AVR1311: Using the XMEGA Timer/Counter Extensions

Features

- **Advanced Waveform eXtensions (AWeX)**
  - **Dead-time insertion**
  - **Pattern generation**
  - **Fault protection**
- **High Resolution Extension (HiRes)**
  - **Increases resolution by 2 bits (4x)**

1 Introduction

Some Timer/Counters on the XMEGA™ have extension modules that are useful for applications such as motor and power control applications. This document gives an introduction to the extension modules available and how to use them.

8-bit AVR® Microcontrollers

Application Note

Rev. 8076A-AVR-04/08

2 AWeX

The Advanced Waveform eXtension (AWeX) is a collection of Timer/Counter extensions that are typically used in motor and power control applications. When the AWeX extension is used, the PWM outputs of the Timer/Counter module is routed through the AWeX module, which controls the port pins.

Figure 2-1. AWeX module overview.

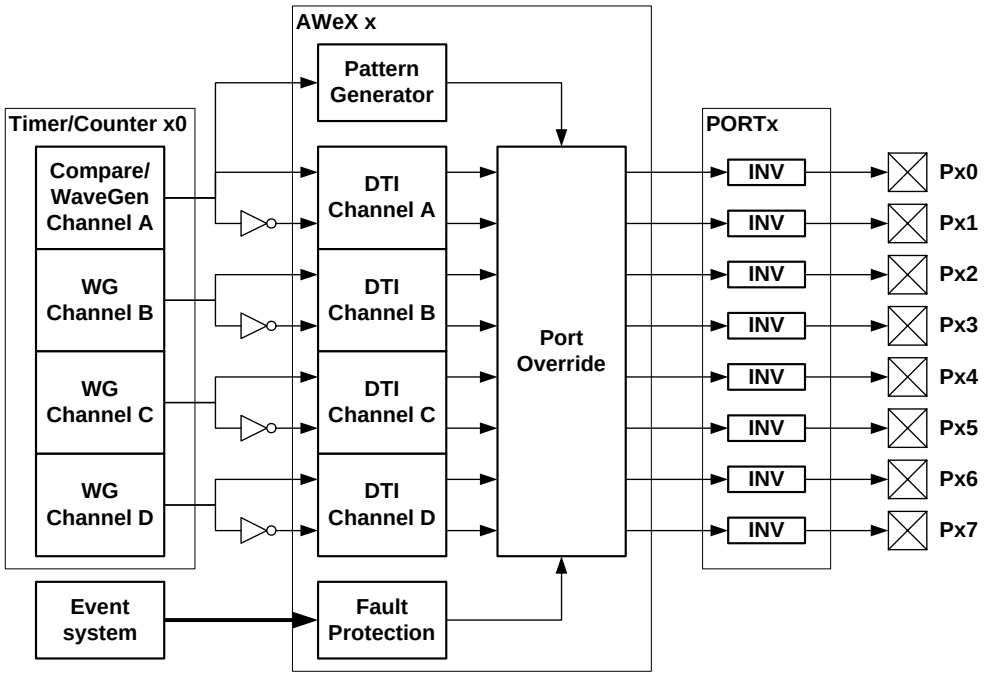


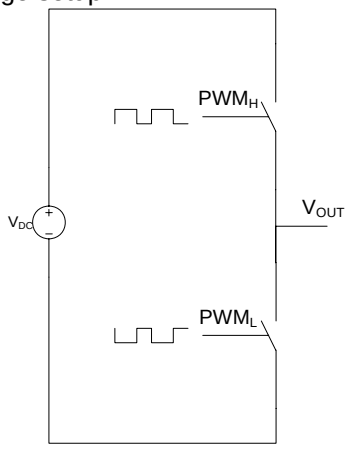
Figure 2-1 shows an overview of the AWeX module with its three sub-functions: Dead-time Insertion (DTI), Pattern generation and Fault protection.

The AWeX module is available on select Timer/Counter modules. Refer to the device data sheet for more information.

2.1 Dead-time Insertion

In many applications, such as motor control, PWM is used to generate waveforms using a half-bridge configuration similar to the one shown in Figure 2-2. If the high- and low-side switches are fed with inverted PWM waveforms (PWM_H and PWM_L), the average output voltage, V_{OUT}, will be proportional to the duty cycle of the PWM_H signal.

Figure 2-2. Typical half-bridge setup.



A half-bridge, like the one in Figure 2-2, is typically realized using MOSFETs or IGBTs. These devices are not capable of turning on/off instantaneously. There is always a small rise/fall time on the output. If the signal applied to the low side switch is just an inverted version of the signal applied to the high side, there will be a small period during the switching where both the high- and low-side switches are conducting, leading to a short-circuit between positive supply and ground for a short period. This is usually known as shoot-through, and should obviously be avoided.

The usual solution to avoid shoot-through is to insert a small dead-time around the switching instant. When the low side is switched off, the high-side is not switched on until after the dead-time has passed. This is called dead-time insertion.

The Dead-time insertion extension handles dead-time insertion automatically, ensuring that shoot-through cannot happen as a result of a software glitch. The dead-time for high-side and low-side can be set individually through the DTHS and DTLS registers respectively. As a shortcut, DTHS and DTLS can be set to the same value by writing to the DTBOTH register. The dead-time value is given in main system clock cycles. The allowable range for the dead-time is thus 0-255 main system clock cycles.

2.1.1 Pin Mapping

The DTI extension overrides the I/O port directly and has higher priority than the Timer/Counter module. The pin mapping is shown in Table 2-1. It is possible to enable dead-time insertion on each channel separately. Only the output pin pairs connected to signals that use DTI extension will be overridden by the AWeX module.

Table 2-1. Dead-time insertion pin mapping.

Pin	Signal
Pin 0	CCAH
Pin 1	CCAL
Pin 2	CCBH
Pin 3	CCBL
Pin 4	CCCH
Pin 5	CCCL
Pin 6	CCDH
Pin 7	CCDL

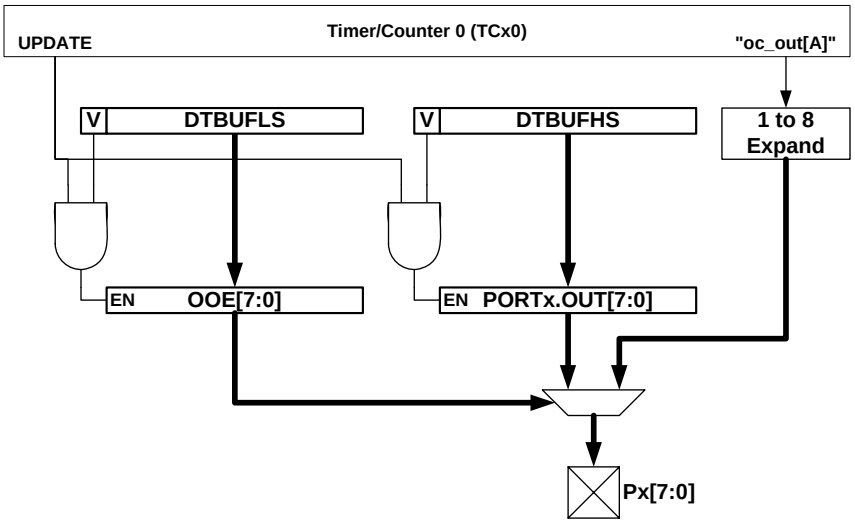
2.2 Pattern Generation

The Pattern Generation extension reuses the DTI registers and double buffer mechanism to produce a synchronized bit pattern on the port it is connected to. In addition, compare channel A from Timer/Counter 0 connected to the same I/O port can be distributed to, and override, all the port pins. This allows complex patterns with pulse-width modulation to be generated on the I/O port.

The pattern generator can be used to generate the commutation sequence for BLDC and stepper motors, controlling arrays of LEDs, or for any application that need to selectively distribute a PWM signal to several destinations.

The functionality of the Pattern Generator is illustrated in Figure 2-3. The DTBUFLS register is used as buffer register for the Output Override Enable (OOE) register in the AWeX module, while DTBUFHS is used as buffer register for the OUT register in the associated I/O port module. The latching of values from buffer registers to their destination is synchronized to the UPDATE condition in the Timer/Counter module. For more information about the UPDATE condition, please consult the Timer/Counter data sheet or the application note AVR1306.

Figure 2-3. Pattern generation overview.



2.3 Fault Protection

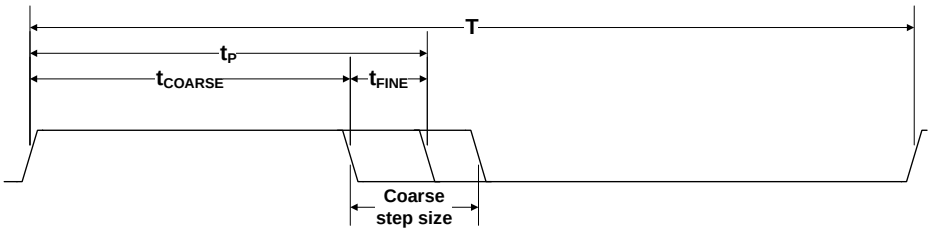
The Fault Protection feature enables fast and deterministic action when a fault is detected. The fault protection is event controlled, thus any event from the Event System can be used to trigger a fault action.

When the Fault Protection is enabled an incoming event from any of the selected event channel can trigger the event action. Each event channel can be separately enabled as fault protection input, and the specified event channels will be ORed together allowing multiple event sources to be used for fault protection at the same time. For more information on the Fault Protection, read the Xmega manual.

3 HiRes

The HiRes extension module increases the PWM output resolution by a factor of four. This is accomplished by combining the PWM-generation abilities of a standard Timer/Counter module with a high-resolution part running at four times higher frequency. The concept is illustrated in Figure 3-1. The Pulse width is divided in two parts, t_{COARSE} and t_{FINE}. The sum of these, t_P, is the total pulse width. The Timer/Counter unit generates the coarse pulse, while the HiRes extension extends the coarse pulse by 0-3 fine clock cycles.

Figure 3-1. HiRes concept.



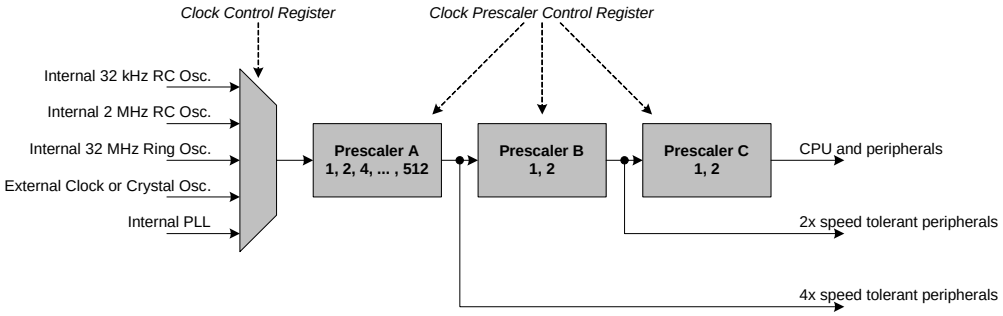
This is achieved by dividing the PWM generation in two parts. The Timer/Counter unit is responsible for generating a PWM signal based on the 14 most significant bits (MSBs) of the Timer/Counter compare channels. This coarse PWM signal is fed to the HiRes extension, which is responsible for adding the fine pulse based on the two least significant bits (LSBs) of the Timer/Counter compare channels.

The HiRes extension is controlled by two bits in the HIRESx.CTRL register, HREN0 and HREN1. These bits enable High Resolution operation for Timer/Counter x0 and Timer/Counter x1 respectively.

3.1 Configuring the System Clocks for HiRes Operation

In order for the HiRes module to work as intended, the HiRes module needs a clock input with four times higher frequency than the main system clock. Figure 3-2 shows a simplified illustration of the clock distribution system. As the illustration shows, enabling Prescaler B and Prescaler C divides the main system clock frequency by 4. The HiRes module is clocked by the output of Prescaler A, four times the CPU clock frequency. The clock system is explained more in detail in application note “AVR1003: Using the XMEGA clock system”.

Figure 3-2. Clock distribution



3.2 Using the HiRes module

As soon as the HiRes module is enabled and the clock system is configured correctly, the Timer/Counter is ready for high-resolution operation. In HiRes mode, the Timer/Counter behaves almost as if it was running at four times the frequency off the CPU. This is not the case, however, and the differences are subtle, but important:

- The Timer/Counter itself is not running at 4 times the system clock frequency, but it will count by 4 for each system clock cycle. In other words, the 2 LSBs are always 0.
- The Period/TOP value of the Timer/Counter cannot be set to a “high resolution” value. The 2 LSBs must be 0.
- A positive or negative output pulse can never be shorter than one coarse step. A compare value of 0 produces a constant low output. A compare value equal to PER produces a constant high output.

4 Driver Implementation

The included driver has functions that control all the major features of the timer/counter modules (including waveform generation). All functions take a pointer to a timer/counter module as its first argument, so the same functions can be reused for all timer/counter modules on one XMEGA.

Note that this driver is **not** written with high performance in mind. It is designed as a library to get started with the Xmega timer/counters 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 driver package consists of the following files:

- awex_driver.c – AWeX driver source file
- awex_driver.h – AWeX driver header file
- hires_driver.c – HiRes driver source file
- hires_driver.h – HiRes driver header file
- tc_extensions_example.c – Examples using the Timer/Counter extensions.

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