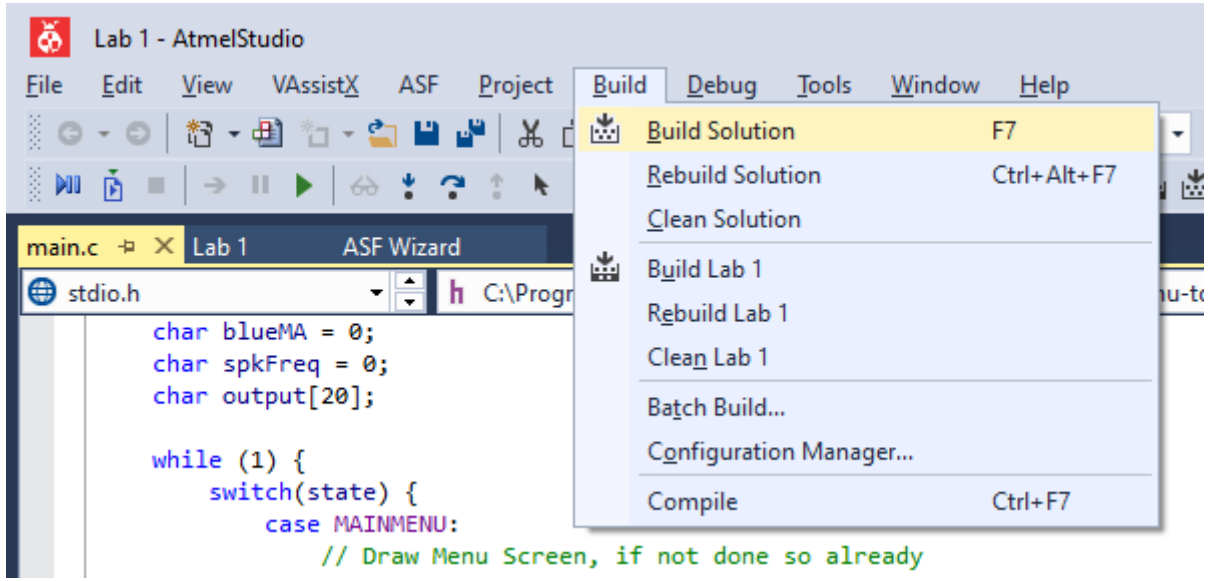
Checking your Code

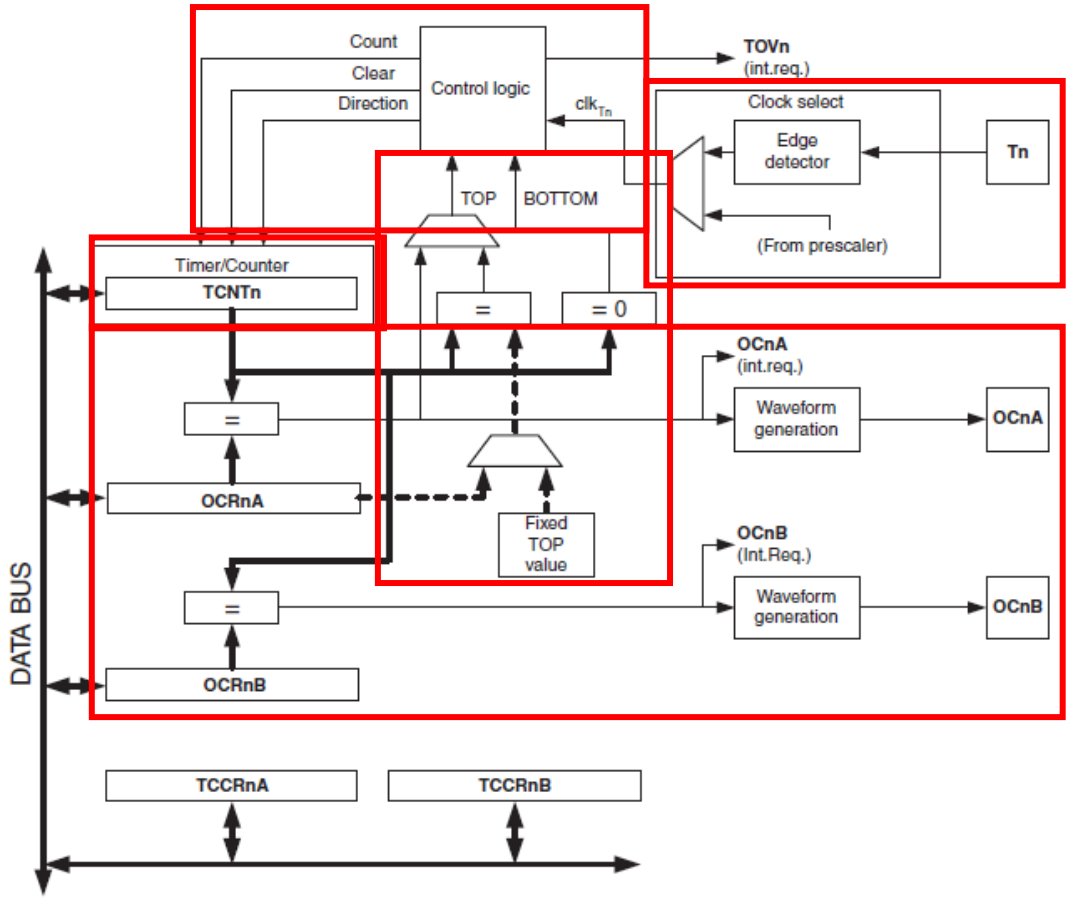


Use the Build function (F7) in ATMEL studio to check for simple errors:

- Syntax Errors
- Undeclared Variables / Variables not in scope
- Incorrect use of functions

Timers

- Clock Select
  - Based off f_{osc} (and Prescaler) <u>or</u> from external source (via Tn Pin)
- Clock Logic
  - This instructs the timer/counter what to do next
  - Considers parameters from other inputs (clock pulse, TOP & BOTTOM values, setup registers)
- Timer/Counter
  - Register TCNTn
  - Will either count up, down, or reset to TOP/BOTTOM value
- Output Compare Registers
  - OCRnA/OCRnB
  - Control the outputs on OCnA/OCnB pins on the µC
  - OCRnA can also influence the Control Logic



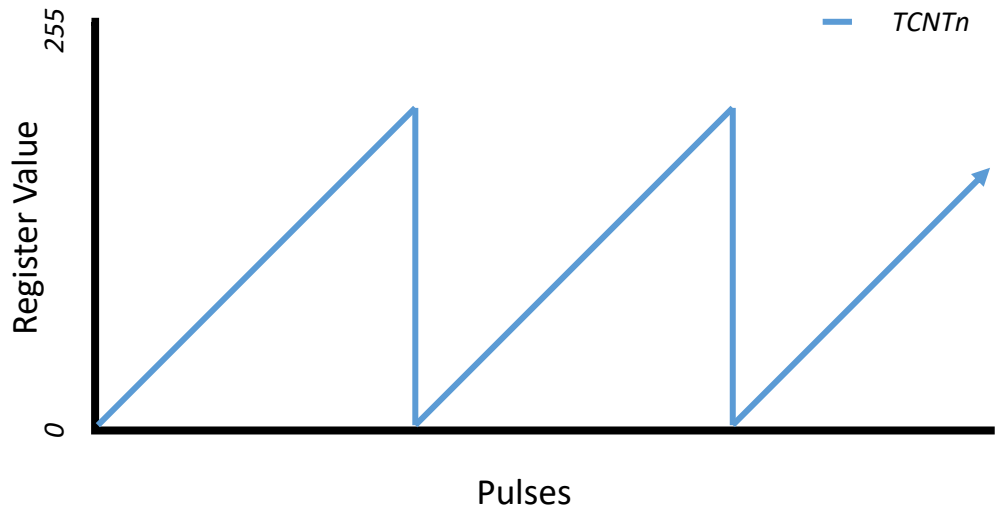
Timer Modes

- There are four different modes
  - Normal
  - Clear Timer on Compare Match (CTC)
  - Fast PWM
  - Phase-Correct PWM

- For this week, we will be able to use either Normal Mode or CTC
  - CTC requires less of the µC's resources
  - Normal mode is more involved to operate (but can be easier to understand)

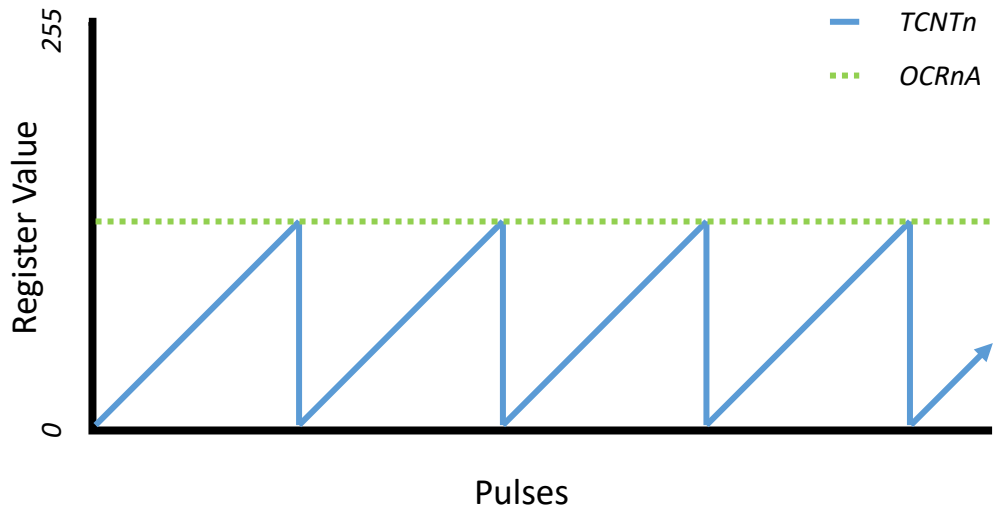
Modes - Normal

- Timer counts up from 0 to 0xFF (25)
- Events can be set on the timer to occur when the values in Output Compare Registers match
- Four Compare Match Modes
  - Normal
  - Toggle
  - Clear
  - Set

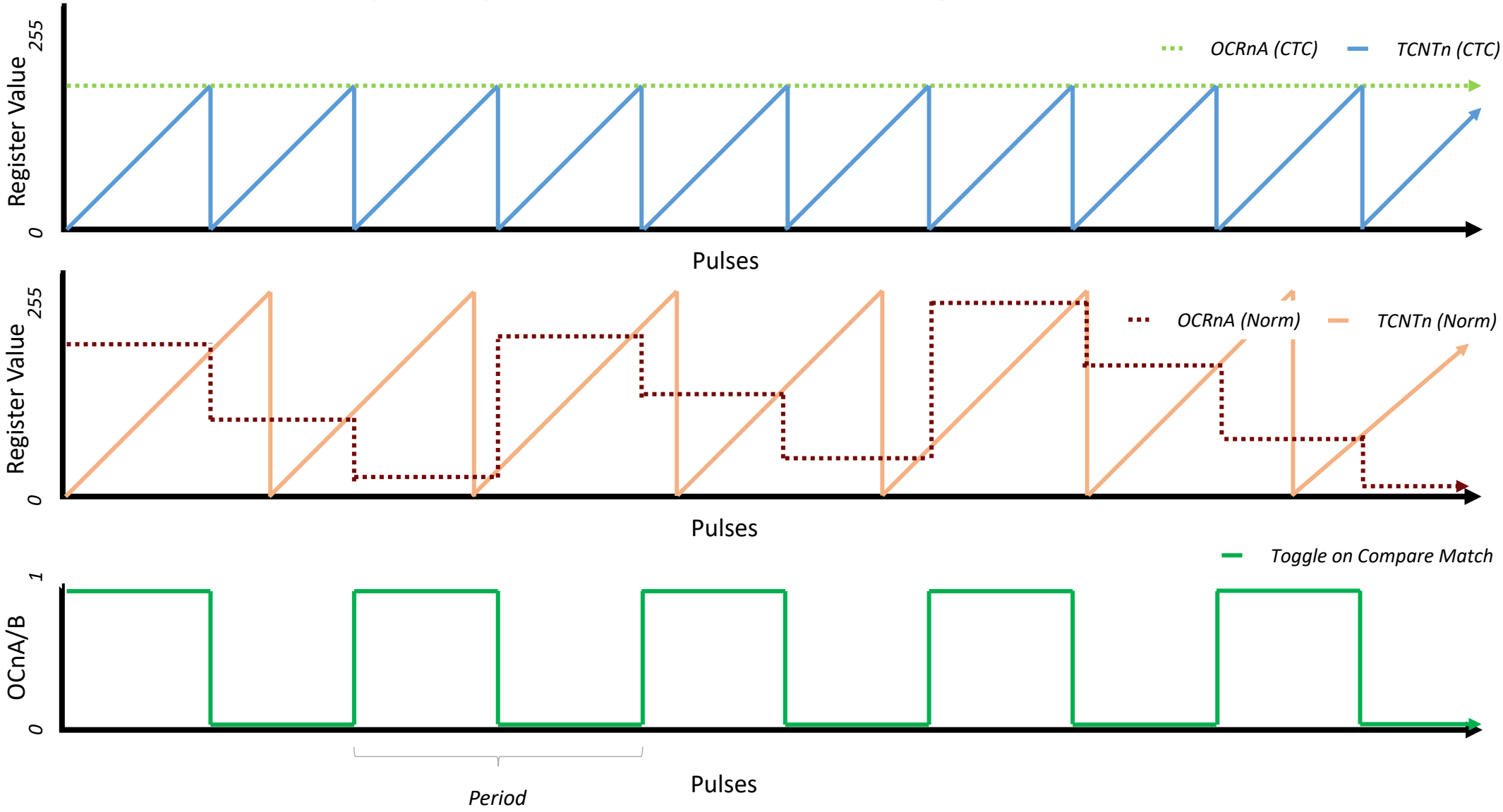


Modes - Clear on Compare Match (CTC)

- Timer counts up from 0 to the TOP value stored in OCRnA
- When reached, the timer clears to 0, and an Interrupt is triggered (if enabled)
- Four Compare Match Modes
  - Normal
  - Toggle
  - Clear
  - Set



Generating Square Wave Signals



Calculating Top Values for Frequencies

$$f_{PWM} = \frac{f_{osc}}{2 \times Prescaler \times (Top + 1)} \quad \leftrightarrow \quad Top = \frac{f_{osc}}{2 \times Prescaler \times f_{PWM}} - 1$$

Example – Speaker Sound Generator

Let's suppose we want to drive a speaker:

- Have a variable frequency from 1kHz to 20kHz
- Speaker is connected to PB4 (OC2A)
- Uses the Timer to generate the frequency w/o code execution

What registers will we need to consider?

- Timer 2 (connected to Speaker)
  - TCCR2A
  - TCCR2B
  - TIMSK2

Example – Speaker Sound Generator

Table 16-7. Waveform Generation mode bit description.

Mode	WGM2	WGM1	WGM0	Timer/Counter mode of operation	TOP	Update of OCRx at	TOV flag set on (1)(2)
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, phase correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	TOP	MAX
4	1	0	0	Reserved	–	–	–

- These are some of the available modes
  - These will affect the timer frequency, and therefore the output PWM frequencies we can generate
- There are two approaches – CTC and Normal Mode
  - Both are included

Example – Speaker Sound Generator (CTC)

Determine Register Values: TCCR2B

Bit	7	6	5	4	3	2	1	0	
	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20	TCCR2B
Read/write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- FOC2A/FOC2B – Force Output Compare A/B
  - Not used
  - Hence, set to `00`
- WGM22– Waveform Generation Mode
  - We will be using CTC (even though it makes sense to use Fast PWM)
  - Hence, set to `0`
- CS22 – CS20 – Clock Select
  - Let us determine the necessary prescaler…

Example – Speaker Sound Generator (CTC)

Table 16-8. Clock Select bit description.

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	clk_{T2S}/(no prescaling)
0	1	0	clk_{T2S}/8 (from prescaler)
0	1	1	clk_{T2S}/32 (from prescaler)
1	0	0	clk_{T2S}/64 (from prescaler)
1	0	1	clk_{T2S}/128 (from prescaler)
1	1	0	clk_{T2S}/256 (from prescaler)
1	1	1	clk_{T2S}/1024 (from prescaler)

- These are the available prescalers
  - These will affect the timer frequency, and therefore the output PWM frequencies we can generate
- Note: Register Booklet contains the wrong table
  - Use the table above for Timer 2

Example – Speaker Sound Generator (CTC)

What should we set the Prescaler to? Let us use trial & error to calculate a value.

$$f_{PWM} = \frac{f_{osc}}{2 \times Prescaler \times (Top + 1)} \rightarrow f_{PWM} = \frac{4 \times 10^6}{Prescaler \times (Top + 1)}$$

Firstly, try with Prescaler = 1 – Find the TOP Value @ $f_{pwm(min)}$ = 1 kHz:

$$1000 = \frac{4 \times 10^6}{1 \times (Top+1)} \rightarrow Top = 3999 \rightarrow \therefore \text{Prescaler} \neq 1$$

Now, try with Prescaler = 8 – Find the TOP Value @ $f_{pwm(min)}$ = 1 kHz:

$$1000 = \frac{4 \times 10^6}{8 \times (Top+1)} \rightarrow Top = 499 \rightarrow \therefore \text{Prescaler} \neq 8$$

Now, try with Prescaler = 32 – Find the TOP Value @ $f_{pwm(min)}$ = 1 kHz:

$$1000 = \frac{4 \times 10^6}{8 \times (Top+1)} \rightarrow Top = 124 \rightarrow \therefore \text{Prescaler} = 32$$

Prescaler is 32

Example – Speaker Sound Generator (CTC)

Determine Register Values: TCCR2B

Bit	7	6	5	4	3	2	1	0	
	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20	TCCR2B
Read/write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

```
// Define the value of TCCR2B
TCCR2B = 0b00000011;
```

- FOC2A/FOC2B – Force Output Compare A/B
  - Not used
  - Hence, set to `00`
- WGM22– Waveform Generation Mode
  - We will be using CTC (even though it makes sense to use Fast PWM)
  - Hence, set to `0`
- CS22 – CS20 – Clock Select
  - Prescaler of 32
  - Hence, set to `011`

Example – Speaker Sound Generator (CTC)

How do we update the TOP Value? Simply by setting the TOP value OC2A:

```
// Update the CTC value for the desired frequency

OC2A = 125 / freqInKHZ;
```

$$Top = \frac{f_{osc}}{2 \times Prescaler \times f_{PWM}} - 1$$

$$OC2A = \frac{125}{f_{PWM\ (kHz)}}$$

Example – Speaker Sound Generator (Normal)

Determine Register Values: TCCR2B

Bit	7	6	5	4	3	2	1	0	
	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20	TCCR2B
Read/write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

```
// Define the value of TCCR2B
TCCR2B = 0b00000011;
```

- FOC2A/FOC2B – Force Output Compare A/B
  - Not used
  - Hence, set to `00`
- WGM22– Waveform Generation Mode
  - We will be using CTC (even though it makes sense to use Fast PWM)
  - Hence, set to `0`
- CS22 – CS20 – Clock Select
  - Prescaler of 32
  - Hence, set to `011`

Example – Speaker Sound Generator (Normal)

How do we update the toggle value? Simply by adding a representative amount of time to OC2A in an Interrupt.

```
// Update the CTC value for the desired
frequency – Call this in the interrupt

OCR2A = OCR2A + 125/freqInKHZ;
```

$$Top = \frac{f_{osc}}{2 \times Prescaler \times f_{PWM}} - 1$$

$$OC2A = OC2A + \frac{125}{f_{PWM\ (kHz)}}$$