A faster PWM-based DAC

Dusan Ponikvar - December 30, 2015

When you need an analog output from a microcontroller that does not have a digital-to-analog converter (DAC), you can connect an external DAC chip. But for a cheaper solution, use a pulse-width modulated (PWM) output and add a low-pass filter (LPF) to extract its average value, which equals the duty cycle of the PWM signal.

Figure 1 An RC LPF extracts the average value of a PWM signal

The RC filter removes the non-DC components; what remains is the average signal $U_{OUT}$. If the period $T$ of the PWM signal equals 63 clocks, the signal $U_{OUT}$ can have one of 64 discrete DC values (0 to 63, six-bit resolution).

The time constant $t$ of the low-pass RC filter must be big enough to smooth the output signal $U_{OUT}$. The ripple, $\Delta U_{OUT}$, should be less than one least significant bit (LSb). The worst case occurs with a duty cycle of 50% (Figure 2). If $t$ is much bigger than the period $T$ then the capacitor charging current $I_c$ and the change $\Delta U_{OUT}$ can be approximated as:

$$I_c \approx \frac{V_{CC}}{2R} \quad \text{and from:} \quad C \Delta U_{OUT} = I_c \frac{T}{2} \quad \rightarrow \quad \Delta U_{OUT} \approx \frac{V_{CC} T}{4RC}$$

For a 6-bit DAC, $\Delta U_{OUT}$ should be less than $V_{CC}/64$, requiring a filter of $t = RC = 16 \cdot T$. 
Some practical numbers: low-power microprocessors often use a crystal oscillator of 32768 Hz, and this clock signal is used for the PWM block. With 6-bit PWM, the period $T$ is $64/32768 \approx 2\text{ms}$, necessitating a time constant of $32\text{ms}$. One has to wait $5t$ (160ms) for a 6-bit converter to settle. Slow. This Design Idea explains how you can speed things up.

PWM blocks in microcontrollers can usually generate more than one PWM signal. Consider summing outputs from two PWM based 3-bit DACs (DACH and DACL), where the output of DACL is reduced to one-eighth amplitude before adding. The resulting signal acts as a 6-bit DAC, with an important advantage over the simple version: the period $T$ is only eight clock periods for the same resolution, and the required time constant $t$ is $1/8$ of the previous, speeding up settling time by factor of eight. Such an arrangement is easy to implement with resistors in the RC filter for two PWM signals (PWMH, PWML):

The output signal $U_{OUT}$ is given by:
\[ U_{OUT} = \left( DUTY_{PWMH} + \frac{DUTY_{PWML}}{8} \right) \cdot \frac{V_{CC} \cdot R_L}{R_H + R_L} \]

This technique has been implemented in a TI MSP430F5132 microcontroller:

// configure PWM - 32 kHz / 8 = 4 kHz :: 6 bit in two PWMs, done only once on power-up
TA0CCR0 = 7; // count up to 7 (including)
TA0CTL = TASSEL_ACLK | MC_1 | TACLR;
TA0CCR1 = 0; TA0CCTL1 = OUTMOD_6; // toggle/set
TA0CCR2 = 0; TA0CCTL2 = OUTMOD_6; // toggle/set

// use :: write to PWM block to achieve the desired DAC output
DAClevel++; // next DAC level, DAClevel is a char
TA0CCR1 = (DAClevel >> 3) & 7; // set PWMH: MSB 3 bits
TA0CCR2 = (DAClevel ) & 7; // set PWML: LSB 3 bits

Figure 4 Code to initialize & write to a 6-bit (3+3) PWM-based DAC
Figure 5  Measured outputs from 6-bit PWM-based DACs; Blue: implementation as in Figure 1 (160ms settling); Violet: implementation as in Figure 3 (20ms settling)

A 7-bit DAC can be implemented using 1% resistors. This time, two PWM signals are used to make two three-bit DACs, totaling six bits, and the MSb is simply set to 0 or 1 at P3.7.

Figure 6  An implementation of a seven-bit PWM-based DAC
// configure PWM - 32 kHz / 8 = 4 kHz :: 7 bit in two PWMs and one digital pin, **done only once on power-up**

// is the **same** as given in **configure** section in figure 5

// **use** :: write to Timer comparators to achieve the desired DAC output

DAClevel++; // next DAC level, DAClevel is a char

TA0CCR1 = (DAClevel >> 3) & 7; // set PWMH, MSB, 3 bits

TA0CCR2 = (DAClevel ) & 7; // set PWML, LSB, 3 bits

if (DAClevel & BIT6) P3OUT |= BIT7; else P3OUT &= ~BIT7; // set MSB, no PWM

**Figure 8** Code to initialize & write to a 7-bit (3+3+1) PWM-based DAC

The speed improvement here is even greater. A simple PWM DAC would have a period of 128 clock periods (128/32768 s⁻¹ = 3.9ms), resulting in t of 32·T = 125ms and a settling time of 5 · 125ms = 625ms. **Figure 7** settles in 40ms — 16× faster. A higher-order LPF would also help to reduce settling...
time.

**References:**

- Stephen Woodward: [Fast-settling synchronous-PWM-DAC filter has almost no ripple](#)
- David M. Alter: [Using PWM Output as a Digital-to-Analog Converter on a TMS320F280x Digital Signal Controller](#)
- Mike Mitchell: [Using PWM Timer_B as a DAC](#)
- Amar Palacherla: [AN538: Using PWM to Generate Analog Output](#)

**Also see:**

- [Three paths to a free DAC](#)
- [Changing the PWM signal to dc](#)