Pulse Width Modulation (PWM)

Pulse width modulation (PWM) is a digital-to-analog conversion method by varying the duty cycle of a square wave from 0% to 100%. SiliconLab-C8051F120) can be effectively used to generate the desired PWM signal, e.g., a hex number 0x00 represent 0% duty cycle and 0xFF 100%. PWM signals are routinely used to control the rotating speed of DC motors through an H-bridge amplifier. C8051F120 provides a programmable counter array PCA0—consisting of a dedicated 16-bit counter/timer and six 16-bit capture/compare modules. PCA0 can be enabled by routing its I/O lines (CEX0) to the crossbar. To configure the module to run in 8-bit PWM mode, it must be properly initialized. For example, PCA Counter/Timer is to be enabled by setting the PCA0 Counter/Timer Run Control bit in the PCA0CN register. In addition, ECOM0 and PWM0 bits in PCA0CPM0 should be set in order to enable 8-Bit PWM mode. An initialization example is given below.

```
SFRPAGE   = PCA0_PAGE;
PCA0CN    = 0x40;           // Enable the PCA Counter/Timer
PCA0CPM0  = 0x42;           // ECOM = 1
                            // PWM = 1
                            // PCA0 Module 0 generates PWM on designated CEX0 pin
PCA0CPH0  = 0x80;           // Set PCA0CPH0 to generate 50% duty cycle (80) 10% - E6
```

The last statement `PCA0CPH0 = 0x80` is to generate a 50% duty cycle square wave. The frequency of a PWM signal depends on the SYSCLK. Because of the H-bridge amplifier, it is desirable for the PWM frequency to operate around 1 kHz. Therefore, it is appropriate to use the internal oscillator at its lowest frequency (3.0625 MHz) as the SYSCLK, which leads to a PWM signal frequency about 1.02 kHz.

The duty cycle of the PWM signal can be controlled by using the module's capture/compare (PCA0CPL0) SFR. The PWM signal is low while PCA0 counter/timer starts counting and its value in the low byte of the PCA0 counter/timer (PCA0L) is less than the value pre-stored in PCA0CPL0. The PWM signal then switches to high “1” after PCA0L increases above the value pre-stored in PCA0CPL0. After the count in PCA0L overflows, the PWM signal changes back to low again and the process repeats. Therefore, the duty cycle of the PWM signal is varied by changing the value stored in the PCA0CPL0 register.

$$Duty\_Cycle = \frac{(256 - PCA0CPH0)}{256}$$

PCA0CPH0 is used in the above formula instead of PCA0CPL0 because in reality when the counter/timer low byte (PCA0L) overflows from 0xFF to 0x00, PCA0CPL0 is reloaded automatically with the value stored in PCA0CPH0.