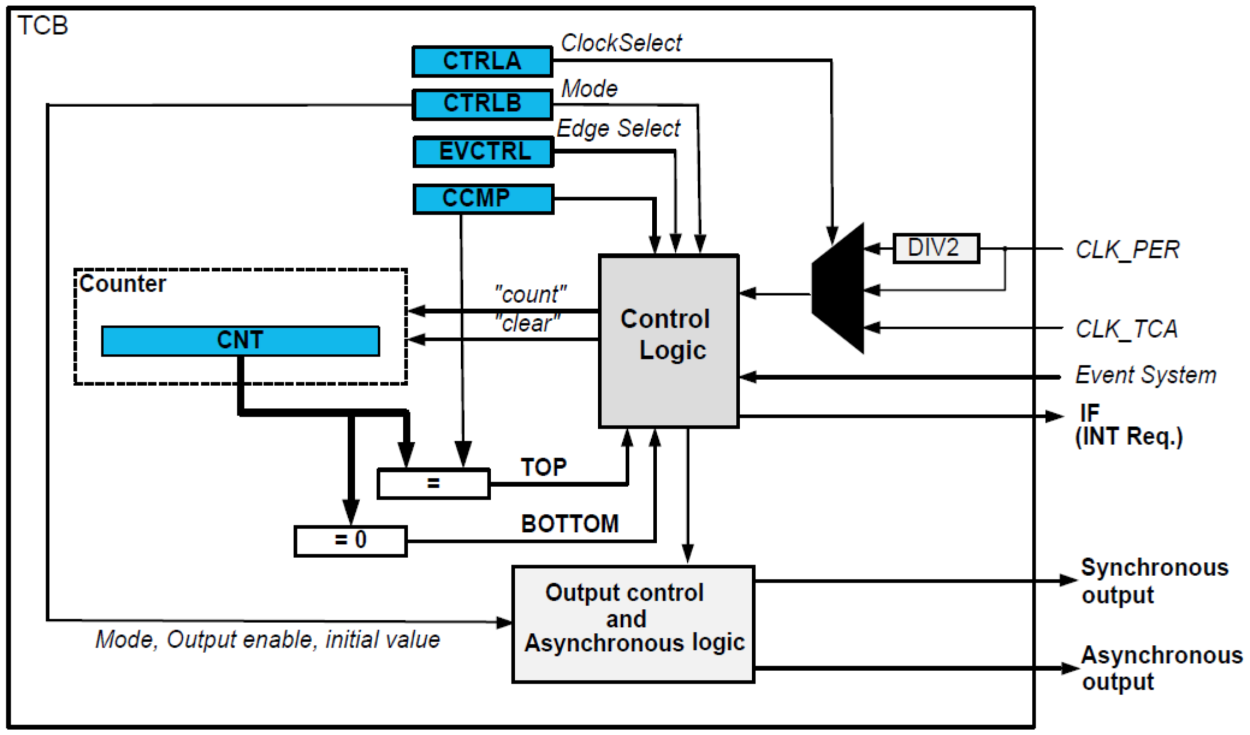
2. Overview

The capabilities of the 16-bit Timer/Counter type B (TCB) include frequency, waveform generation, and input capture on an event with time and frequency measurement of the digital signals. The TCB consists of a base counter and a control logic that can be set in one of the eight different modes, each mode providing unique functionality. The base counter is clocked by the peripheral clock with optional prescaling, as shown below.

Figure 2-1. Timer/Counter Type-B Block Diagram



3. Using TCB in 8-bit PWM mode

Use case description: TCB (the TCB3 instance) will be configured in 8-bit PWM mode and generate a 1-second period PWM signal, at a 50% duty cycle. A GPIO pin (Port B pin 5 - PB5) will be used as an output to showcase the signal.

Result: TCB will generate a 50% duty cycle PWM signal for a 1-second period. The on-board LED (the PB5 pin) will toggle every 500 ms.

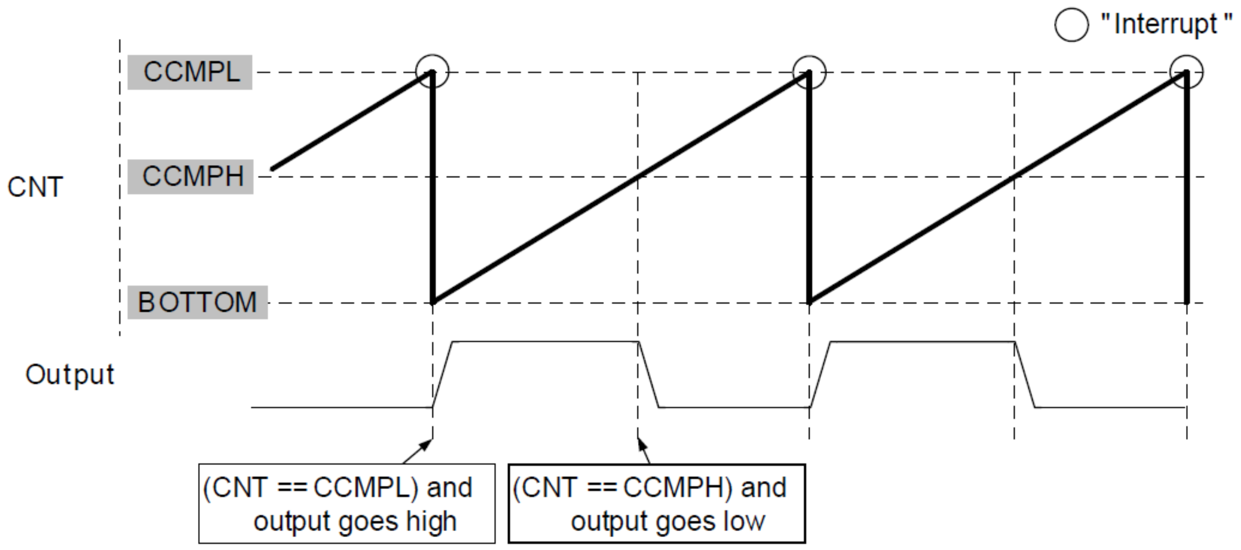
This timer can be configured to run in 8-bit PWM mode where each of the register pairs in the 16-bit Capture/Compare register (TCBn.CCMPH and TCBn.CCMPL) is used as an individual compare register. The counter will continuously count from zero to CCMPL and the output will be set at BOTTOM and cleared when the counter reaches CCMPH.

When this peripheral is in PWM mode and enabled, changing the value of the Capture/Compare register will change the output, but the transition may render invalid values. It is hence recommended to:

1. Disable the peripheral.
2. Write Capture/Compare register to {CCMPH, CCMPL}.
3. Write 0x0000 to count register.
4. Re-enable the module.

CCMPH is the number of cycles for which the output will be driven high, while CCMPL+1 is the period of the output pulse.

Figure 3-1. 8-Bit PWM Mode


"Interrupt"
CCMPL
CCMPH
CNT
BOTTOM
Output
(CNT == CCMPL) and output goes high
(CNT == CCMPH) and output goes low

For example, for a 256 Hz frequency clock that serves as input to the timer, the result is a 1-second period (CCMPL is an 8-bit register, which means it can have values from 0 to 255).

Configuring a pin as output for visualizing the PWM signal

To visualize the PWM, a pin will be configured in the Output mode. The following code sets PB5 as output low.

```
PORTB_DIR |= PIN5_bm;
PORTB_OUT |= PIN5_bm;
```

Configuring the system clock

According to the diagram in Figure 2-1, there are two main clock sources for TCB:

- CLK_PER (the peripheral clock, derived from main clock CLK_MAIN)
- CLK_TCA (the TCA clock, which can be derived from CLK_PER)

For the purpose of this use case of TCB, CLK_PER clock source and instance 3 of TCB will be used (TCB3). In order to get a 1-second period, the input clock must be as low as possible. The following configuration must be made:

- Internal 32 kHz ultra low-power oscillator for CLKSEL must be selected
- The clock prescaler (PEN) must be enabled
- The highest prescaler division (PDIV 64) must be used

In order to use the 32 kHz internal oscillator as clock source for TCB, the user must configure the following registers and bits or bit fields in the following register.

Figure 3-2. MCLKCTRLB Register Configuration

Bit	7	6	5	4	3	2	1	0
				PDIV[3:0]				PEN
Access				R/W	R/W	R/W	R/W	R/W
Reset				1	0	0	0	1

bits 4:1 PDIV[3:0]: Prescaler Division bits
If the Prescaler Enable (PEN) bit is written to '1', these bits define the division ratio of the main clock prescaler.

Value	Description
Value	Division
0x5	64

bit 0 PEN: Prescaler Enable bit
This bit must be written '1' to enable the prescaler. When enabled, the division ratio is selected by the PDIV bit field.

The Main Clock and Prescaler Configuration registers (CLKCTRL.MCLKCTRLA, CLKCTRL.MCLKCTRLB) are protected by the Configuration Change Protection (CCP) mechanism, employing a timed-write procedure for changing these registers. In order to write to these registers, a certain key must be written to the CPU.CCP register first, followed by a write access to the protected bits within four CPU instructions.

The key that must be written to the CPU.CPP register is 'IOREG', which translates into the line of code from below. For more details, check Configuration Change Protection (CCP) in the megaAVR 0-series family data sheet.
Note: The CPU_CPP register will allow only one READ/WRITE instruction and the change must be made in the following four cycles, after writing the key.

```
CPU_CCP = CCP_IOREG_gc;
```

Since the desired frequency is the lowest possible, the Prescaler (PEN) must be enabled and PDIV must be set to ‘64’ divider.

```
CLKCTRL.MCLKCTRLB = CLKCTRL_PDIV_64X_gc | CLKCTRL_PEN_bm;
```

Figure 3-3. MCLKCTRLA Register Configuration

Bit	7	6	5	4	3	2	1	0
	CLKOUT						CLKSEL[1:0]	
Access	R/W						R/W	R/W
Reset	0						0	0

bits 1:0 CLKSEL[1:0]: Clock Select bits
This bit field selects the source for the Main Clock (CLK_MAIN).

Value	Name	Description
0x0	OSC20M	20 MHz internal oscillator
0x1	OSCULP32K	32 KHz internal ultra low-power oscillator
0x2	XOSC32K	32.768 kHz external crystal oscillator
0x3	EXTCLK	External clock

OSCULP32K must be selected, which means CLKSEL bit field must be set to 0x1 value. This translates into the following code:

```
CLKCTRL.MCLKCTRLA = CLKCTRL_CLKSEL_OSCULP32K_gc;
```

Figure 3-4. MCLKSTATUS Register Configuration

Bit	7	6	5	4	3	2	1	0
	EXTS	XOSC32KS	OSC32KS	OSC20MS				SOSC
Access	R	R	R	R				R
Reset	0	0	0	0				0

bit 0 SOSC: Main Clock Oscillator Changing bit

Value	Description
0	The clock source for CLK_MAIN is not undergoing a switch
1	The clock source for CLK_MAIN is undergoing a switch and will change as soon as the new source is stable

The clock switching process is indicated by the SOSC bit. The program should halt during an undergoing switch of the clock source, so a wait until the switch is over should be implemented:

```
while (CLKCTRL.MCLKSTATUS & CLKCTRL_SOSC_bm)
{
    ;
}
```

Configuring the TCB input clock and operation mode

In order to generate the 8-bit PWM signal to PB5, the following registers must be changed:

- TCBn.CCMP
- TCBn.CTRLA
- TCBn.CTRLB

The TCBn.CCMPL and TCBn.CCMPH register pair represents the 16-bit value TCBn.CCMP. The low byte <7:0> (suffix L) is accessible at the original offset. The high byte <15:8> (suffix H) can be accessed at offset +0x01. In 8-bit PWM mode, TCBn.CCMPL and TCBn.CCMPH act as two independent registers.

Figure 3-5. TCBn.CCMP Register Configuration

Bit	15	14	13	12	11	10	9	8
	CCMP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

bits 15:8 CCMP[15:8]: Capture/Compare Value High Byte
These bits hold the MSB of the 16-bit compare, capture, and top value.

bits 7:0 CCMP[7:0]: Capture/Compare Value Low Byte
These bits hold the LSB of the 16-bit compare, capture, and top value.

When running TCB in 8-bit PWM mode, TCBn.CCMPL must be loaded with the PWM signal period, of 1 second in this case. Since the period of the output pulse is defined by TCBn.CCMPL+1, the value that must be loaded into the TCBn.CCMPL register is 0xFF (255 in decimal).

Subsequently, TCBn.CCMPH must be loaded with the number of cycles for which the output will be driven high. The goal is to set the duty cycle at 50% in order to make PB5 toggle every 500 ms.

$$CCMPH = \left(CCMPL + 1\right) \times \frac{50}{100}$$

$$CCMPH = 128$$

This results in TCBn.CCMPH=128=0x80

This means that TCBn.CCMP must be loaded with the following value, obtained from TCBn.CCMPH and TCBn.CCMPL:

```
TCB3.CCMP = 0x80FF;
```

Figure 3-6. TCBn.CTRLA Register Configuration

Bit	15	14	13	12	11	10	9	8
	CCMP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

bits 15:8 CCMP[15:8]: Capture/Compare Value High Byte
These bits hold the MSB of the 16-bit compare, capture, and top value.

bits 7:0 CCMP[7:0]: Capture/Compare Value Low Byte
These bits hold the LSB of the 16-bit compare, capture, and top value.

TCB3 can be enabled by setting the ENABLE bit to '1' in the TCB3.CTRLA register. This translates into the following code:

```
TCB3.CTRLA |= TCB_ENABLE_bm;
```

In order to get the lowest possible frequency, CLK_PER will be further divided by 2, by configuring the CLKSEL bit field in the TCB3.CTRLA register. The corresponding value for CLKSEL in this case is 0x1. This translates into the following code:

```
TCB3.CTRLA |= TCB_CLKSEL_CLKDIV2_gc;
```

Figure 3-7. TCBn.CTRLB Register Description

Bit	7	6	5	4	3	2	1	0
		ASYNC	CCMPINIT	CCMPEN		CNTMODE[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

bit 4 CCMPEN: Compare/Capture Output Enable bit
This bit is used to enable the output signal of the Compare/Capture.

Value	Description
`0`	Compare/Capture Output is zero
`1`	Compare/Capture Output has a valid value

bits 2:0 CNTMODE[2:0]: Timer Mode bits
Writing these bits selects the Timer mode.

Value	Description
`0x0`	Periodic Interrupt mode
`0x1`	Time-out Check mode
`0x7`	8-Bit PWM mode

CCMPEN must be enabled. This translates into the following code:

```
TCB3.CTRLB |= TCB_CCMPEN_bm;
```

TCB must be configured for the 8-bit PWM mode. This translates to the following code:

```
TCB3.CTRLB |= TCB_CNTMODE_PWM8_gc;
```

View Code Example on GitHub
Click to browse repository

Tip: The full code example is also available in the Appendix section.