91 95 97 94 91 95 97 94

- **Index In**
  - Assume IMV[1:0] = 00, Index Mark is Recognized; POSxCNT Register is Set to '0001'

- **QEI Interrupt Signal**
  - Index Mark Match

- **Effective Index**
  - POSxCNT Register is Set to '0001': Index Mark Match

**Note 1:** Position count update is shown when CCM[1:0] = 00.

**Note 2:** Position Counter (POSxCNTH:POSxCNTL) contents are incremented and decremented at each new count state although it is not shown in this diagram.
3.8 Position Counter Initialization Modes

By using the PIMOD[2:0] bits in the QEIx Control register (QEIxCON[12:10]), the user application can specify how the Position Counter is initialized during the module operation.

- **Mode 0** – The Position Counter is unaffected by the Index input.
- **Mode 1** – The Position Counter is cleared whenever an Index input event is detected.
- **Mode 2** – The Position Counter is initialized with the contents of the QEIxIC register on the next detected Index input event. When the Index event occurs, the PIMOD[2:0] bits are cleared and then the counter operates in Mode 0.
- **Mode 3** – The Position Counter is initialized with the contents of the QEIxIC register on the next detected Index input event following the assertion of the Home input. When an Index event occurs following the Home event, the PIMOD[2:0] bits are cleared and then the counter operates in Mode 0.
- **Mode 4** – The Position Counter is initialized with the contents of the QEIxIC register on the second detected Index input event following the assertion of the Home input. When the second Index event occurs following the Home event, the PIMOD[2:0] bits are cleared and then the counter operates in Mode 0.
- **Mode 5** – The Position Counter is cleared when the Position Counter value equals the QEIxGEC register value.
- **Mode 6** – The Position Counter is loaded with the contents of the QEIxLEC register when the Position Counter value equals the QEIxGEC register value and a count up pulse is detected. The counter is loaded with the contents of the QEIxGEC register when the Position Counter value equals the QEIxLEC register value and a count down pulse is detected.
- **Mode 7** – Modulo Count mode with Index pulse clearing. The Position Counter is loaded with the contents of the QEIxLEC register when the Position Counter equals the QEIxIC register’s contents and a count up pulse is detected. The Position Counter is loaded with the contents of the QEIxIC register when the Position Counter equals the QEIxLEC register contents and a count down pulse is detected. If an Index pulse is detected, the Position Counter is cleared whenever an Index input event is detected.

**Note:** Mode 7 is not implemented on all devices. Refer to the device-specific data sheet for availability. If not implemented, the selection is reserved.
3.9 Digital Input Filter

The QEI module uses digital noise filters to reject noise on the incoming Index and quadrature phase signals. These filters reject low-level noise and large, short duration noise spikes that typically occur in motor systems.

The filtered output signals can change only after an input level has the same value for three consecutive rising clock edges. The result is that short noise spikes between rising clock edges are ignored and pulses shorter than two clock periods are rejected.

The filter clock’s rate determines the low passband of the filter. A slower filter clock results in a passband rejecting lower frequencies.

The digital filter is enabled by setting the FLTREN bit in the QEIx I/O Control register (QEIxIOC[14]). The QFDIV[2:0] bits in the QEIx I/O Control register (QEIxIOC[13:11]) select the filter clock divider ratio for the clock signal.

Figure 3-7 illustrates the simplified block diagram of the digital noise filter.

3.10 Interrupts

The following are the sources of QEI interrupts:

- Position Counter overflow or underflow event
- Velocity Counter overflow or underflow event
- Position Counter initialization process complete
- Position Counter greater than or equal compare interrupt
- Position Counter less than or equal compare interrupt
- Index event interrupt
- Home event interrupt

The QEIx Status register (QEIxSTAT) contains the individual interrupt enable bits and the corresponding interrupt status bits for each interrupt source. A status bit indicates that an interrupt request has occurred. The module reduces all of the QEI interrupts to a single interrupt signal to the interrupt controller module.
4.0 QEI OPERATION IN POWER-SAVING MODES

4.1 Sleep Mode
When the device enters Sleep mode, QEI operations cease. The POSxCNT register stops at the current value. The QEI does not respond to active signals on the QEAx, QEBx or INDXx pins. The QEIxCON register remains unchanged.

4.2 Idle Mode
When the device enters Idle mode, the QEISIDL bit in the QEIx Control register (QEIxCON[13]) determines whether the QEI module stops in Idle mode or continues to operate in Idle mode.

If QEISIDL = 1, the QEI module enters into a power-saving mode and performs the same functions as in Sleep mode. If QEISIDL = 0, the module does not enter into a power-saving mode and continues operation in Idle mode.

4.3 Doze Mode
The QEI operation in Doze mode is similar to the normal mode.

5.0 EFFECTS OF A RESET
A Reset forces module registers into their initial Reset state.
6.0 DESIGN TIPS

**Question 1:** How do I configure the software when hardware signals are interchanged?

**Answer:** The HOMPOL, IDXPOL, QEBPOL and QEAPOL bits in the QEIxIOC register allow the user application to invert the polarity of their associated input signals. The SWPAB bit in the QEIxIOC, when set, swaps the QEAx and QEBx signals prior to their input into the quadrature decoder. Swapping of the signals reverses the direction of count. The OUTFNC[1:0] bits in the QEIxIOC register (QEIxIOC[10:9]) allow the user application to output the internal module state or the status of the Position Counter comparator on a device pin. The output timing is non-critical because the selected signal is used by the customer's application circuit, which is external to the device.

**Question 2:** How do I debug the software using the QEI module?

**Answer:** The QEIxIOC register can be used to configure the external inputs and outputs of the QEI module.

When setting up or troubleshooting a motion control-based application, the HOME, INDEX, QEB and QEA status bits in the QEIxIOC register can be used to monitor the individual state of their associated inputs.

A sensor or signal can be connected to the Home input by setting the QCAPEN bit in the QEIx I/O Control register (QEIxIOC[15]). When set, it allows the user application to capture the current Position Counter value and save it in the QEIxIC register. If the filter is enabled, the Home event is detected when a rising edge of the filtered Home signal occurs. When a Home event occurs, an interrupt generation can be enabled by setting the HOMIEN bit in the QEIx Status register (QEIxSTAT[2]).
7.0 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the dsPIC33/PIC24 device families, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Quadrature Encoder Interface (QEI) module are:

<table>
<thead>
<tr>
<th>Title</th>
<th>Application Note #</th>
</tr>
</thead>
<tbody>
<tr>
<td>Servo Control of a DC-Brush Motor</td>
<td>AN532</td>
</tr>
<tr>
<td>PIC18CXXX/PIC16CXXX DC Servomotor Application</td>
<td>AN696</td>
</tr>
<tr>
<td>Using the dsPIC30F for Vector Control of an ACIM</td>
<td>AN908</td>
</tr>
</tbody>
</table>

Note: Please visit the Microchip website (www.microchip.com) for additional application notes and code examples for the dsPIC33/PIC24 family of devices.
Quadrature Encoder Interface (QEI)

8.0 REVISION HISTORY

Revision A (June 2009)
This is the initial released version of this document.

Revision B (May 2010)
This version of the document includes the following updates:

- The document has been renamed from dsPIC33E Family Reference Manual to dsPIC33E/PIC24E Family Reference Manual
- Updated the QEIxCON: QEI Control Register (Register 2-1):
  - Renamed the register from QEIICON to QEIXCON
  - Updated the PIMOD[12:10] = 111 definition
  - Changed the bit value definitions of CCM[1:0]
- Updated the QEIxIOC: QEI I/O Control Register (Register 2-2):
  - Renamed the register from QEIIOC to QEIXIOC
  - Updated the QCAPEN[15] = 0 definition
  - Changed the bit value definitions of OUTFNC[10:9]
  - Changed the bit value definitions of SWPAB[8]
  - Updated the bit name of QFDIV[13:11], HOMPOL[7], IDXPOL[6], QEBPOL[5] and QEAPOL[4]
  - Updated the bit name and bit value definitions of HOME[3], INDEX[2], QEB[1] and QEA[0]
- Updated the QEIxSTAT: QEI Status Register (Register 2-4):
  - Renamed the register from QEISTAT to QEIXSTAT
  - Updated the bit name and bit value definitions of PCHEQIRQ[13], PCLEQIRQ[11], PCIIRQ[7], HOMIRQ[3] and IDXIRQ[1]
  - Updated the bit name of PCHEQIEN[12], PCLEQIEN[10] and IDXIEN[0]
  - Changed the bit value definitions of POSOVIRQ[9] and VELOVIRQ[5]
  - Add Note 1 to PCIIRQ[7] bit
- Updated the POSxCNTH: Position Counter High Word Register (Register 2-6):
  - Renamed the register from POSCNTH to POSxCNTH
  - Changed the bit name of POSCNTH to POSCNT and updated the definition
- Updated the POSxCNTL: Position Counter Low Word Register (Register 2-5):
  - Renamed the register from POSCNTL to POSxCNTL
  - Changed the bit name of POSCNTL to POSCNT and updated the definition
- Updated the POSxHLD: Position Counter Hold Register (Register 2-7):
  - Renamed the register from POSHLDH to POSxHLD
  - Updated the bit definition of POSHLD[15:0]
- Updated the VELxCNT: Velocity Counter Register (Register 2-8):
  - Renamed the register from VELCNT to VELxCNT
  - Updated the bit name of VELCNT
- Updated the INDXxCNTH: Index Counter High Word Register (Register 2-16):
  - Renamed the register from IDXCNTH to INDXxCNTH
  - Changed the bit name of IDXCNTH to INDXCNT and updated the bit definition
- Updated the INDXxCNTL: Index Counter Low Word Register (Register 2-15):
  - Renamed the register from IDXCNTL to INDXxCNTL
  - Changed the bit name and updated the definition of INDXCNT[15:0]
  - Changed the bit name of IDXCNTL to INDXCNT and updated the bit definition
- Updated the INDXxHLD: Index Counter Hold Register (Register 2-17):
  - Renamed the register from IDXHLDH to INDXxHLD
  - Changed the bit name of IDXHLDH to INDXxHLD and updated the bit definition
Revision B (May 2010) (Continued)

- Updated the QEIxICH: Initialization/Capture High Word Register (Register 2-19):
  - Renamed the register from ICCH to QEIxICH
  - Changed the bit name of ICCH to QEIIC and updated the bit definition
- Updated the QEIxICL: Initialization/Capture Low Word Register (Register 2-18):
  - Renamed the register from ICCL to QEIxICL
  - Changed the bit name of ICCL to QEIIC and updated the bit definition
- Added the QEIxLECH: Less Than or Equal Compare High Word Register (Register 2-21)
- Added the QEIxLECL: Less Than or Equal Compare Low Word Register (Register 2-20)
- Added the QEIxGECH: Greater Than or Equal Compare High Word Register (Register 2-23)
- Added the QEIxGECL: Greater Than or Equal Compare Low Word Register (Register 2-22)
- Updated the INTxTMRH: Interval Timer High Word Register (Register 2-12):
  - Renamed the register from INTTMRH to INTxTMRH
  - Changed the bit name of INTTMRH to INTTMR
- Updated the INTxTMRL: Interval Timer Low Word Register (Register 2-11):
  - Renamed the register from INTTMRL to INTxTMRL
  - Changed the bit name of INTTMRL to INTTMR
- Updated the INTxHLDH: Interval Timer Hold High Word Register (Register 2-14):
  - Renamed the register from INTHLDH to INTxHLDH
  - Changed the bit name of INTHLDH to INTHLD
- Updated the INTxHLDL: Interval Timer Hold Low Word Register (Register 2-13):
  - Renamed the register from INTHLDL to INTxHLDL
  - Changed the bit name of INTHLDL to INTHLD
- Deleted the CMPLH: Compare Register High Word and CMPLL: Compare Register Low
  Word registers
- All new register names and acronyms are updated throughout the document
- Updated all the timing diagrams in the document
- Updated the content in Section 3.7 “Index Event”
- Updated the bulleted list in Section 3.10 “Interrupts”
- Deleted UPDN pin in Section 4.1 “Sleep Mode”
- Minor typographical and formatting corrections were made throughout the document

Revision C (February 2018)

- Updated Section 1.0 “Introduction”:
  - Changed 16-bit Velocity Counter to up to 32-bit Velocity Counter
- Updated Section 3.2 “Velocity Counter”:
  - Changed 16-bit wide register to up to 32-bit wide register
  - Added additional information regarding 32-bit Velocity registers
- Updated Section 3.5 “Initialization/Capture Register”:
  - Added additional information for using a position capture
- Updated Section 3.6 “Position Comparator”:
  - Adds additional information for when the interrupt enable bits, PCHEQIEN or
    PCLEQIEN, are set
- Updated Section 3.8 “Position Counter Initialization Modes”:
  - Updates Mode 7 information.
- Updated Table 1-1:
  - Changed headings to QEA and QEB
- Updated Figure 1-2:
  - Changed 16-bit Velocity Counter Register to up to 32-bit Velocity Counter Register
Revision C (February 2018) (Continued)

- Updated and moved Table 2-1 to the beginning of Section 2.0 “Control and Status Register Map”.
- Updated Register 2-1, Register 2-2, Register 2-3, Register 2-4
- Added Register 2-9, Register 2-10
- Updated Family Reference Manual to the current document template.

Revision D (August 2020)

This version of the document includes the following updates:

- Updated Register 2-1 and Register 2-2.
- Updated Figure 3-3, Figure 3-5 and Figure 3-6
- Updated Section 3.4 “Interval Timer” and Section 3.8 “Position Counter Initialization Modes”.
Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip’s intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable.” Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication is provided for the sole purpose of designing with and using Microchip products. Information regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks
The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBox, KeeLoq, Kleer, LANCheck, LinkMD, maxSTylus, maxTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BAL, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntellIMOS, Libero, motorBench, mTouch, Powermate 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.


SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.
The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries. All other trademarks mentioned herein are property of their respective companies.

© 2009-2020, Microchip Technology Incorporated, All Rights Reserved.

For information regarding Microchip’s Quality Management Systems, please visit www.microchip.com/quality.

ISBN: 978-1-5224-6637-6
<table>
<thead>
<tr>
<th>AMERICAS</th>
<th>ASIA/PACIFIC</th>
<th>ASIA/PACIFIC</th>
<th>EUROPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corporate Office</td>
<td>Australia - Sydney</td>
<td>India - Bangalore</td>
<td>Austria - Wels</td>
</tr>
<tr>
<td>2355 West Chandler Blvd. Chandler, AZ 85224-6199</td>
<td>Tel: 61-2-9888-6733</td>
<td>Tel: 91-80-3090-4444</td>
<td>Tel: 43-7242-2244-39</td>
</tr>
<tr>
<td>Tel: 480-792-7200</td>
<td>China - Beijing</td>
<td>India - New Delhi</td>
<td>Fax: 43-7242-2244-393</td>
</tr>
<tr>
<td>480-792-7277</td>
<td>China - Chengdu</td>
<td>India - Pune</td>
<td>Denmark - Copenhagen</td>
</tr>
<tr>
<td>Technical Support: <a href="http://www.microchip.com/support">http://www.microchip.com/support</a></td>
<td>China - Chongqing</td>
<td>Tel: 91-20-4121-0141</td>
<td>Tel: 45-4485-5910</td>
</tr>
<tr>
<td>Web Address: <a href="http://www.microchip.com">www.microchip.com</a></td>
<td>China - Dongguan</td>
<td>Japan - Osaka</td>
<td>Fax: 45-4485-2829</td>
</tr>
<tr>
<td>Atlanta</td>
<td>China - Guangzhou</td>
<td>Japan - Tokyo</td>
<td>Finland - Espoo</td>
</tr>
<tr>
<td>Duluth, GA</td>
<td>China - Hangzhou</td>
<td>Japan - Tokyo</td>
<td>Tel: 358-9-4520-820</td>
</tr>
<tr>
<td>Tel: 678-957-9614</td>
<td>China - Hong Kong SAR</td>
<td>Korea - Daegu</td>
<td>France - Paris</td>
</tr>
<tr>
<td>Fax: 678-957-1455</td>
<td>Tel: 852-2943-5100</td>
<td>Korea - Seoul</td>
<td>Tel: 33-1-69-53-63-20</td>
</tr>
<tr>
<td>Austin, TX</td>
<td>China - Nanjing</td>
<td>Malaysia - Kuala Lumpur</td>
<td>Fax: 33-1-69-30-90-79</td>
</tr>
<tr>
<td>Tel: 512-257-3370</td>
<td>China - Qingdao</td>
<td>Tel: 60-3-7651-7906</td>
<td>Germany - Garching</td>
</tr>
<tr>
<td>Boston</td>
<td>China - Shenyang</td>
<td>Malaysia - Penang</td>
<td>Tel: 49-8931-9700</td>
</tr>
<tr>
<td>Westborough, MA</td>
<td>China - Shenzhen</td>
<td>Tel: 60-4-227-8870</td>
<td>Germany - Haan</td>
</tr>
<tr>
<td>Tel: 774-760-0087</td>
<td>China - Suzhou</td>
<td>Philippines - Manila</td>
<td>Tel: 49-2129-3766400</td>
</tr>
<tr>
<td>Fax: 774-760-0088</td>
<td>China - Wuhan</td>
<td>Tel: 63-2-634-9065</td>
<td>Germany - Heilbronn</td>
</tr>
<tr>
<td>Chicago</td>
<td>China - Xian</td>
<td>Singapore</td>
<td>Tel: 49-7131-72400</td>
</tr>
<tr>
<td>Itasca, IL</td>
<td>China - Xiamen</td>
<td>Tel: 66-334-8870</td>
<td>Germany - Karlsruhe</td>
</tr>
<tr>
<td>Tel: 630-285-0071</td>
<td>China - Zhuhai</td>
<td>Taiwan - Hsin Chu</td>
<td>Tel: 49-721-625370</td>
</tr>
<tr>
<td>Fax: 630-285-0075</td>
<td>Tel: 86-756-3210040</td>
<td>Tel: 886-3-577-8366</td>
<td>Germany - Munich</td>
</tr>
<tr>
<td>Dallas</td>
<td>Tel: 281-894-5983</td>
<td>Taiwan - Kaohsiung</td>
<td>Tel: 49-89-627-144-0</td>
</tr>
<tr>
<td>Addison, TX</td>
<td>Indianapolis</td>
<td>Taiwan - Taipei</td>
<td>Fax: 49-89-627-144-44</td>
</tr>
<tr>
<td>Tel: 972-818-7243</td>
<td>Noblesville, IN</td>
<td>Tel: 886-2-258-8600</td>
<td>Germany - Rosenheim</td>
</tr>
<tr>
<td>Fax: 972-818-2924</td>
<td>Tel: 317-773-8323</td>
<td>Thailand - Bangkok</td>
<td>Tel: 49-8031-354-560</td>
</tr>
<tr>
<td>Detroit</td>
<td>Tel: 317-773-5453</td>
<td>Tel: 66-2-694-1351</td>
<td>Israel - Ra'anana</td>
</tr>
<tr>
<td>Novi, MI</td>
<td>Tel: 317-536-2380</td>
<td>Vietnam - Ho Chi Minh</td>
<td>Tel: 972-9-744-7705</td>
</tr>
<tr>
<td>Tel: 248-848-4000</td>
<td>Los Angeles</td>
<td>Tel: 84-28-5448-2100</td>
<td>Italy - Milan</td>
</tr>
<tr>
<td>Mission Viejo, CA</td>
<td>Tel: 49-462-9523</td>
<td>Tel: 39-0331-742611</td>
<td>Tel: 39-0331-466781</td>
</tr>
<tr>
<td>Tel: 949-462-9608</td>
<td>Fax: 949-462-9608</td>
<td>Fax: 39-0331-466781</td>
<td></td>
</tr>
<tr>
<td>Tel: 951-273-7800</td>
<td>Raleigh, NC</td>
<td>Raleig</td>
<td>Italy - Padova</td>
</tr>
<tr>
<td>Tel: 919-844-7510</td>
<td>New York, NY</td>
<td>Tel: 39-049-7625286</td>
<td>Tel: 39-049-7625286</td>
</tr>
<tr>
<td>Tel: 631-435-6000</td>
<td>San Jose, CA</td>
<td>Netherlands - Drunen</td>
<td>Tel: 31-416-690399</td>
</tr>
<tr>
<td>Tel: 408-735-9110</td>
<td>Canada - Toronto</td>
<td>Fax: 31-416-690340</td>
<td>Tel: 31-416-690340</td>
</tr>
<tr>
<td>Tel: 408-436-4270</td>
<td>Tel: 905-695-1980</td>
<td>Norway - Trondheim</td>
<td>Tel: 47-7288-4388</td>
</tr>
<tr>
<td>Tel: 905-695-2078</td>
<td>Tel: 905-695-1980</td>
<td>Poland - Warsaw</td>
<td>Tel: 48-22-3325737</td>
</tr>
<tr>
<td>Tel: 905-695-2078</td>
<td>Tel: 905-695-1980</td>
<td>Romania - Bucharest</td>
<td>Tel: 40-21-407-87-50</td>
</tr>
<tr>
<td>Tel: 905-695-2078</td>
<td>Tel: 905-695-1980</td>
<td>Spain - Madrid</td>
<td>Tel: 34-91-708-08-90</td>
</tr>
<tr>
<td>Tel: 905-695-2078</td>
<td>Tel: 905-695-1980</td>
<td>Sweden - Gothenberg</td>
<td>Fax: 34-91-708-08-91</td>
</tr>
<tr>
<td>Tel: 905-695-2078</td>
<td>Tel: 905-695-1980</td>
<td>Sweden - Stockholm</td>
<td>Tel: 46-31-704-60-40</td>
</tr>
<tr>
<td>Tel: 905-695-2078</td>
<td>Tel: 905-695-1980</td>
<td>UK - Wokingham</td>
<td>Tel: 44-118-921-5800</td>
</tr>
<tr>
<td>Tel: 905-695-2078</td>
<td>Tel: 905-695-1980</td>
<td>Fax: 44-118-921-5820</td>
<td></td>
</tr>
</tbody>
</table>