

AVR1600: Using the XMEGA Quadrature Decoder

Features

- Quadrature Decoders
- 16-bit angular resolution
- Rotation speed and acceleration

1 Introduction

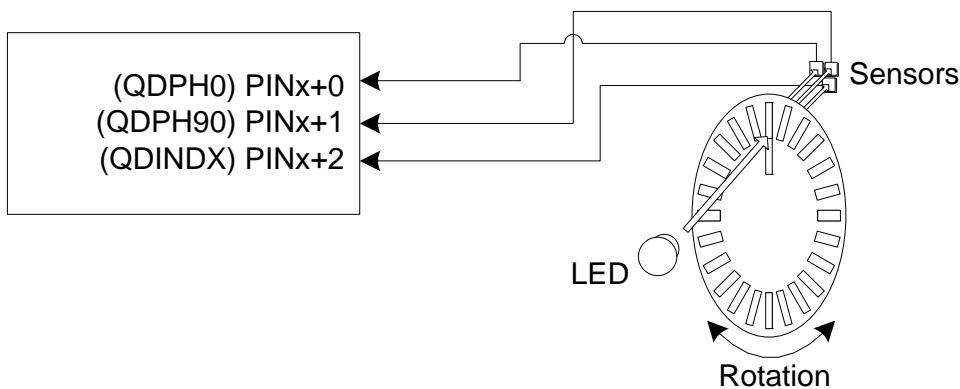
Quadrature encoders are used to determine the position and speed of a rotary device, such as servo-motors, volume control wheels, PC mice etc. The decoded quadrature signals are used as a sensory input to a system to determine the absolute or relative position of the rotary device, which again can be used e.g. in a control loop (for e.g. the servo-motor).

The AVR® XMEGA™ includes hardware support for reading position from Quadrature Encoders. A combination of peripheral modules is used to decode the Quadrature Encoder signals: IO pins are used as input to the Event System's Quadrature Decoder (QDEC), which connects to a Timer/Counter.

The XMEGA hardware supports incremental encoders (Quadrature Encoders). The Quadrature Encoder also supports encoders with index signal for absolute positioning.

This application note describes the basic functionality of the XMEGA QDECs with code example.

Figure 1-1. Example system.



Rev. 8109A-AVR-08/08



2 Quadrature Encoders

A quadrature encoder uses two signals to encode rotation and direction. The two quadrature encoder signal ($QDPH0$ and $QDPH90$) are characterized by having two square waves phase shifted 90 degrees relative to each other. This can be implemented using a quadrature encoder disk shown in Figure 2-1 or with a rate encoder disk (shown in Figure 2-2) with the sensors logical 90 degrees out of phase.

Figure 2-1. Quadrature Encoder disk.

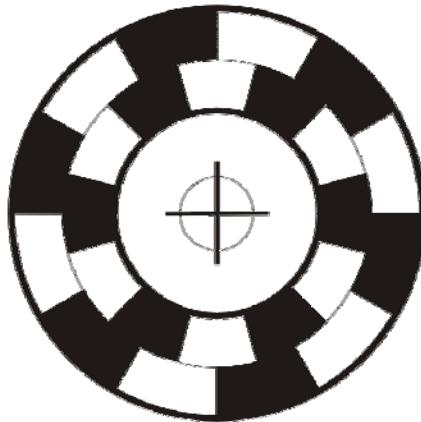
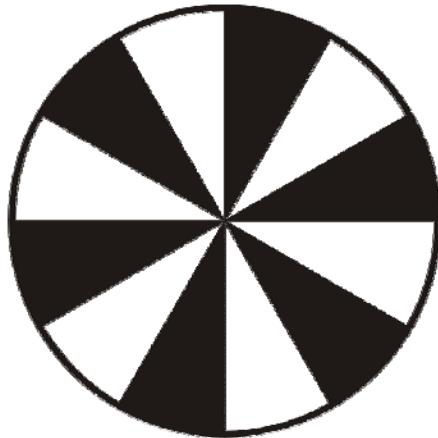


Figure 2-2. 30 Degree Rate Encoder disk.



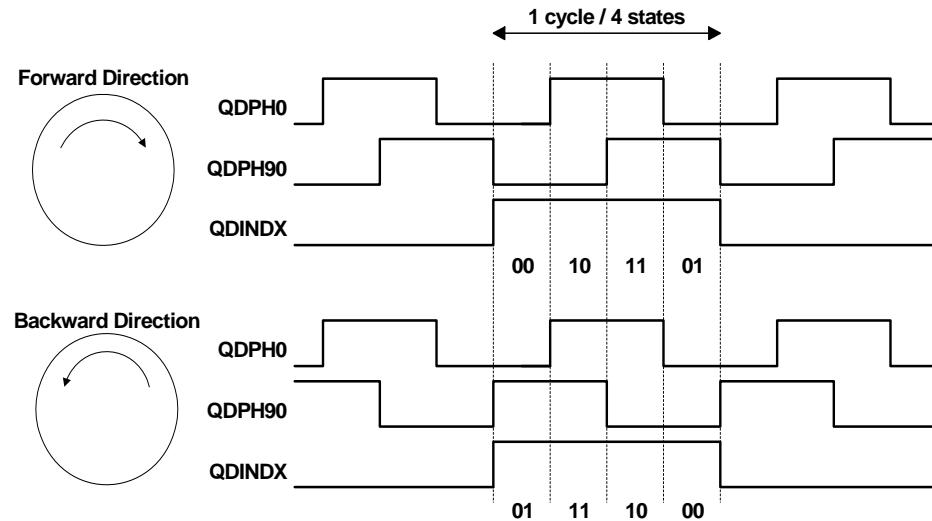
Rotational movement can be measured by counting the edges of the two waveforms. The phase relationship between the two square waves determines the direction of rotation.

Figure 2-3 shows typical quadrature signals from a rotary encoder. The signals named $QDPH0$ and $QDPH90$ are the two quadrature signals. The figure shows how the phase relationship determines the direction of rotation. When $QDPH0$ leads $QDPH90$, the rotation is defined as positive or forward. When $QDPH90$ leads $QDPH0$, the rotation is defined as negative or reverse. The concatenation of the two phase signals is called the quadrature state or the phase state.

The index signal shown in the figure as QDINDX, used for absolute positioning, can be high for a maximum of four states. If the index is high for four states as shown in Figure 2-3, any of the states can be chosen to be the index recognition state.

Quadrature encoders are commonly used as position sensors in motor applications, but are also found in other rotary sensors, such as the ball tracker in computer mice.

Figure 2-3. Quadrature signals from a rotary encoder.



2.1 Quadrature encoder output signals

Quadrature encoders have two or three output lines: Two-output encoders can provide information about the relative position for a rotary device. These two outputs have four (quad) states – from which it has its name. Unless the initial rotary displacement is known, a two-output encoder can only be used to calculate relative movement, speed and position. The absolute rotary displacement will not be known. Having a third signal, referred to as an index signal, generating a pulse once per revolution, can resolve this.

3 Quadrature decoding

The event system has extensions that make it possible to decode a quadrature signal and use this as a source for a Timer/Counter.

The rotary displacement using a two-output encoder is shown in Figure 3-2. Using a three-output encoder the absolute position will be known, as shown in Figure 3-1.

When the QDINDX signal occurs the Timer/Counter value will be reset if not equal to BOTTOM, and an error bit will be set (ERRIF in INTFLAGS-interrupt flag register to the Timer/Counter). This enables the system to detect skip/error in the system or reset the counter for first pass.

The speed and acceleration can be calculated by timing the rate of change in the Timer/Counter register (shown in Figure 3-3).



Figure 3-1. Timer/Counter value with index/reset signal.

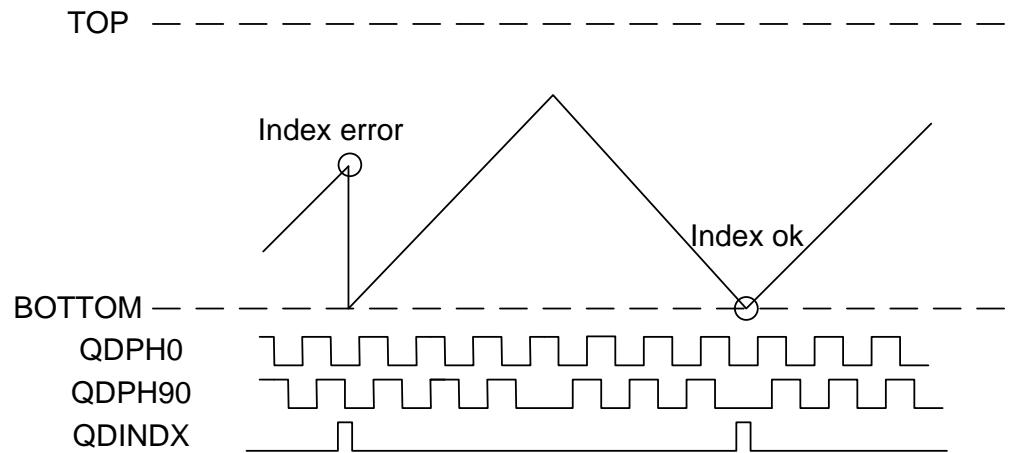


Figure 3-2. Timer/Counter value without index/reset signal.

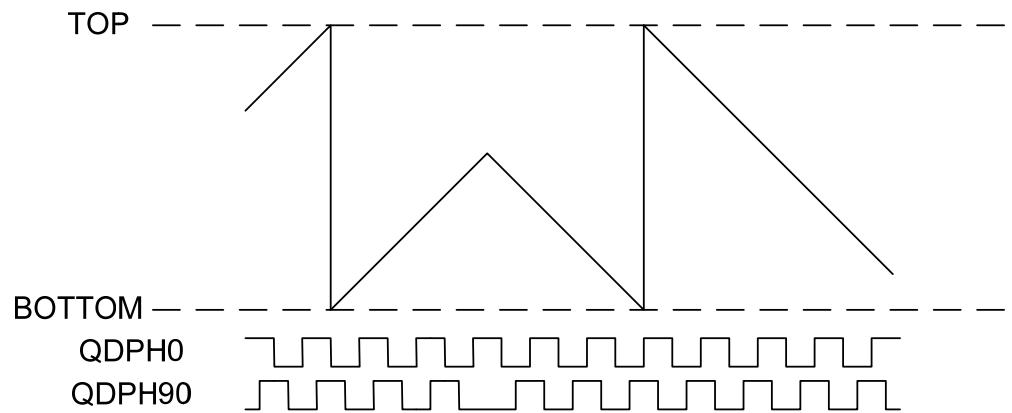
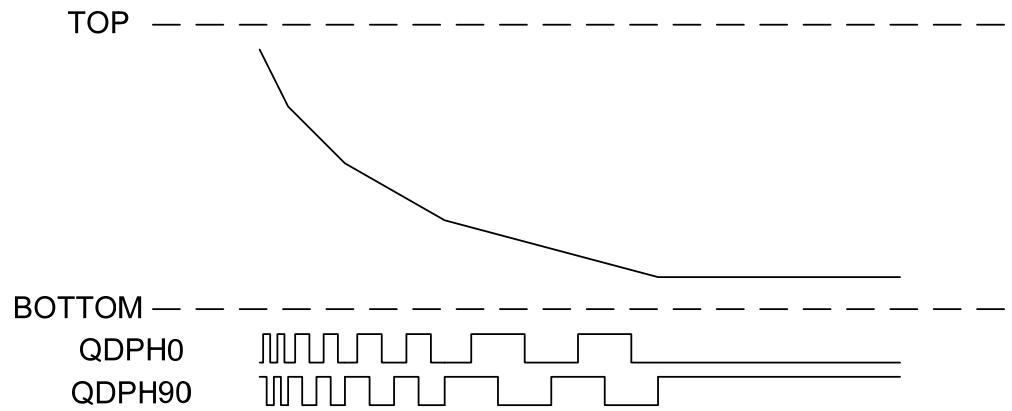


Figure 3-3. Timer/Counter value with decreasing speed (backwards rotation)



3.1 The XMEGA quadrature decoder

The XMEGA quadrature decoder supports automatic decoding of a quadrature signal, with optional reset by an index signal. To utilize the quadrature decoder, three modules are used:

- I/O port pins – quadrature signal input
- The Event System – quadrature decoding
- A Timer/Counter – keeping track of the current position

The XMEGA Timer/Counter modules are able to use the quadrature events to count up/down when the event action is set to quadrature encoding. As a result, the current position of the rotary device is tracked by the Timer/Counter, and can be directly read from the Timer/Counter CNT[H:L] register. The Timer/Counter works normally and interrupts/events can be used. E.g. an event can be sent to another Timer/Counter counting revolution. Interrupts can be given at relative/absolute positions. An extra Timer/Counter can be used to calculate rotation speed and acceleration.

3.1.1 Index signal and error states

The index signal can be used to reset the Timer/Counter: When the index signal is high and the quadrature state match that selected by the QDIRM bits (shown in Table 3-1) the Timer/Counter is reset. The QDIRM[1:0] bits are located in the event channel control register - CHnCTRL.

When using an encoder with index output, the Timer/Counter period (PER[H:L]) register should be set to match the number of pulses per revolution from the encoder wheel: When the period register corresponds to the number of quadrature pulses per revolution, the index should occur and the position counter should be equal to BOTTOM. If the position counter is different from BOTTOM when the index is recognized, the Timer/Counter ERRIF bit is set in the INTFLAGS – Interrupt Flag Register. Similarly the ERRIF is set if the position counter passes BOTTOM without recognition of the index.

The index signal can be implemented differently from different manufacturers. The index can be high one state and the corresponding state must be chosen in the QDIRM bit settings for the index recognition state (shown in Table 3-1). The index can be high for a maximum of 4 states and then any one of the index recognition states can be chosen.

A detailed example of how to use the XMEGA quadrature decoder can be found in section 3.2.

Table 3-1. QDIRM Bit Settings.

QDIRM[1:0]		Index Recognition State
0	0	{QDPH0, QDPH90} = 0b00
0	1	{QDPH0, QDPH90} = 0b01
1	0	{QDPH0, QDPH90} = 0b10
1	1	{QDPH0, QDPH90} = 0b11



3.2 Quadrature Decoding

In this example, TCC0 will be used as a quadrature counter, using Event Channel 0 for quadrature decoding and Event Channel 1 for the index signal. PORTD will be used for input of the three quadrature signals: QDPH0 on PD0, QDPH90 on PD1 and QDIND on PD2.

1. Configure PD0 and PD1 as inputs.
2. Configure PD0 and PD1 input sense control register to level sensing (transparent for events).
3. Select PD0 as multiplexer input for event channel 0.
4. Optional for index:
 - a. Configure PD2 as input.
 - b. Configure PD2 input sense control register to sense both edges.
 - c. Select PD2 as multiplexer input for event channel 1.
 - d. Set the Quadrature Index Enable bit in event channel 0.
 - e. Select the Index Recognition mode for channel 0.
5. Enable quadrature decoding and digital filtering in event channel 0.
6. Set Quadrature decoding as event action for TCC0.
7. Select event channel 0 as event source for TCC0.
8. Set the period register of TCC0 to $(n * 4 - 1)$, where n is the line count of the quadrature encoder.
9. Enable TCC0 by setting CLKSEL to a CLKSEL_DIV1.

The angle of a quadrature encoder attached to QDPH0, QDPH90 (and QINDX) can now be read directly from the Timer/Counter Count register. If the Count register is different from BOTTOM when the index is recognized, the Timer/Counter error flag is set. Similarly the error flag is set if the position counter passes BOTTOM without the recognition of the index.

A code example using the Quadrature decoder is included in the source code for this application note. This implementation uses the index signal for absolute position.

4 Driver/Example Implementation

The included driver has functions that can be used to configure the Quadrature Decoder. The driver is written in ANSI[®] C, and should compile on all compilers with XMEGA support.

Note that this driver is **not** written with high performance in mind. It is designed as a library to get started with the XMEGA Quadrature Decoder and an easy-to-use framework for rapid prototyping. For time and code space critical application development, consider replacing function calls with macros or direct access to registers.

4.1 Files

The source code package consists of the following files:

- qdec_driver.c – Quadrature decoder driver source file
- qdec_driver.h – Quadrature decoder driver header file
- qdec_example.c – Example of using the quadrature decoder

4.2 Doxygen Documentation

All source code is prepared for automatic documentation generation using Doxygen. Doxygen is a tool for generating documentation from source code by analyzing the source code and using special keywords. For more details about Doxygen please visit <http://www.doxygen.org>. Precompiled Doxygen documentation is also supplied with the source code accompanying this application note, available from the *readme.html* file in the source code folder.





Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Request
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN ATTEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATTEL'S WEB SITE, ATTEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATTEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2008 Atmel Corporation. All rights reserved. Atmel®, logo and combinations thereof, AVR® and others, are the registered trademarks or trademarks, XMEGA™ and others are trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.