AN2182
Application note

Filters using the ST10 DSP library

Introduction

The ST10F2xx family provides a 16-bit multiply and accumulate unit (MAC) allowing control-oriented signal processing and filtering widely used in digital applications.

An ST10 DSP software library, developed by STMicroelectronics, contains a set of basic arithmetic operations such as multiplication as well as two main filter functions, FIR (finite impulse response) and IIR (infinite impulse response), mainly used in digital signal processing.

The first chapter of this application note describes a theoretical digital implementation of four different filters:

- Low-pass filter
- High-pass filter
- Passband filter
- Cut-band filter

The method adopted for each filter is the approximation of the ideal filter model by a FIR filter. This theory aims to compute the FIR's coefficients by truncating the real signal with a known window.

The second chapter illustrates a practical implementation of a low-pass filter using the ST10 DSP library, its results and its limitations.

Contents

1 ST10 DSP library

The ST10 DSP free library is a set of arithmetic and signal processing functions based on the ST10 MAC unit. These functions are callable from C and fully compatible with the Tasking compiler.

This library manipulates signed integers coded on 16 or 32 bits. These integers represent numbers belonging to the interval [–1, 1[. We name these formats: Q1.31 and Q1.15.

Table 1. Examples of integer representations

	–1	**-0.5**	**-0.25**	**0**	**0.25**	**0.5**	**1 - 1/2^{15}**	**1 - 1/2^{31}**
Q1.15	0xFFFF	0xC000	0xA000	0x0000	0x2000	0x4000	0x7FFF	0x7FFF
Q1.31	0xFFFF FFFF	0xC000 0000	0xA0000 000	0x00000 000	0x20000 000	0x40000 000	0x7FFF 0000	0x7FFFFFFF

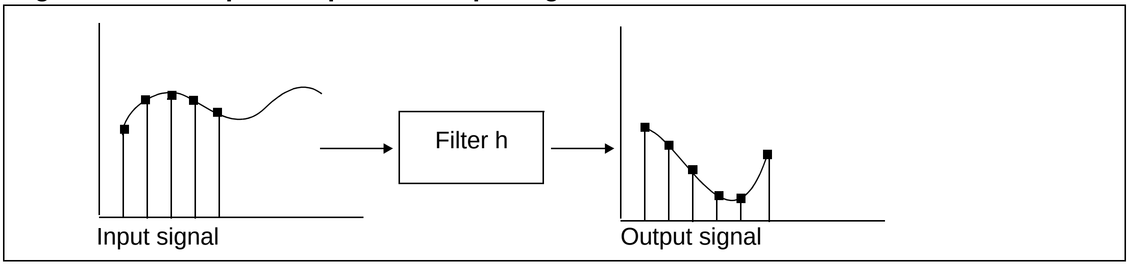
For a detailed description of the ST10 DSP library, please refer to the application note AN1442 “Signal processing with ST10-DSP”.

2 Digital filtering principles

Assume a continuous signal x(t) (the complex form corresponds to the signal's phase and magnitude at the instant t) with a pass band B. Assume that this signal will be filtered using a filter with a continuous impulse response h(t).

When digital processing has to be used, it is necessary to sample the input signal with a frequency of $F_S = 1/T_S$ (T_S being the sampling period). The output signal is then reconstituted from the samples obtained at the filter's output.

Figure 1. Example of input and output signals



The Shannon theorem states that when sampling a signal at discrete intervals, the sampling frequency F_S should be greater than twice the highest frequency of the input signal.

2.1 Fourier transform of a sampled signal

Signals are converted from the time domain to the frequency domain usually through the Fourier transform. With the Fourier transform, the signal is converted to a magnitude and phase at each frequency.

The Fourier transform of the sampled signal x(k) has the following expression:

$$X(f) = \sum_{-\infty}^{\infty} x(k) e^{i2k\pi f}$$

The time representation can be computed from the X(f) as follows:

$$x(k) = \sum_{-\infty}^{\infty} X(f) e^{-i2k\pi f}$$

2.2 Linear filtering

Using the notations defined in the previous section, the output signal y(n) is the convolution of the input signal x(k) and the filter impulse response h(k) .

$$y(n) = \sum_{-\infty}^{\infty} h(n-k)x(k) = (h(k) \otimes x(k))|_n$$

where $x(n) = x(nT_S)$, $h(n - k) = h((n - k)T_S)$ and $y(n) = y(nT_S)$.

The output signal frequency response is given by the following expression:

$$Y(f) = H(f) \cdot X(f)$$

where H(f), X(f) and Y(f) are the respective Fourier transforms of h(k), x(k) and y(k).

2.3 Finite impulse response filters

The FIR (Finite Impulse Response) are non-recursive filters, meaning that the output signal y(i) is a linear combination of N input samples x(k) in the case of a N–1 order filter. Its equation is

$$y(n) = \left(\sum_{k=0}^{N-1} a_k x(n-k) \right)$$

where a_k are the FIR's coefficients.

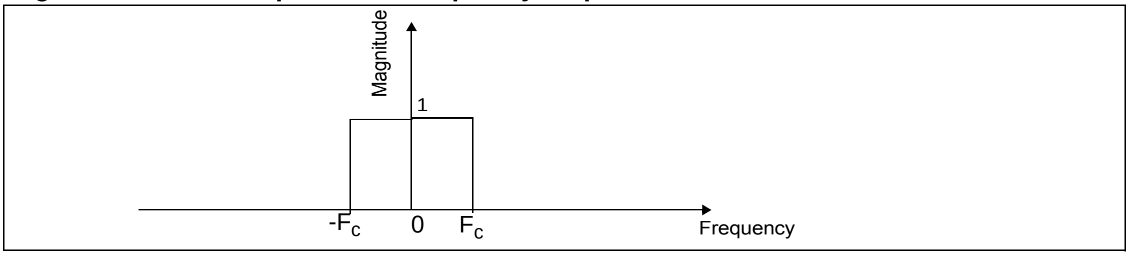
A FIR filter is characterized by its order and its coefficients and can be used to implement any kinds of filters (low-pass, high-pass, pass-band or cutoff band).

3 Low-pass filter

The aim of this section is to create a low-pass filter with a cutoff frequency F_c and a gain G = 1, by determining a FIR filter using the digital approach. The FIR coefficients correspond to h(n) where h is its continuous time response and $h(n) = h(nT_s)$.

In the frequency domain, the ideal filter corresponding to these criteria has the following response:

Figure 2. Ideal low-pass filter frequency response



This filter's impulse response in the time domain is

$$(1)\quad h(t) = \int_{-\infty}^{\infty} H(f)e^{2\pi ift}df = 2F_c \operatorname{sinc}(2tF_c)$$

This response is sampled with a rate $F_s = 1/T_s$ (sampling frequency), so the discrete response has the following expression

$$(2)\quad h_s(t) = \sum_{-\infty}^{\infty} h(nT_s)\delta(t - nT_s)$$

The impact of sampling h(t) with a rate F_s is a periodization of the analog signal spectrum around F_s and a gain of F_s. In fact, the frequency response corresponding to the sampled impulse response is

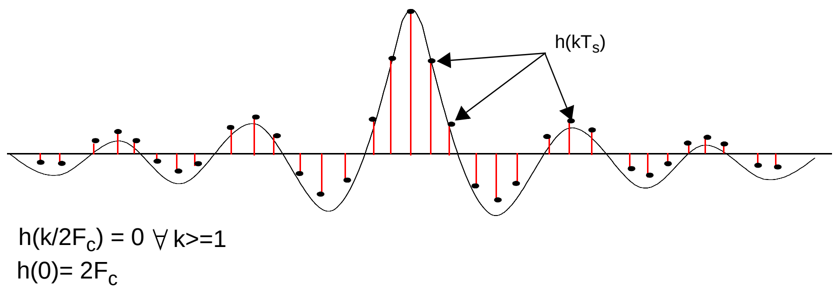
$$H_s(f) = F_s \sum_{-\infty}^{\infty} H(f - nF_s)$$

Therefore, to obtain a gain of 1, the h_s response filter should be divided by F_s. Using equations (1) and (2), the low-pass filter time response becomes:

$$(3)\quad h_s(t) = \sum_{-\infty}^{\infty} 2\frac{F_c}{F_s} \operatorname{sinc}(2nF_cT_s)\delta(t - ns)$$

The impulse response is a sinus cardinal (sinc) function centered at the origin.

Figure 3. Ideal low-pass filter impulse response

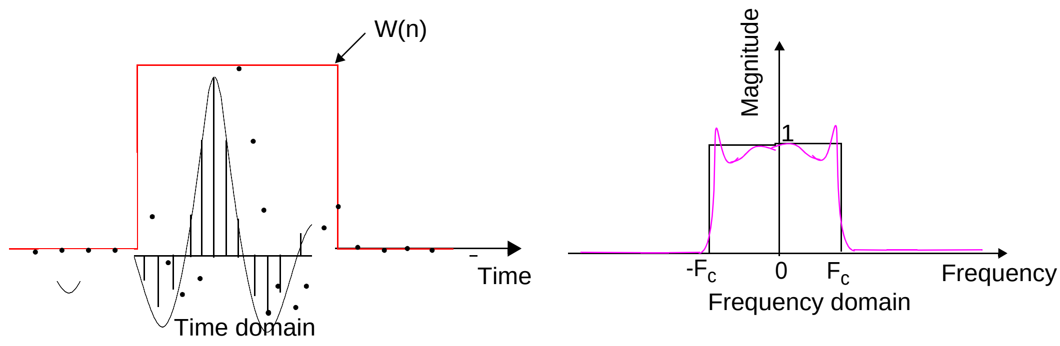


Two issues can be observed at this stage:

First, the filter length is infinite. This means that in order to have the ideal low-pass filter, an infinite number of filter coefficients is required. In practice, this cannot be done because of the calculation complexity. To solve this feasibility problem, the filter's impulse response should be truncated by a known window. In this application note, we will truncate this response using a rectangular window.

Truncating the filter's impulse response, however, changes the ideal rectangular response; more fluctuations are observed in the pass band. In order to reduce this effect, the FIR's coefficients' numbers can be increased.

Figure 4. Effect of truncating the ideal impulse with a rectangular window



In the following sections, we will note:

- W is the window's time response
- N the number of coefficients included in the window W. This means that the window's length is $(N–1)T_S$. The window should be large enough to include at least the first lobe of the sinus cardinal.

Second, the truncated response is finite but not causal because there are non-null coefficients on the negative time axis. This means that the filter reacts before being excited by the input signal. Practically, this filter is not feasible. To solve this issue, we will shift the filter's impulse response by $(N–1)T_S/2$. By doing so, all coefficients on the negative time axis are 0 and the filter becomes feasible.

The impact of shifting the filter's time response is a phase shift of $(N-1) \cdot \pi \cdot \frac{f}{F_S}$ in the frequency domain.

By fixing a number of points N and a sampling period T_S, the FIR coefficients h(n) corresponding to a low-pass filter with a cutoff frequency F_C are

$$(4)\quad h(n) = 2\frac{F_C}{F_S}\operatorname{sinc}\left(2F_C\left(n - \frac{N-1}{2}\right)T_S\right) \cdot W\left(n - \frac{N-1}{2}\right)$$

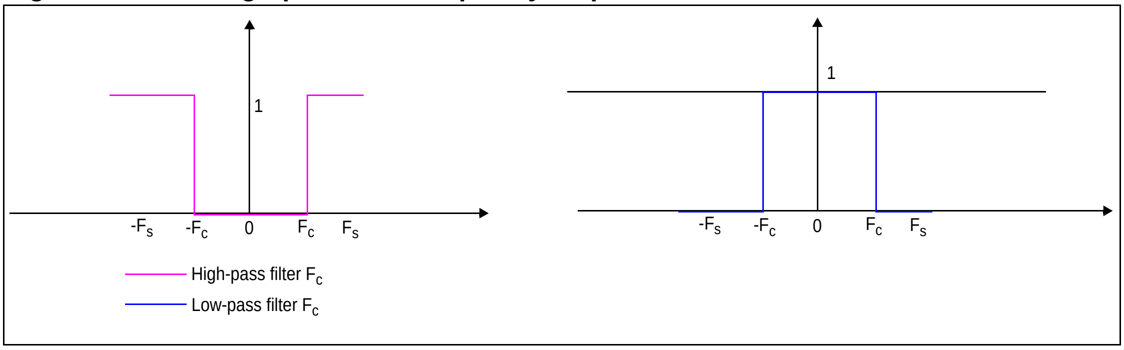
if n = 0 .. N – 1 and 0 if n > N – 1.

4 High-pass filter

The aim of this section is to create a high-pass filter with a cutoff frequency F_c and a gain G = 1, by determining a FIR h filter using the digital approach. The FIR coefficients correspond to h(n) where h is its continuous time response and $h(n) = h(nT_S)$.

In the frequency domain, the ideal high-pass filter corresponding to these criteria has the following response:

Figure 5. Ideal high-pass filter frequency response



If we consider $H_H(f)$ the frequency response of the high-pass filter with a cutoff frequency F_c and $H_L(f)$ the frequency response of the low-pass filter with a cutoff frequency F_c, we can easily notice that

$$(5)\quad H_H(f) = 1 - H_L(f)$$

In the time domain, this gives

$$(6)\quad h_H(n) = \left(\delta\left(n - \frac{N-1}{2}\right) - 2\frac{F_c}{F_S}\operatorname{sinc}\left(2F_c\left(n - \frac{N-1}{2}\right)T_S\right)\right) \cdot W\left(n - \frac{N-1}{2}\right)$$

where N is the number of the filter's coefficients, N–1 is the filter's order and W is the window's time function.

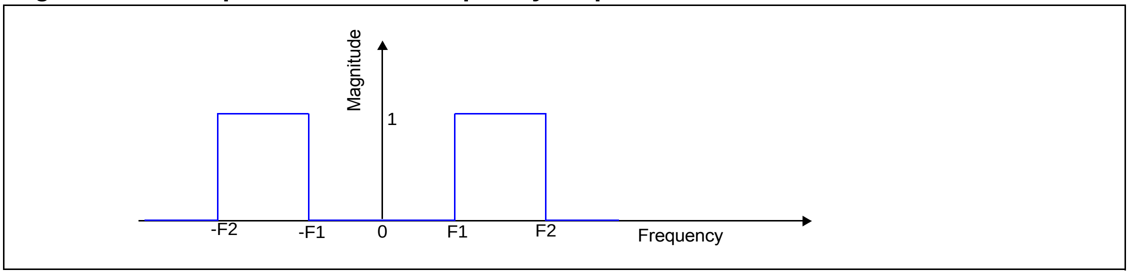
In the case of a high-pass filter, N should be odd because of the dirac.

5 Passband filter

The aim of this section is to create a passband filter with a pass frequency F1, a cutoff frequency F2> F1, and a gain G=1, by determining a FIR h filter using the digital approach.

In the frequency domain, the ideal passband filter corresponding to these criteria has the following response:

Figure 6. Ideal passband filter frequency response



This filter's impulse response in the time domain is

$$h(t) = \int_{-\infty}^{\infty} H(f)e^{2\pi ift}df = \int_{-F2}^{-F1} e^{2\pi ift}df + \int_{F1}^{F2} e^{2\pi ift}df = 2F_2\mathrm{sinc}(2F_2t) - 2F_1\mathrm{sinc}(2F_1t)$$

Here, the same method described in the section entitled *Low-pass filter* is handled. Please refer to this section for more details.

If we fix N the number of the filter's coefficients and T_s the sampling period then we obtain

$$(7)\ h(n) = \left(2\frac{F_2}{F_s}\mathrm{sinc}\left(2F_2\left(n - \frac{N-1}{2}\right)T_s\right) - 2\frac{F_1}{F_s}\mathrm{sinc}\left(2F_1\left(n - \frac{N-1}{2}\right)T_s\right)\right) \cdot W\left(n - \frac{N-1}{2}\right)$$

with n = 0.. N – 1 and where

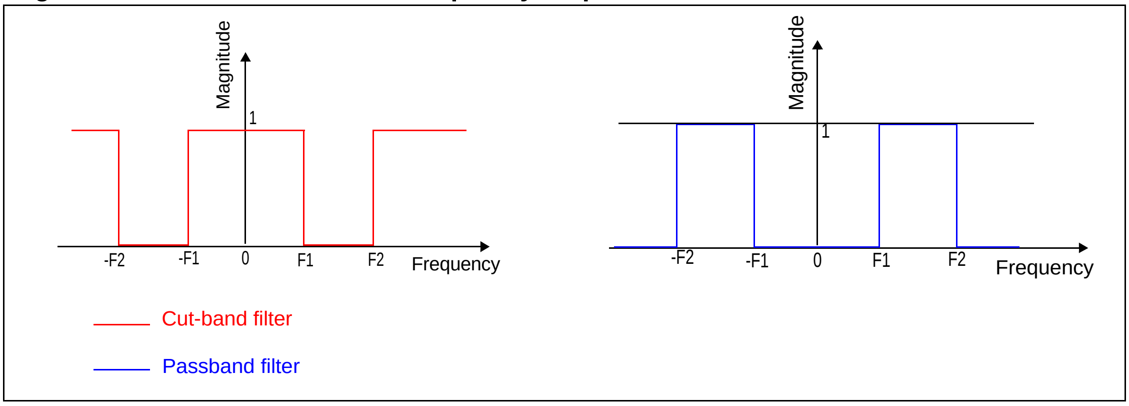
- N is the number of the FIR coefficients
- N – 1 is the FIR's order
- W is the window time response

6 Cutoff band filter

The aim of this section is to create a cutoff band filter with a cutoff frequency F1, a pass frequency F2> F1, and a gain of G=1, by determining a FIR filter using the digital approach. The FIR coefficients correspond to h(n) where h is the continuous time response of the filter and $h(n) = h(nT_S)$.

In the frequency domain, the ideal cutoff band filter corresponding to these criteria has the following response:

Figure 7. Ideal cut-band filter frequency response



If we consider

- $H_P(f)$ the frequency response of the passband filter with a pass frequency F1 and cutoff frequency F2
- $H_C(f)$ the frequency response of the cutoff band filter with a cutoff frequency F1 and pass frequency F2
- T_S is the sampling period

then, we can easily notice that

(8) $H_C(f) = 1 - H_P(f)$

In the time domain, this gives

$$h_C(n) = \delta\left(n - \frac{N-1}{2}\right) - \left(2\frac{F_2}{F_S}\operatorname{sinc}\left(2F_2\left(n - \frac{N-1}{2}\right)T_S\right) - 2\frac{F_1}{F_S}\operatorname{sinc}\left(2F_1\left(n - \frac{N-1}{2}\right)T_S\right)\right)$$

with n = 0.. N – 1 and where

- N is the number of FIR coefficients
- N – 1 is the FIR's order
- W is the window time response

In the case of a cutoff band filter, N should be odd because of the dirac, that is the FIR's order should be even.

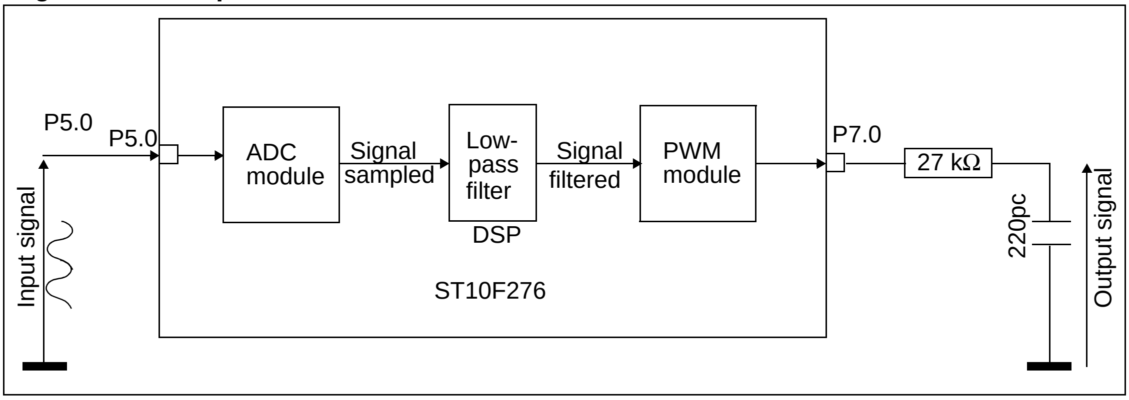
7 Implementation example using the ST10 DSP library

7.1 Overview

In this example, an input signal is fed to an analog input of the ST10F27x, processed by a digital low-pass filter and then output on a PWM output setup as a digital to analog converter.

The digital filter is a low-pass filter with a cutoff frequency equal to 2 kHz and an order of 14 implemented on the ST10F27x using the DSP library provided by STMicroelectronics.

Figure 8. Low-pass filter modules

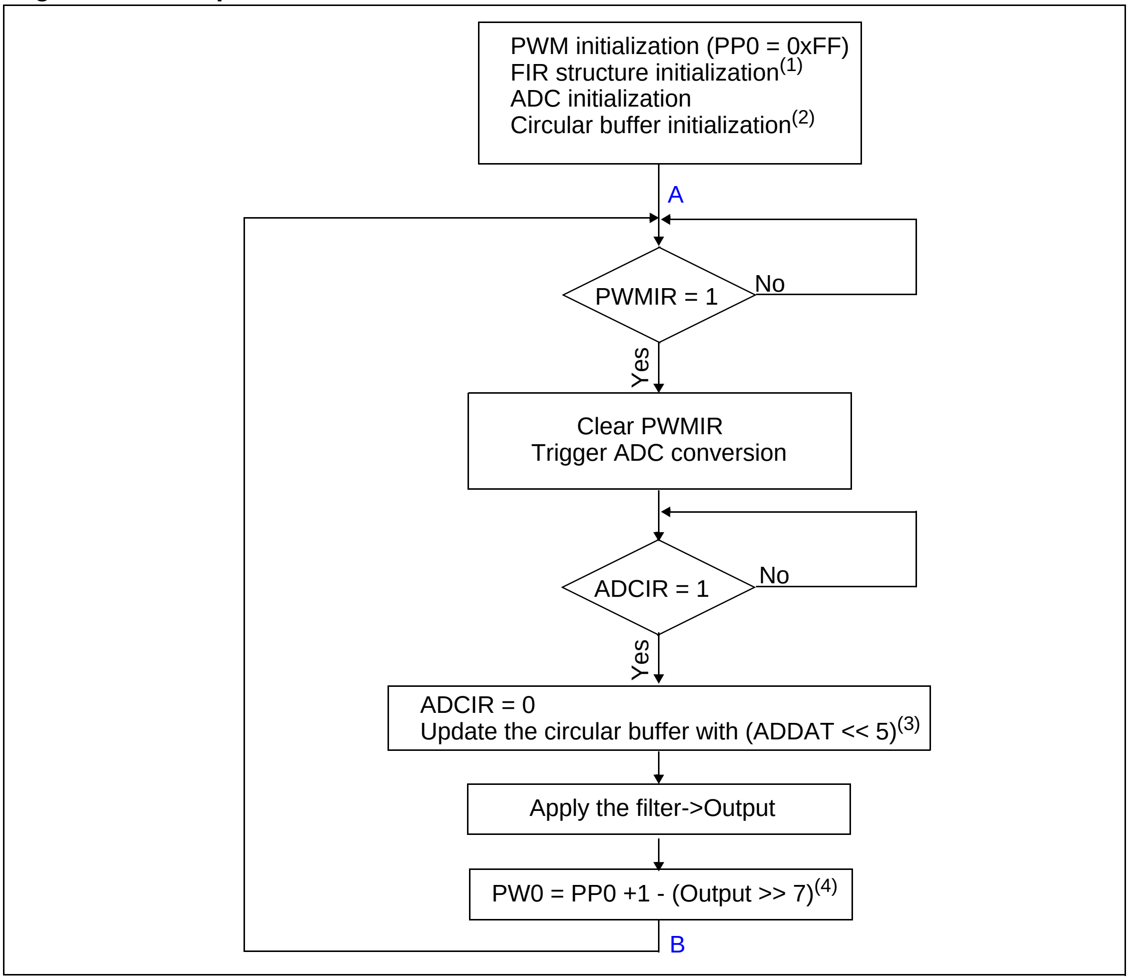


The whole implementation includes the following elements:

- ADC (analog-to-digital converter): the 10-bit ADC of the ST10F27x is used. The input signal is entered on channel 0 (P5.0) of the ST10F27x. The ADC conversion time is 4.85 µs (ADCST = ADCTC and ST10 frequency = 40 MHz). When the conversion is complete, the result stored in the ADDAT register is provided to the FIR filter module.
- Digital filter module: it provides a digital filtered sample. The function used in this application is the fir_q15_q15_q15() provided in the LibST10.h. This means that the inputs, outputs and the FIR coefficients are fractional between –1 and 1 and are coded in the format Q1.15.
- DAC (digital-to-analog converter): this module converts the digital filtered signal into an analog one. An 8-bit DAC is implemented using the ST10F27x PWM and an analog low-pass filter RC. The PWM frequency is 158.8 kHz. The RC filter is used to retain the continuous components and to remove the frequency of the PWM module. In our case, the RC cutoff frequency is 26.8 kHz. The digital filtered sample sets the pulse width of the PWM signal.

The following flowchart summarizes the software implementation of the low-pass filter:

Figure 9. Low-pass filter flowchart



1. Please refer to the technical note ST10 DSP library for a detailed description of the FIR structure used and its fields.
2. If the filter's order is 14, 15 input samples are needed to compute an output. A circular buffer is used to store the 15 input samples of the input signal. The size of the circular buffer is 2 x (filter's order +1), which is 30 in this example. An input sample is stored twice in the circular buffer: at buffer[i] and buffer[i + filter's order +1] where 0<= i<= filter's order.
3. The ST10F27x ADC converts analog voltages between 0 and 5V into 10-bit digital values. The inputs for the fir_q15_q15_q15 should be fractional. To normalize these samples, they are shifted to the left.
4. The FIR function delivers a fractional output coded on 16 bits(Q1.15). The DAC realized with the ST10F27x is an 8-bit one.

7.2 Sampling frequency and FIR coefficients

Using the flowchart in the previous section, the sampling period is the time needed between point A and point B.

To calculate the FIR’s coefficients, the sampling period should be known. We measure the time between points A and B and the result is 17.5 kHz.

Note that to reconstitute the filtered signal, the input signal should not contain frequencies greater than 8.75 kHz.

The FIR’s coefficients are then computed according to this sampling frequency

$$\begin{bmatrix} h(0) \\ h(1) \\ h(2) \\ h(3) \\ h(4) \\ h(5) \\ h(6) \\ h(7) \\ h(8) \\ h(9) \\ h(10) \\ h(11) \\ h(12) \\ h(13) \\ h(14) \end{bmatrix} = \begin{bmatrix} 0.23\,\mathrm{sinc}(-1.61) \\ 0.23\,\mathrm{sinc}(-1.37) \\ 0.23\,\mathrm{sinc}(-1.14) \\ 0.23\,\mathrm{sinc}(-0.914) \\ 0.23\,\mathrm{sinc}(-0.68) \\ 0.23\,\mathrm{sinc}(-0.46) \\ 0.23\,\mathrm{sinc}(-0.23) \\ 0.23\,\mathrm{sinc}(0) \\ 0.23\,\mathrm{sinc}(0.23) \\ 0.23\,\mathrm{sinc}(0.46) \\ 0.23\,\mathrm{sinc}(0.68) \\ 0.23\,\mathrm{sinc}(0.914) \\ 0.23\,\mathrm{sinc}(1.14) \\ 0.23\,\mathrm{sinc}(1.37) \\ 0.23\,\mathrm{sinc}(1.61) \end{bmatrix} = \begin{bmatrix} -0.0428 \\ -0.049 \\ -0.0273 \\ 0.0214 \\ 0.0902 \\ 0.1579 \\ 0.2105 \\ 0.23 \\ 0.2105 \\ 0.1579 \\ 0.0902 \\ 0.0214 \\ -0.0273 \\ -0.049 \\ -0.0428 \end{bmatrix} = \begin{bmatrix} -1402 \\ -1605 \\ -895 \\ 701 \\ 2956 \\ 5174 \\ 6898 \\ 7537 \\ 6898 \\ 5174 \\ 2956 \\ 701 \\ -895 \\ -1605 \\ -1402 \end{bmatrix} \cdot 2^{-15}$$

7.3 Results

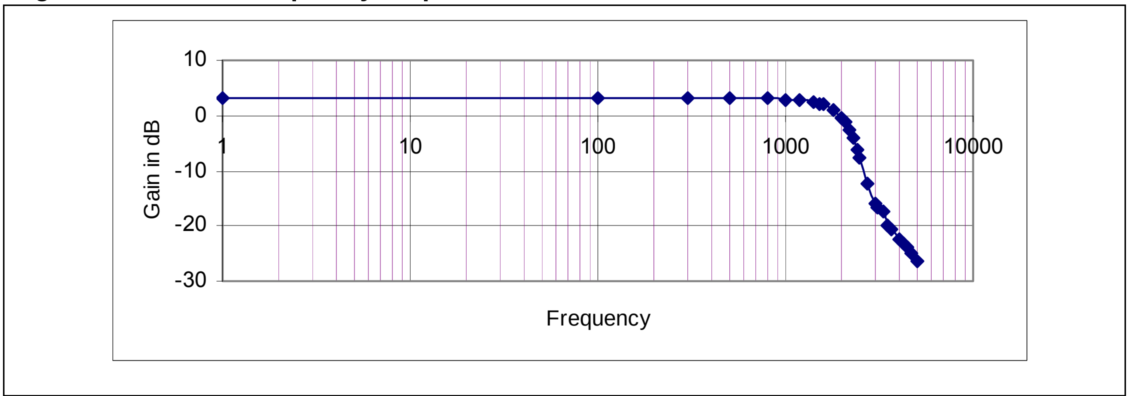
To analyze the filter’s frequency and phase responses, a sinusoidal signal of a 2V amplitude is used. The input voltage varies between 0 and 2 V.

7.3.1 Frequency response

To draw the filter’s frequency response, the amplitude of the output signal is measured.

The following figure gives the filter’s response.

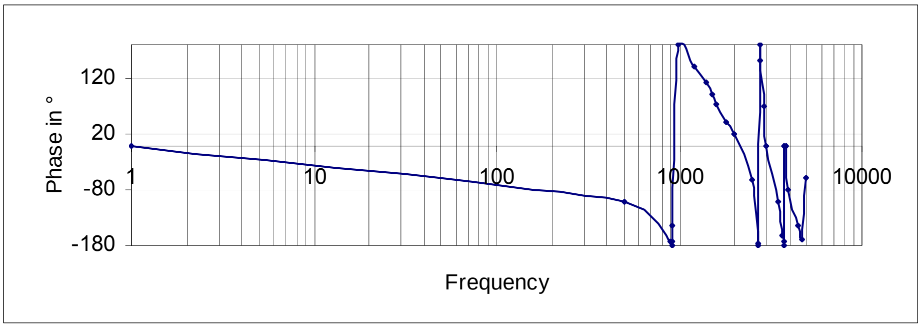
Figure 10. Filter’s frequency response



7.3.2 Phase response

The following figure gives the phase response.

Figure 11. Phase response



The code implemented to generate this digital low-pass filter can be used to implement a high-pass, a passband or a cutoff band filter. One simply needs to change the filter's coefficients and the number of coefficients. However, with this implementation, care should be taken with the sampling frequency, which changes with the filter's coefficients number and the ST10F27x frequency.

8 References

1. ST10F276 user manual
2. ST10F276 datasheet
3. AN1442 *Signal processing with ST10-DSP*, application note

9 Revision history

Table 2. Document revision history

Date	Revision	Changes
19-Jul-2007	1	Initial release.