Experiment 10

PWM Generation Using Timers

The Tiva-C General-Purpose Timer Module (GPTM) in TM4C123 contains six 16/32-bit GPTM blocks and six 32/64-bit Wide GPTM blocks. Each 16/32-bit GPTM block provides two 16-bit timers/counters(referred to as Timer A and Timer B) that can be configured to operate independently as timers or event counters. When a timer is used as an output device, its main purpose is to generate signals with desired attributes. For instance, we can use timer to generate periodic signal (square wave shape) of certain frequency as well as to generate pulses of varying width. It can also be used to generate time delays.

Pulse Width Modulation

Pulse width modulation (PWM) is a technique to generate variable width pulses, while maintaining a constant signal frequency. PWM signals are widely used in many control applications. Below are some common uses of PWM signals.

1. Motor speed control
2. Power converters
3. Brightness control for LEDs
4. Encoded data transmission

The time period, t_p and the pulse width t_w are the two parameters that must be specified to generate the desired PWM signal, such that $t_w < t_p$. An important attribute of PWM signal is its duty cycle, d, which is defined as a percentage value using the following expression

How to Configure a Timer for PWM

A timer is configured to PWM mode using the following sequence:

1. Configure the respective GPIO pin for alternate functionality.
2. Ensure the timer is disabled (the TnEN bit is cleared) before making any changes.
3. Write the GPTM Configuration (GPTMCFG) register with a value of 0x0000.0004.
4. In the GPTM Timer Mode (GPTMTnMR) register, set the TnAMS bit to 0x1, the TnCMR bit to 0x0, and the TnMR field to 0x2.
5. Configure the output state of the PWM signal (whether or not it is inverted) in the TnPWML field of the GPTM Control (GPTMCTL) register.
6. If a prescaler is to be used, write the prescale value to the GPTM Timer n Prescale Register (GPTMTnPR).

7. If PWM interrupts are used, configure the interrupt condition in the TnEVENT field in the GPTMCTL register and enable the interrupts by setting the TnPWMIE bit in the GPTMTnMR register. Note that edge detect interrupt behavior is reversed when the PWM output is inverted
8. Load the timer start value into the GPTM Timer n Interval Load (GPTMTnILR) register.
9. Load the GPTM Timer n Match (GPTMTnMATCHR) register with the match value.
10. Set the TnEN bit in the GPTM Control (GPTMCTL) register to enable the timer and begin generation of the output PWM signal.

In PWM Time mode, the timer continues running after the PWM signal has been generated. The PWM period can be adjusted at any time by writing the GPTMTnILR register, and the change takes effect at the next cycle after the write.

Note1: *The register map for System Control module (for RCGCTIMER) is to be consulted from article* ***5.4 - System Control - Register Map*** *of the controller datasheet.*

Note2: *The register map for Timers is to be consulted from article* ***11.5 - General-Purpose Timers - Register Map*** *of the controller datasheet.*

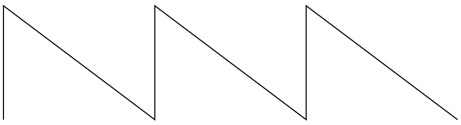
Note3: *Consult table 2-8 and 2-9 for the entries of the vector table from article* ***2.5.2 The Cortex-M4F Processor - Exception Types*** *of the datasheet.*

Note4: *The register map for NVIC is to be consulted from article* ***3.2 - Cortex-M4 Peripherals - Register Map*** *of the controller datasheet.*

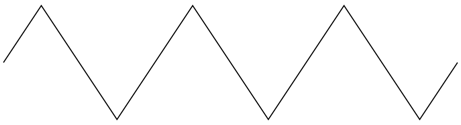
Note5: *The register map for GPIO is to be consulted from article* ***10.4 - General-Purpose Input/Outputs (GPIOs) - Register Map*** *of the controller datasheet.*

Exercise

Configure PWM for Timer(your column number in the lab) for a period of 2*column number to display first half of saw tooth or triangular wave when SW1 is pressed and the rest when SW2 is pressed.



sawtooth



triangle