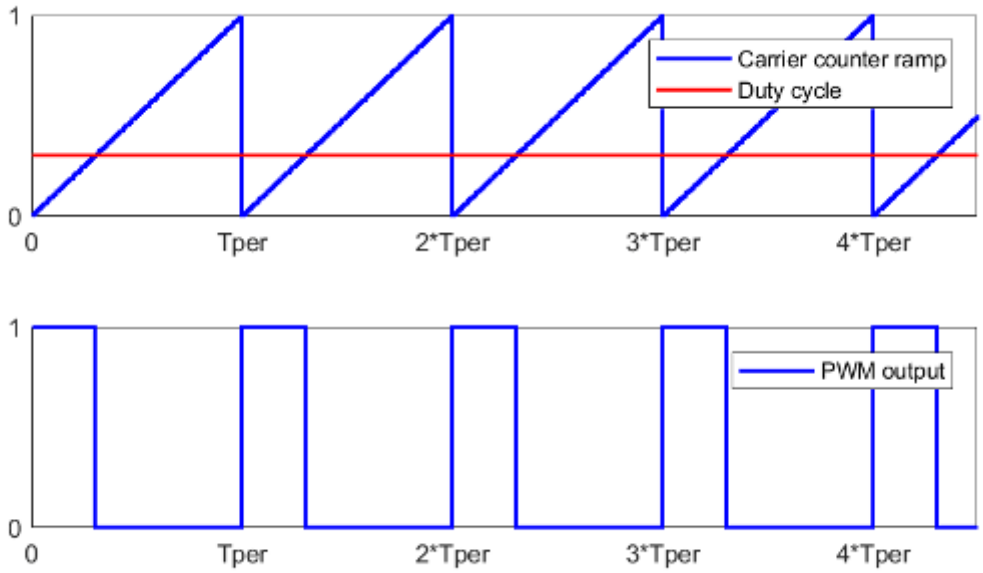
Lab2 Report

PWM

The PWM is an electrical signal that is turned on and off very quickly. Due to the inertia of the electrical circuitry this creates an average voltage over each period that is proportional to the duty cycle. The duty cycle is the portion of each period that the output of the PWM is high, as illustrated in the figure below. By controlling the duty cycle the average voltage of each period can be precisely controlled. This is what allows us to control the intensity of the LED.



Source: [PWM Generator - Generate pulse width modulated signal or waveform - Simulink - MathWorks Nordic] (2024-09-27)

Source Code

In our code, to set the PWM duty cycle we call `PWMPulseWidthSet`. This function requires the pin base, in this case `PWM0_BASE`, the output bits `PWM_OUT_2` and an integer value that controls the width of the PWM. We set this integer value as a portion of the total number of system clock ticks per period, shown in the code block below:

```c
PWMPulseWidthSet(PWM0_BASE, PWM_OUT_2, pwm_word/2);
```

For the special cases of 0% and 100% we instead disable the PWM peripheral and use the pin as a GPIO output to completely enable or disable the LED, which is shown in the code block below:

```c
void enableGpio(void)
{
    // If PWM is enabled, we need to disable it.
    if (SysCtlPeripheralReady(LED_PWM_PERIPH))
    {
        SysCtlPeripheralDisable(LED_PWM_PERIPH);
    }

    GPIOPinTypeGPIOOutput(LED_GPIO_BASE, LED_GPIO_PIN);
}
```

When we go back to a value that is between 0% and 100% we reenable the PWM peripheral.

Authors:

- Erik Folkesson
- Johan Sandred