Application Note, V1.2, Jul. 2002

AP32053

Real-time FIR Filter with the TriCore TC1775

32–Bit Microcontroller Product

Microcontrollers

N e v e r   s t o p   t h i n k i n g .

Real-Time FIR Filter with the TriCore TC1775

Revision History:	2002-07	V1.2
Previous Version:	V1.1, 2002-06	

Page	Subjects (major changes since last revision)
11	FIR filter equation corrected
16	filter order corrected: filter of **third** order reaches 60 dB/ decade

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing at all? Your feedback will help us to continuously improve the quality of this document. Please send your proposal (including a reference to this document) to:

mcdocu.comments@infineon.com

Edition 2002-07

Published by
Infineon Technologies AG
81726 München, Germany

LEGAL DISCLAIMER

THE INFORMATION GIVEN IN THIS APPLICATION NOTE IS GIVEN AS A HINT FOR THE IMPLEMENTATION OF THE INFINEON TECHNOLOGIES COMPONENT ONLY AND SHALL NOT BE REGARDED AS ANY DESCRIPTION OR WARRANTY OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE INFINEON TECHNOLOGIES COMPONENT. THE RECIPIENT OF THIS APPLICATION NOTE MUST VERIFY ANY FUNCTION DESCRIBED HEREIN IN THE REAL APPLICATION. INFINEON TECHNOLOGIES HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND (INCLUDING WITHOUT LIMITATION WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY) WITH RESPECT TO ANY AND ALL INFORMATION GIVEN IN THIS APPLICATION NOTE.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

1 Introduction 4

2 AD-Conversion 6
2.1 Analog Input-Low-Pass 6
2.2 Analog-to-Digital Converter 6
2.3 Level of the Input Signal 7
2.4 Oversampling 8

3 FIR Filter 11
3.1 TriLib 12
3.1.1 Selection of the Filter Algorithm 12
3.2 Calculation of Coefficients with ScopeFIR 13

4 Digital-to-Analog Conversion 14
4.1 GPTA 14
4.2 Analog Output Low-Pass 15

5 Software 18
5.1 Ranges of values in the system 21

6 System test 22

7 Conclusion 25

8 Appendix 26
8.1 Used Port Pins 26
8.2 Software 26
8.3 Filter coefficients and transfer functions 27
8.3.1 Low-pass 28
8.3.2 High-pass 29
8.3.3 Band-pass 30
8.3.4 Band-stop 31

9 Literature 32

1 Introduction

This application note describes the realisation of a real-time FIR filter with the TC1775 TriBoard. The TC1775 with its OnChip peripherals (AD-converter and GPTA) and DSP functionality fits very well to this task.

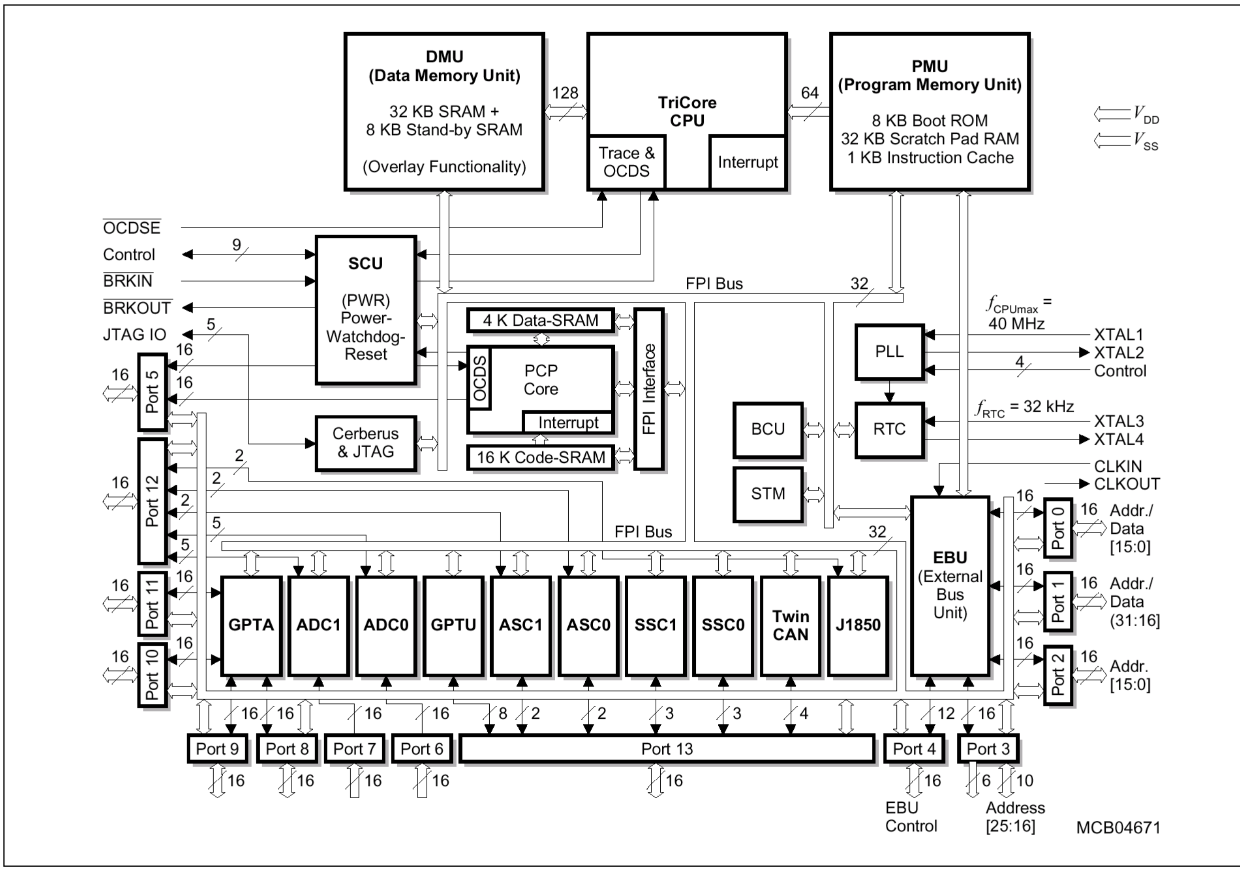

Figure 1: Block Diagram TC1775

The filter-functions high-pass, low-pass, band-pass and band-stop are realised with a FIR filter of 32. order each.

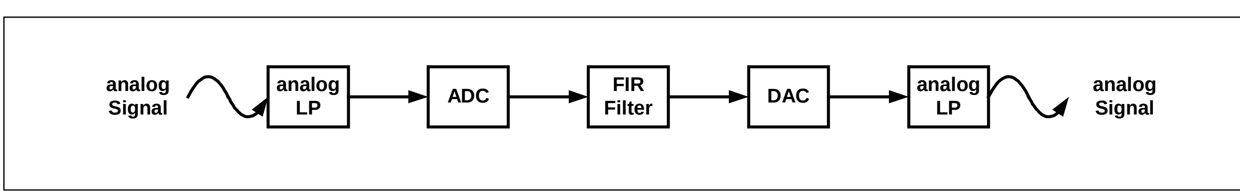

Figure 2: System Overview

The analog input-signal (e.g. from a signal-generator) is digitised with one of TC1775's two Analog-to-Digital converters.

The calculation of the digital filter function (DSP) is done by the CPU. For that a FIR algorithm of Infineon's TriLib is used. The coefficients for the filter can be calculated with an appropriate software program. For this application note the program "ScopeFIR" was used.
Finally the GPTA of the TC1775 generates a PWM signal with the digital filtered signal. To get an analog signal a low-pass has to be connected.

2 AD-Conversion

In this application a resolution of 12-bit and a sample rate of 10 kHz is used for the conversion. According to the sampling theorem a signal up to 5 kHz can be converted. This is sufficient to demonstrate and test the different filter functions.
In the next chapter some hints concerning the digitalisation are given which have to be considerated.

2.1 Analog Input-Low-Pass

The analog signal fed in the system mustn't exceed the frequency $f_{max} = f_s / 2$ where f_s is the sampling frequency. Otherwise the signal will be distorted by the so called aliasing.
To avoid this a low-pass has to be connected before the ADC, so the spectrum of the input-signal is reduced to the correct bandwidth. This low-pass has to be designed according to the system requirements. In this application a passive low-pass of first order is used. That's sufficient because only signals up to 5 kHz are fed in by the signal generator. Some hints how to design this low-pass can be found in the chapter DA-Conversion.

2.2 Analog-to-Digital Converter

The TC1775 has 2 AD-Converters with 16 channels each and a resolution up to 12-bit. The conversion time is 8.1 µs at 12-bit resolution and 7.1 µs at 10-bit resolution.
For each AD-Converter several Conversion Request Sources are available which can request the conversion of a dedicated channel.

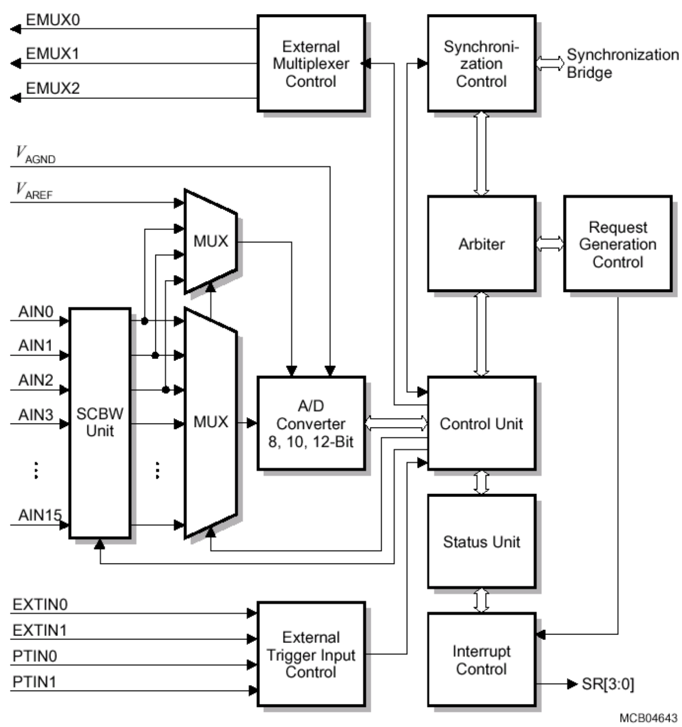

Figure 3: Block Diagram of ADC Kernel

To digitise an analog signal without distortion it is necessary that the time between two conversions is always constant (equidistant). With the TC1775 this can be realised by using the integrated timer working as Conversion Request Source for the ADC. After initialising the ADC and the timer it is guaranteed that always after a defined time a conversion is done. With this time the sampling frequency f_S is determined. In this application f_S is 10 kHz; this means every 100 µs a conversion is started.
To avoid that a timer triggered conversion is delayed because of a currently running other conversion the start of a conversion can be locked a short time before the timer triggered conversion will start by using an Arbitration Lock. The time for the Arbitration Lock has to be chosen longer than the maximum used conversion time.
Of course, this AD-channel has to have the highest priority. Otherwise equidistant AD-conversion will not be guaranteed.

2.3 Level of the Input Signal

The ADC of the TC1775 have an input voltage range of 0V to +5V. A signal source often has a symmetrical signal from $-U_{max}$ to $+U_{max}$. So it is necessary to adapt this signal to the asymmetrical ADC input. This can be done by using a potentiometer and a capacitor.

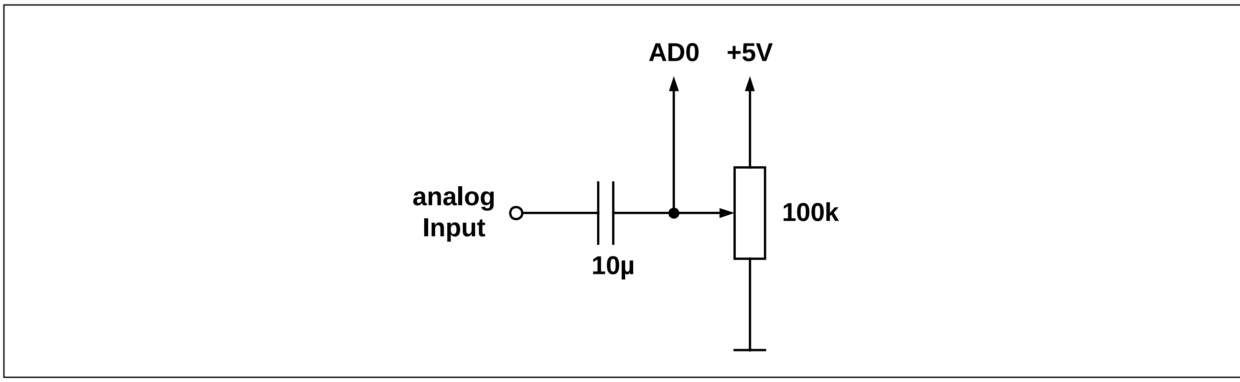

Figure 4: Input Circuit

With the potentiometer the now virtual zero point can be adjusted to 2.5 V.

2.4 Oversampling

If the frequency of the input signal increases towards the Nyquist frequency (f_s / 2) it becomes obvious that the amplitudes of the signals are adulterated and a beat occurs.

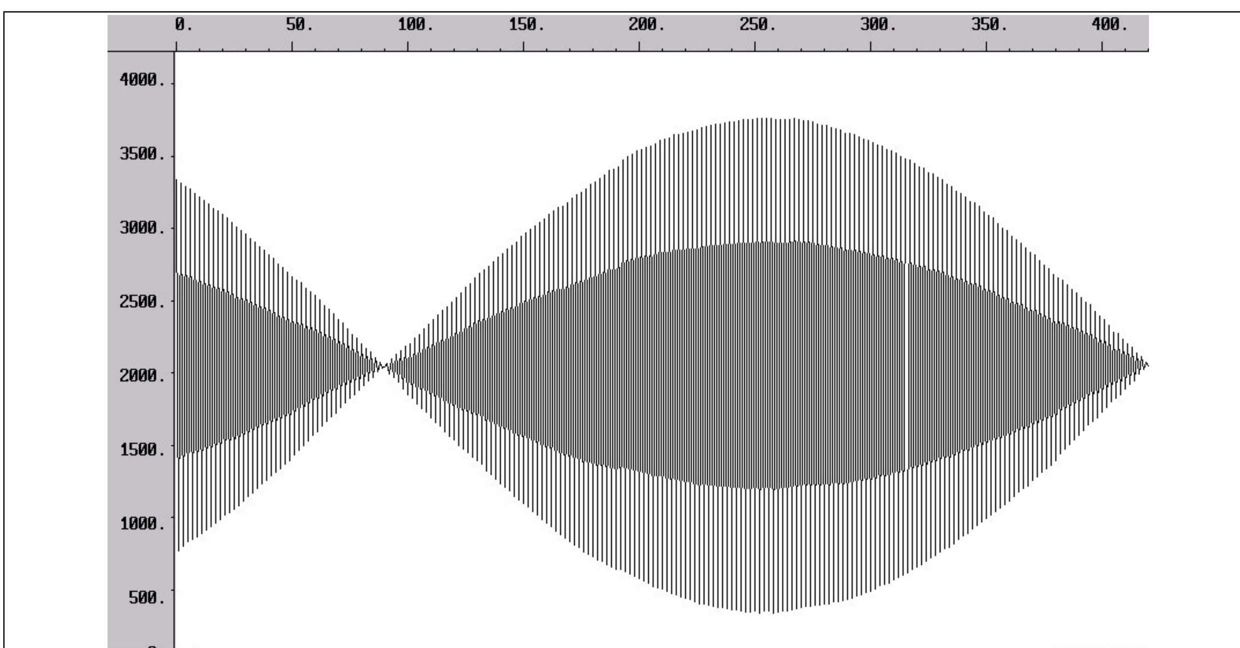

Figure 5: Digitised 5 kHz Signal with f_s = 10 kHz

This is because the sample point for the digitalisation isn't always at the maximum level of the signal but moves in the phase domain. At relative high frequencies only one value is sampled per half-cycle and so the discretization becomes obvious.

To avoid this an oversampling can be introduced, whereby the sampling frequency is not double the maximum input frequency but a factor of e.g. 10 or higher. The maximal input frequency stays at 5 kHz where the sampling frequency will now be e.g. 100 kHz. So the signal will not be adulterated in the way described above.
For that the timer triggering the ADC has to be initialised accordingly. The signal digitised this way has to be converted down to a sampling frequency of 10 kHz again by a decimating filter.

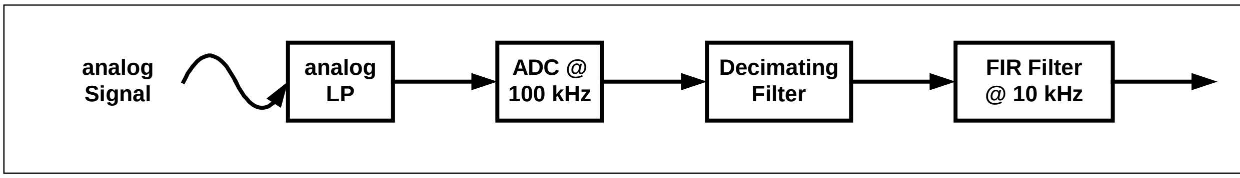

Figure 6: Oversampling and Decimation

This can be done by using a special filter algorithm or by calculating the average.

Following is an example to calculate the average where the decimating factor is 10:

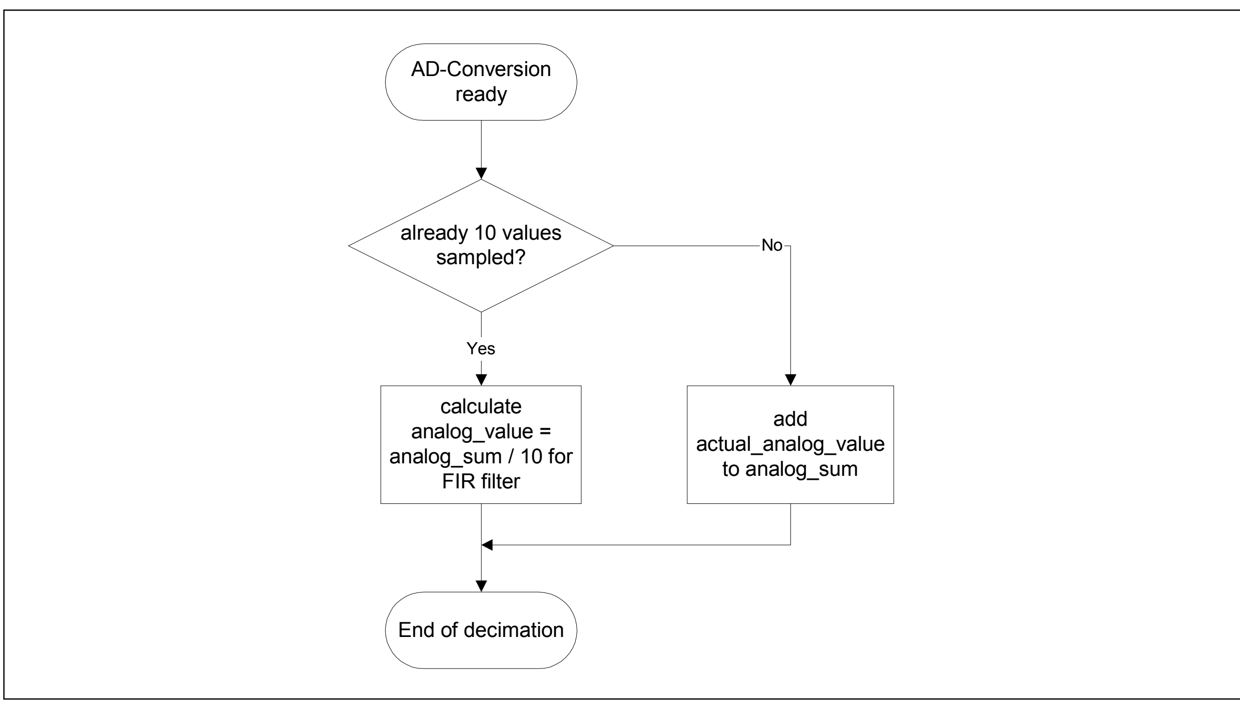

Figure 7: Flowchart of Decimation

The appropriate source code can look like this:

```
if (i<10)
{
   analog_sum = analog_sum + actual_analog_value;
   i++;
}

else
{
   analog_value = analog_sum / 10;
   analog_sum = 0;
   i=0;
   // normal FIR filter processing also in this section
}
```

By calculating the average the maximum amplitude is never reached. The signal shows a attenuation at the relative high frequencies because there the steepness of the signal is higher an so the attenuation becomes more obvious.

3 FIR Filter

Digital signals can be processed by DSP (Digital Signal Processing). There are many different algorithms which are used according to the requirements.
FIR (Finite Impulse Response) filter are suitable for different filter functions. They can be implemented easily on processors and have the advantage that they are always stable. The stability is e.g. for IIR (Infinite Impulse Response) filters not guaranteed.
For a general nH tap FIR filter the difference equation is

$$R(n) = \sum_{i=0}^{nH-1} H_i \cdot X(n-i)$$

where:

$X(n)$	:	Filter input for n^{th} sample
$R(n)$	:	Output of the filter for n^{th} sample
H_i	:	Filter coefficients
nH	:	Filter order

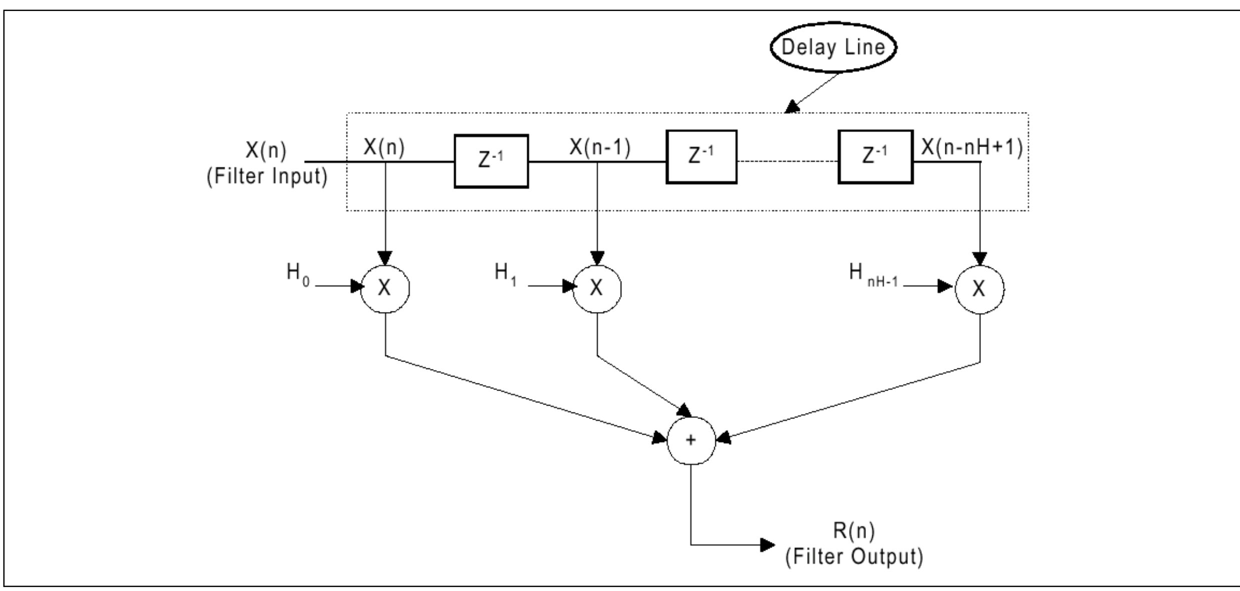

Figure 8: Block Diagram of the FIR Filter

In this application the high-pass, low-pass, band-pass and band-stop are realised using FIR filters according to the description.
These filters have an order of 31 (high-pass, band-stop) resp. 32 (low-pass, band-pass), whereby the filters of 31. order are realised in the software as filters of 32. order (H_{31} = 0), to get a runtime optimised solution. More detailed information can be found in the chapters "Selection of the Filter Algorithm" and "Calculation of Coefficients with ScopeFIR".

TriCore’s super-scalar architecture consists of three instruction pipelines, an Integer Execution Pipeline, a Load/Store Pipeline and a Zero-overhead Loop Pipeline, allowing issue of up to three instructions per clock cycle.

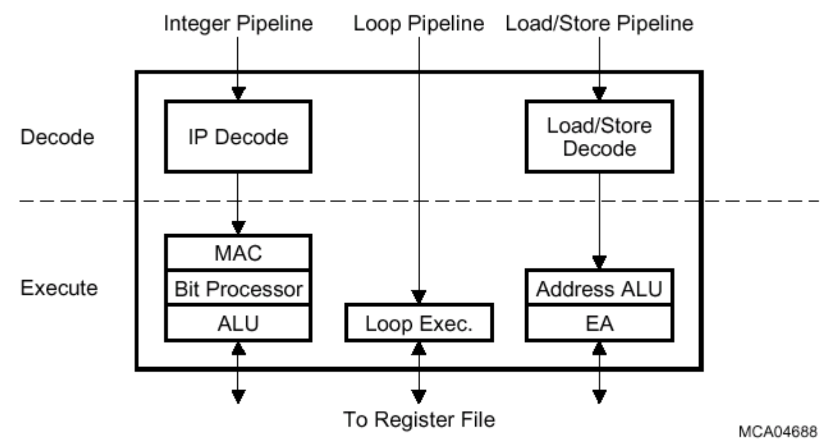

Figure 9: Pipelines of the TriCore

In addition, TriCore has SIMD (Single Instructions Multiple Data) capability, allowing operation of multiple sets of data. A typical example is TriCore’s Dual-MAC instruction, which multiplies two signed operand, adds the result to a third operand and store the result in a destination, within one clock cycle. With this Single Cycle Dual-MAC capability a FIR filter can be implemented on TriCore very efficiently, because a FIR filter can be directly done with the Multiply-Accumulate arithmetic.

3.1 TriLib

The TriLib, which is available free of charge at Infineon, is a DSP library for the TriCore, including the most commonly DSP algorithms. They are available as source code and can be integrated easily in own projects. By using this tested and runtime optimised algorithms a significant reduction of the development time can be reached. The assembly routines are optimised by hand and are callable from C/ C++.

3.1.1 Selection of the Filter Algorithm

In TriLib different FIR algorithms for different requirements are available. They differ in runtime behaviour, memory requirement or restrictions in the quantity of coefficients.
If a restriction for the quantity of the coefficients (multiple of 4) is acceptable a faster FIR algorithm can be used, which makes full use of TriCore's DualMAC capability. If the filter has no filter order with a multiple of 4 additional zero-coefficients can be inserted so that this requirement can be fulfilled. This is done for the high-pass and band-stop.

In the library the filters are also distinguished if they are symmetrical or normal (not symmetrical) filters. The advantage of a symmetrical filter is that the memory requirement is only half for the coefficients, but the performance is worse. In this application filters of 32. order are used, so the occupied memory is not relevant and a normal filter (not symmetrical) is used.
Finally the filters a distinguished by block-processing or sample-processing. For the real-time filter samples are continuously processed, so a filter with sample-processing is chosen.
The selection according to this points lead to a filter algorithm called "FIR_4_16" in TriLib. With this algorithm and the appropriate coefficients the desired signal-processing can be done.

3.2 Calculation of Coefficients with ScopeFIR

The impulse response of a FIR filter and with that the function as high-pass, low-pass, band-pass or band-stop is determined through the coefficients. There are varying PC programs available to calculate the desired filter function. For this application the program "ScopeFIR" was used.

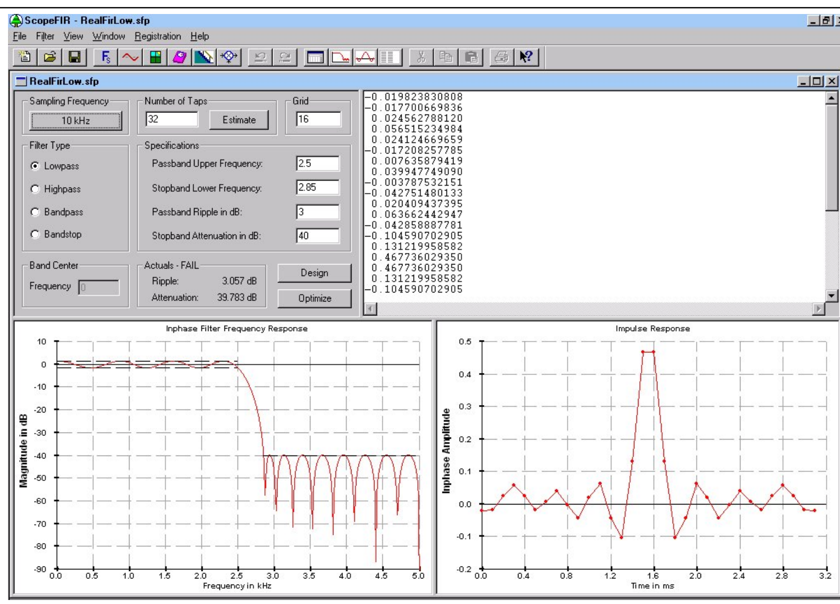

Figure 10: Screenshot ScopeFIR

To calculate the coefficients the program needs the sampling frequency, the filter type, number of taps (filter order), the cut-off-frequencies and attenuation resp. ripple. The calculation is done with the Parks-McClellan method. The coefficients can be exported and copied to the software. The used filters can be found in the appendix including the frequency response and coefficients.
A demoversion of "ScopeFIR" can be downloaded at www.iowegian.com.

4 Digital-to-Analog Conversion

The signal which was calculated with the FIR filter has to be converted from digital to an analog signal. To do this there are different possibilities: A special Digital-to-Analog Converter IC can be used which is interfaced either parallel or serial to the TC1775. Another possibility is to generate a PWM (pulse width modulation) signal with a high frequency and use a low-pass to filter it. The second method was chosen because this is less expensive and no special hardware is necessary.

4.1 GPTA

To generate a PWM signal with the TC1775 its GPTA (General Purpose Timer Array) can be used.
The GPTA is an array with a big number of timercells, which can be programmed in different modes so it is a flexible module for the generation of signals. Of course signals can also be captured and processed with special filtercells.

The generation of a simple PWM signal can be realised with the Local Timer Cells (LTC). Three LTCs has to be used; one for the timer, one for the period and one for the duty cycle.

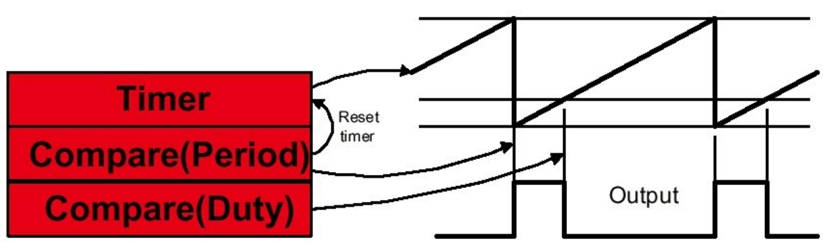

Figure 11: PWM Generation with Local Timer Cells (LTC)

A pulse-width of 100% can't be reached if the cells are programmed in this way, but that doesn't matter in this case, because this only reduces the operation band slightly what hardly will be noticed.

To produce a PWM signal with the same resolution as the signal coming from the ADC with 12 bit also 12 bit are necessary in the GPTA. At this resolution the maximum period-frequency is at about 10 kHz. It is calculated with the maximum GPTA frequency and the resolution n:

$$f_{period} = \frac{f_{GPTA}}{2^n}$$

The maximum frequency for the GPTA is 40 MHz with the TC1775. This leads to following period-frequencies of the PWM signal:

Resolution n	f_{period}	Reloadvalue
12-bit	9.77 kHz	FFFh
10-bit	39.06 kHz	3FFh
8-bit	156.25 kHz	FFh

Table 1: Resolution and PWM-Frequency of PWM Signal

Higher period-frequencies have the advantage that the requirements for the low-pass are lower. But otherwise the resolution of the signal gets lower.

With future derivatives the CPU and GPTA frequency will be higher (> 100 MHz) so better resolution at higher frequencies can be achieved.

4.2 Analog Output Low-Pass

For the analog low-pass two different solutions are possible: Either a passive or active low-pass can be used.

The design of a passive filter of first order is very simple; just a resistor and a capacity are needed.

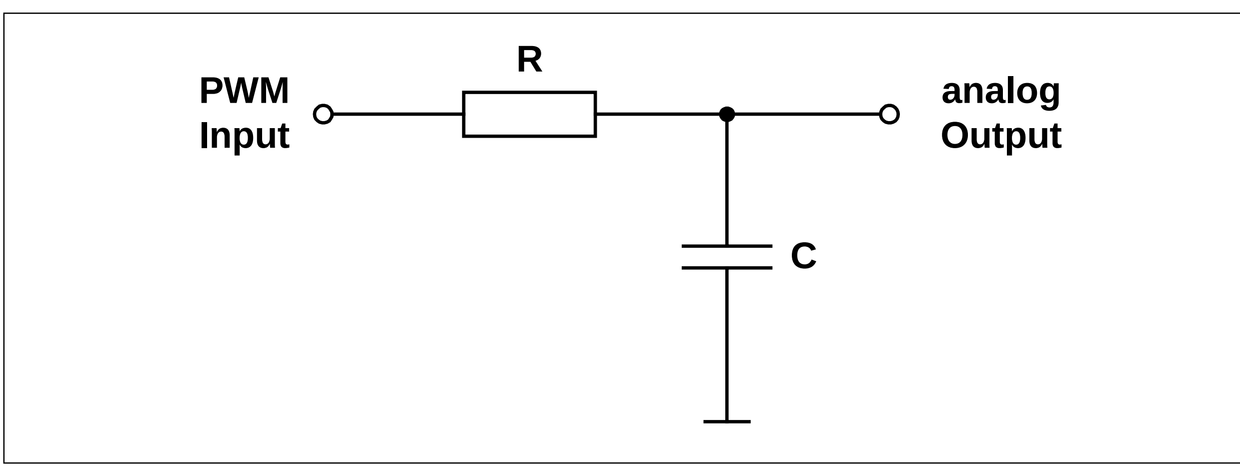

Figure 12: Low-Pass of first Order

For a low-pass of first order applies:

$$f_c = \frac{1}{2\pi RC}$$

The attenuation is 20 dB/ decade, at the cut-off-frequency the attenuation is 3 dB.

To get the values for the resistor and the capacitor one value can be chosen (e.g. the capacitor) and the other value (the resistor) can be calculated with the formula:

$$R = \frac{1}{2\pi C f_c}$$

In practice a capacitor and a potentiometer can be used so that the filter can be adjusted in a certain range.

Because a passive filter has a small steepness the useable bandwidth will be limited. To get a useable signal also with this simple filter it is a possibility to increase the period frequency as described in the previous chapter. Depending on that the cut-off-frequency has to be chosen. A resolution of 8-bit and a period-frequency of 156.25 kHz makes it possible to represent frequencies up to 5 kHz with a passive filter of first order.

A passive filter has the disadvantage that the behaviour depends on the output load. To avoid this an impedance converter has to be used. This leads to an active filter of first order.

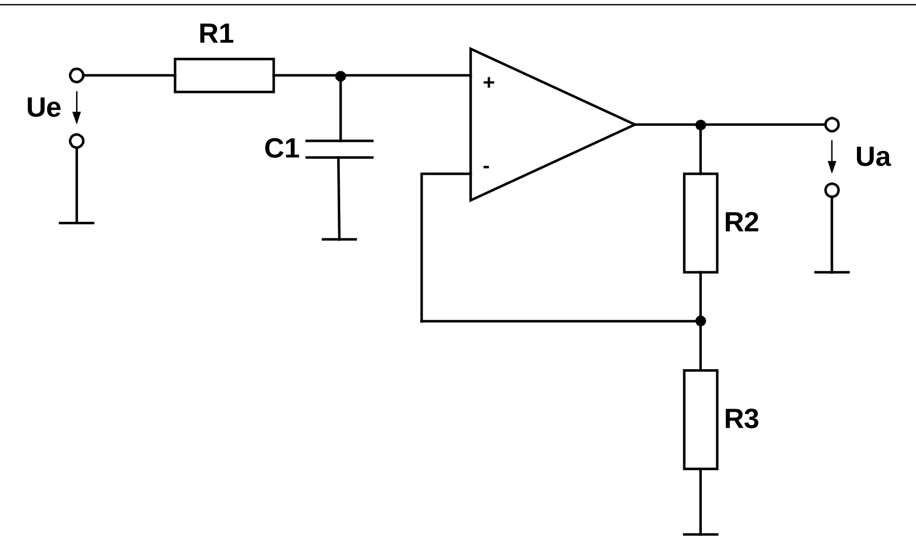

Figure 13: Low-Pass of first Order with Impedance Converter

With an active filter better results can be achieved, because with an active filter a higher steepness can be achieved. By using an active filter of second order an attenuation of 40 dB/ decade is reached, a filter of third order reaches 60 dB/ decade. The attenuation at the cut-off-frequency is still 3 dB.

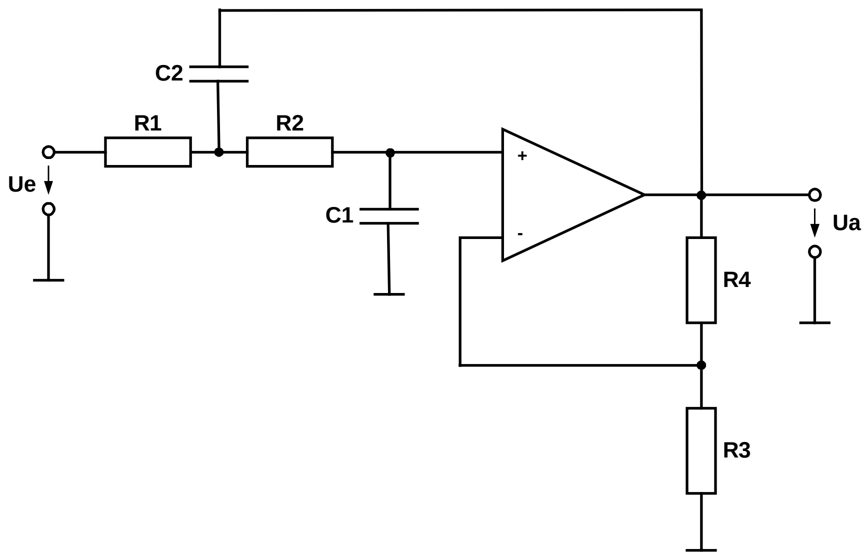

Figure 14: Active Low-Pass of second Order

The calculation of an active filter is much more complex and depends on the chosen filter type. The different filter types like Butterworth, Bessel or Tschebyscheff have different characteristics and are chosen depending on the system's requirements.
Such an active filter is more expensive to build. Some op-amps with an appropriate external circuit and power supplies are required for this solution. Because in series production this additional costs often can't be afforded no more investigation has been spent to this solution.

The same problem has to be taken into account for the input low-pass.

It may be possible to correct the attenuation forced by the analogue low-passes with a more complex FIR filter, but not with the used one with 32. order. There the ripple is already 3 dB so there is no possibility to eliminate the attenuation of the analogue low-passes.

5 Software

The software is compiled with the Tasking C/C++ Compiler V1.1 R2 for TriCore. To initialise the different peripherals the softwaretool DAvE V2.1 R22 with the update for TC1775 V2.2 from Infineon was used. With DAvE the registers can be initialised under Windows using Wizards. The C-Code generated by DAvE can directly be used in the project. Modifications done by the user in the code will be recognised and won't be overwritten. DAvE is of good use for the creation of a new project, the time for initialisation can be shortened and potential errors can be avoided.

The project is based on the files generated by DAvE and the standard TriBoard configuration. The software has following structure:

In "MAIN.C" the initialisation of the peripherals is done. After that an endless loop is executed, all further processing is interrupt driven.

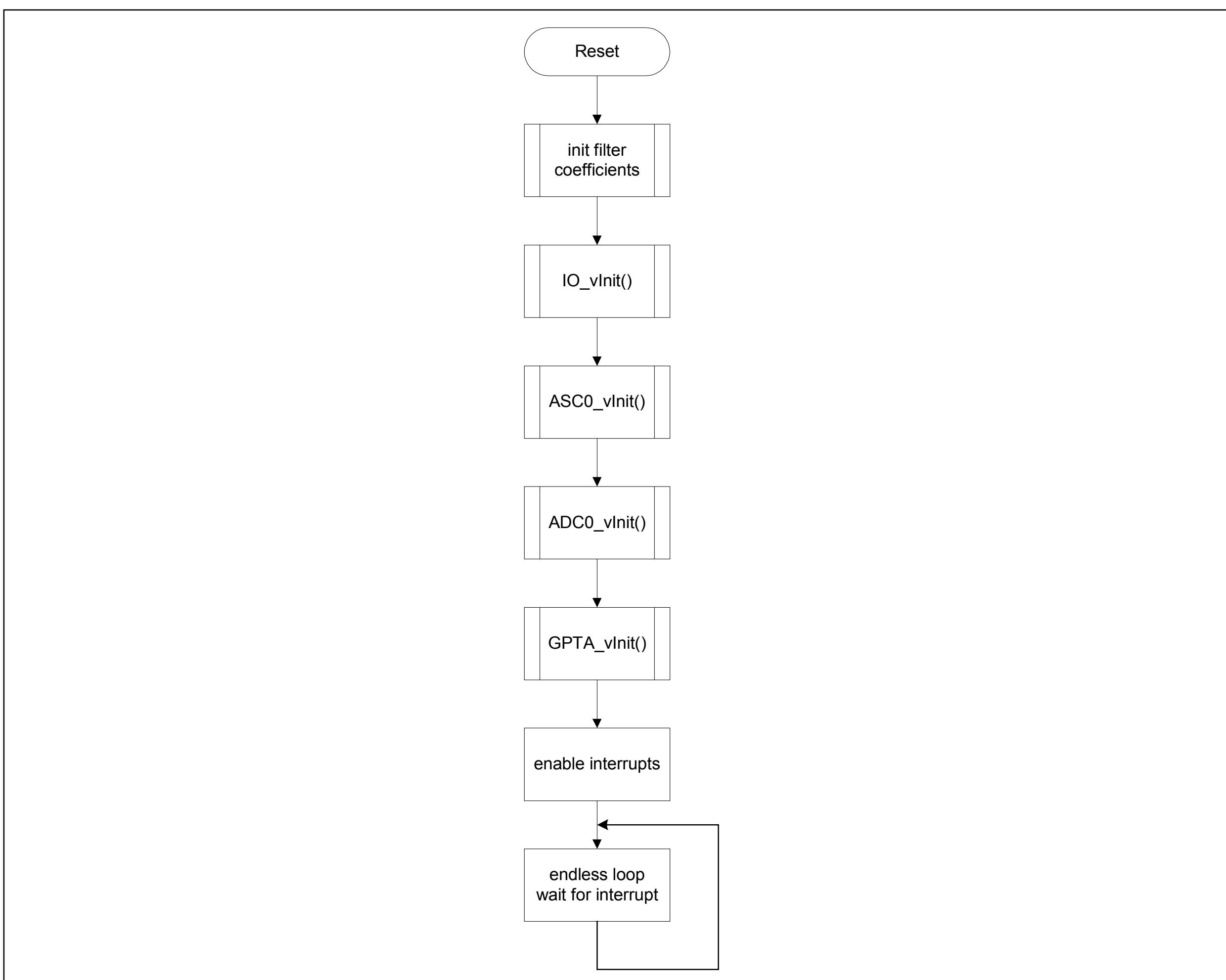

Figure 15: Flowchart "MAIN.C"

Two interrupts are activated: The ADC interrupt, showing the end of an AD conversion, and the ASC interrupt to change the filter type. The ADC interrupt is triggered every 100 µs, the ASC interrupt is triggered on user input only.

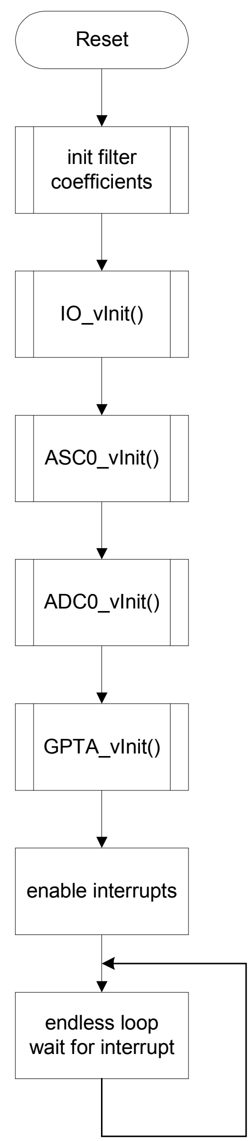

Figure 16: Flowchart ADC Interrupt

In the ADC interrupt all the calculation for the real-time filter is be done. The analog value from the ADC is converted to a fractional data type. With this data the FIR filter routine is started. After that the result is again converted so it can be used by the GPTA to generate the PWM signal.
The measured execution time can be found in chapter "System test".

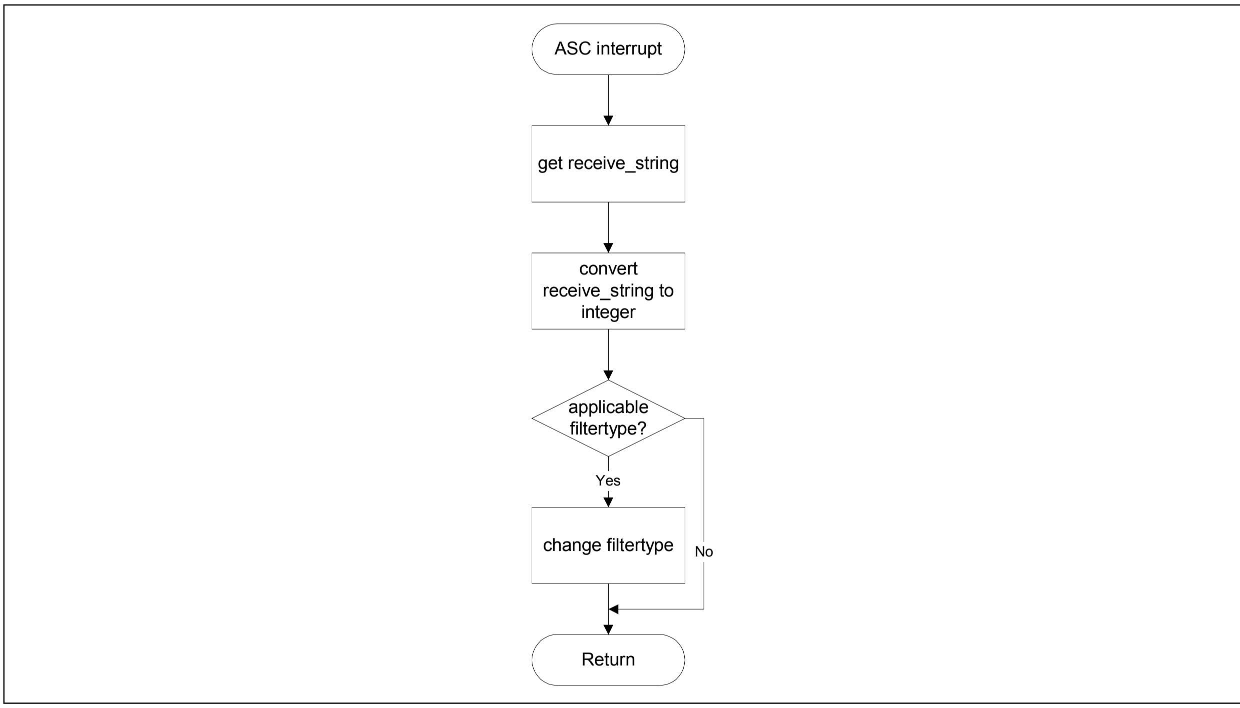

Figure 17: Flowchart ASC-Interrupt

The ASC interrupt is activated when the user sends a message to the TriCore. In the interrupt the received string is converted to an integer and evaluated if it is a applicable filter type. If this is true the filter type is changed.

After starting the software the conversion of the input signal starts immediately, the output signal is calculated and the PWM signal is generated. The filtertype used as default is the low-pass. The filter type can be changed using a terminal program like MTTTY from Microsoft. The different filter types are activated just by typing the corresponding number:

[0] Low-pass
[1] High-pass
[2] Band-pass
[3] Band-stop

The baudrate for the RS-232 has to be 9600, 8 databits, 1 stopbit and no parity has to be chosen.

To choose another period frequency for the PWM signal the code in the files "ADC0.C" and "GPTA.C" has to be adapted accordingly and the project has to be rebuilt. The initialisation for other frequencies are already inserted in the code as comments. The default period frequency is 156.25 kHz.

An extension of the number of filters or a change of the filter order is possible by modifying the "MAIN.C":

```
#define        nH        32    //Filter Order
#define        nF        4     //number of Filtertypes
```

Of course the filter coefficients has to be adapted accordingly.

5.1 Ranges of values in the system

The different systems use different ranges of values. The peripherals like the ADC and GPTA use integer data types; DSP calculation is normally done with fractional data. A short fractional data type has 1 sign bit and 15 mantissa bits.

System	Range of values	Size	Used Data Type
ADC	0 to + 4095	12-bit	Integer (signed)
FIR filter	[-1 to +1>	16-bit	_sfract
DAC 10 kHz	0 to + 4095	12-bit	Integer (signed)
DAC 40 kHz	0 to + 1023	10-bit	Integer (signed)
DAC 160 kHz	0 to + 255	8-bit	Integer (signed)

Table 2: Used Ranges of Values

For the different DAC frequencies the resolution according to Table 1 on page 15 has to be used.

Before the CPU can do the DSP calculation it is necessary to convert the integer to fractional, after the DSP calculation the fractional data has again to be converted.
It's not possible to convert an integer directly to a fractional. It is possible to do this by a multiplication of the integer with a fractional data type (int_to_fract):

```
// convert ADC value(int) to X(_sfract)
  X = value * int_to_fract;
```

To reconvert the fractional data to integer the intrinsic function mulfractlong(...) of the Tasking Compiler can be used whereby fract_to_int is an integer:

```
// convert R(_fract) to value(int)
  value = _mulfractlong(R, fract_to_int);
```

6 System test

There are different possibilities to test the system. To verify the data coming from the ADC it is possible to write them to an array. The Lauterbach Incircuit Debugger offers to display an array as graph so it is possible to evaluate the values very quick.

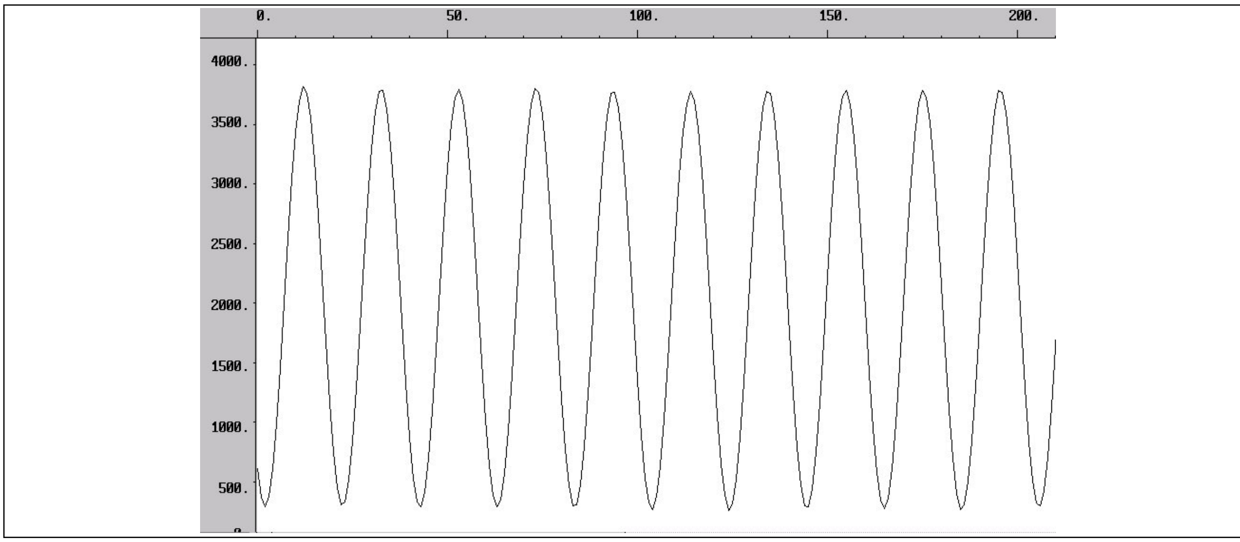

Figure 18: Digitised 500 Hz Signal monitored with an Array

A second array can be used to evaluate the data coming from the FIR filter.

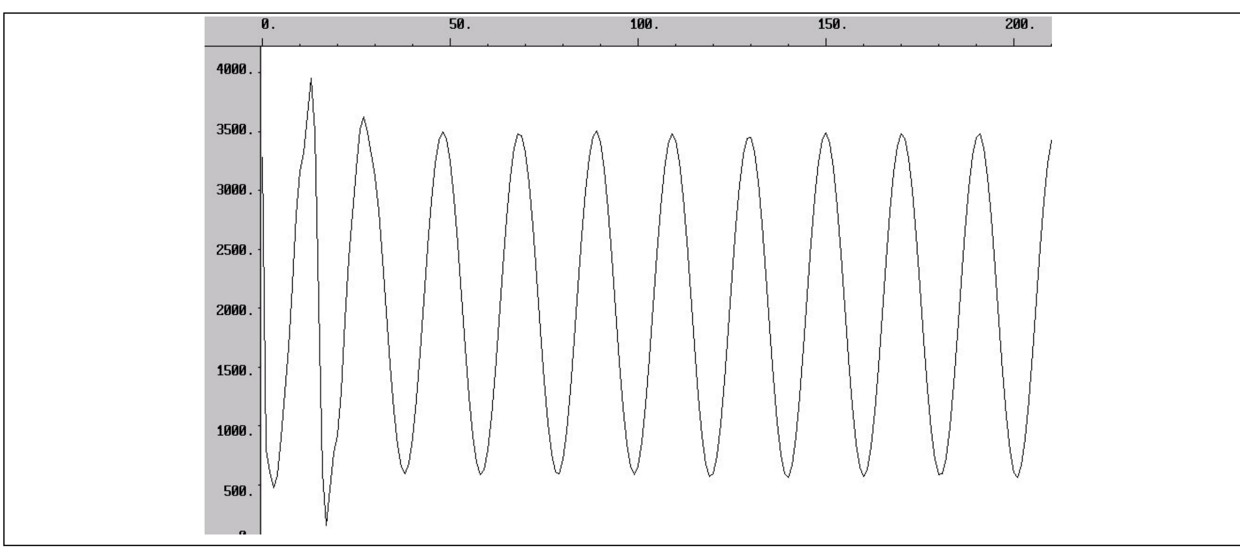

Figure 19: Data coming from FIR Filter (Low-Pass)

In the graph it can be seen that a FIR filter needs some time to get the right output. This time corresponds to the filter order.

Another possibility is of course to measure the input and output signals with an oscilloscope. So the overall system can be tested including the phase-delay of the analog low-pass filters.

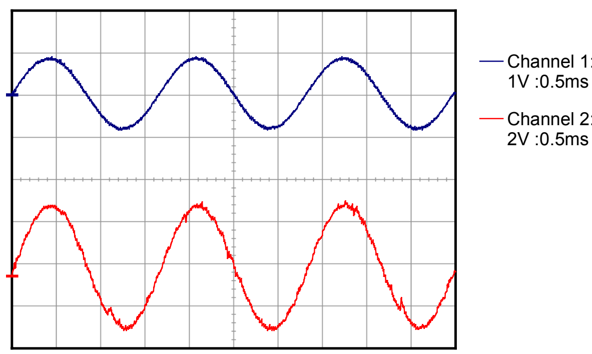

Figure 20: Oscilloscope shot of a 600 Hz signal

In this way the transfer functions of the filters have been measured. This showed that the filters meet the characteristic as defined in ScopeFIR.

The phase delay is mainly determined by the FIR filter. The two low-passes add a phase delay of maximal 90 degrees each; at the cut-off-frequency maximal 45 degrees each. The low-pass FIR filter adds a phase up to 1600 degrees!

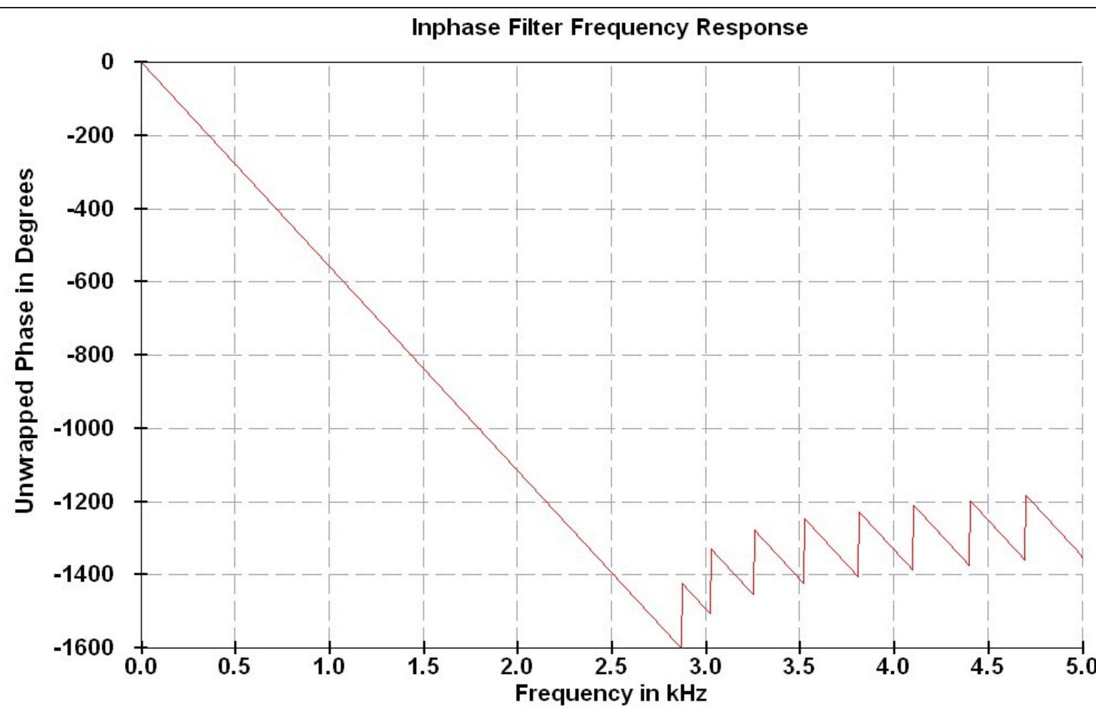

Figure 21: Phase Delay of Low-Pass FIR Filter

The high-pass has a phase delay from 500° to -900°, the band-pass from 200° to -800° and the band-stop from 0° to -1600°.

To measure the time the TriCore needs to do the filter calculation the system timer of the TC1775 can be used. The system timer is a 56-bit timer which is clocked with the CPU frequency. So the execution time can be measured with a resolution of 25 ns @ 40 MHz CPU frequency by capturing the system timer two times and calculate the difference.

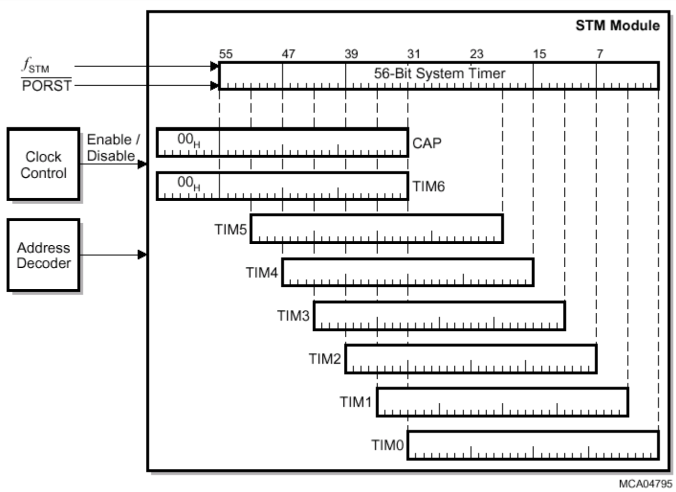

Figure 22: Block Diagram of System Timer

When the program is executed from external RAM 387 cycles respectively 9.7 µs are measured for one AD-value, respectively 32 FIR filter taps; execution from internal RAM leads to 77 cycles respectively 1.9 µs @ 40 MHz.

The system performance can be measured by toggling a port pin when entering an interrupt routine and toggle it again at the end. In this application only the ADC interrupt is relevant. The main program is just an endless loop, the ASC interrupt is only used on user request what will be very rarely.

P8.0/ GPTA0 is used in the software for the measurement. When the program is executed from internal memory the interrupt takes 3.0 µs every 100 µs. This is 3 % of the available performance @ 40 MHz.

7 Conclusion

The TriCore TC1775 offers with its DSP capability and the OnChip peripherals a powerful solution for even complex applications using control tasks with digital signal processing what is demonstrated with this application note.
Another advantage is that only one toolchain has to be used instead of two when a microprocessor and a DSP is used.
For this task a system performance of only 3 % is used. So there is enough performance for further tasks and additionally the PCP (Peripheral Control Processor) can be used offering additional computing power.
So the TriCore TC1775 fits very well to complex and time critical tasks.

Please note that this real-time FIR filter application can also be implemented on the TC1765, another powerful member of the TriCore family!

8 Appendix

8.1 Used Port Pins

Signal	Symbol	Function
Analog input	P6.0	AN0 Analog input 0
PWM output	P10.2	IN34 / OUT34 line of GPTA
ADC Int. output	P8.0	IN0 / OUT0 line of GPTA

Table 3: Used Port Pins

Remark: The ADC reference voltage and reference ground has to be connected to 5V resp. ground externally.

8.2 Software

The Software sources are included in the self extracting file (AP3253.exe):

Source files:

- Main.c — CPU initialisation, FIR routine call
- Asc.c — ASC initialisation; ASC interrupt routine
- Adc.c — ADC initialisation; ADC interrupt routine
- Gpta.c — GPTA initialisation
- Stm.c — STM initialisation
- Io.c — IO initialisation
- Fir_4_16.asm — FIR filter routine in assembly

Header files:

- Main.h
- Asc.h
- Adc.h
- Gpta.h
- Stm.h
- Io.h
- TC1775BRegs.h
- Trilib.h
- Triconv.inc

Tasking EDE files:

- RealFIR.pjt: — Project file
- RealFIR.mak: — Make file

DAvE files:

- RealFIR.dav: DAvE project file

The TriLib is available via Infineons regional sales.

The Terminal program MTTTY is available on the TriBoard Tools CD-ROM in the folder utilities.

8.3 Filter coefficients and transfer functions

For all filters a sampling frequency of 10 kHz is used. If the targeted parameters as steepness and attenuation resp. ripple can not be reached ScopeFIR gives a message and shows the reached parameters. Then a recalculation with modified parameters can be started.

8.3.1 Low-pass

Filter order:	32
Pass-band upper frequency:	2.5 kHz
Stop-band lower frequency:	3.0 kHz
Pass-band ripple:	3 dB (target) / 3 dB (reached)
Stop-band attenuation:	40 dB (target) / 39.8 dB (reached)

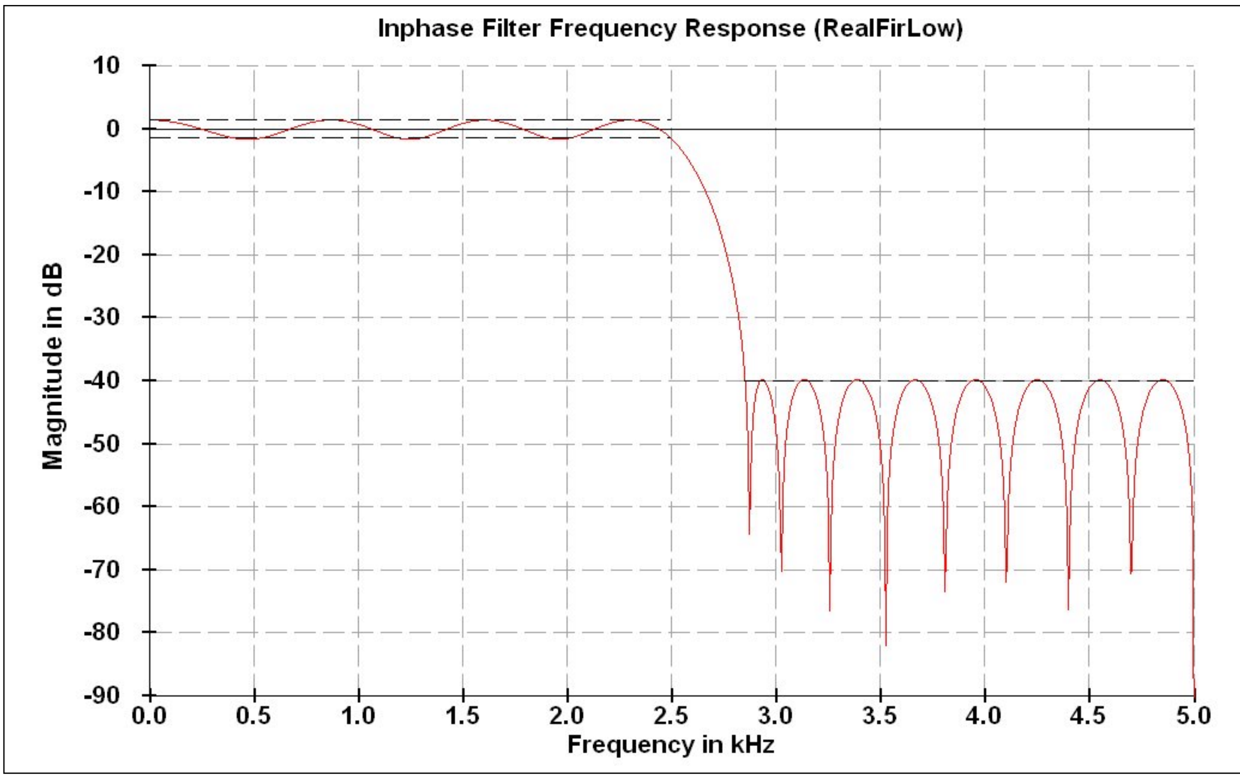

Figure 23: Inphase Filter Frequency Response of Low-pass

Coefficients H[0] ... H[31]:

-0.019823830808, -0.017700669836, 0.024562788120, 0.056515234984, 0.024124669659,
-0.017208257785, 0.007635879419, 0.039947749090, -0.003787532151, -0.042751480133,
0.020409437395, 0.063662442947, -0.042858887781, -0.104590702905, 0.131219958582,
0.467736029350, 0.467736029350, 0.131219958582, -0.104590702905, -0.042858887781,
0.063662442947, 0.020409437395, -0.042751480133, -0.003787532151, 0.039947749090,
0.007635879419, -0.017208257785, 0.024124669659, 0.056515234984, 0.024562788120,
-0.017700669836, -0.019823830808

8.3.2 High-pass

Filter order:	31
Stop-band upper frequency:	2.5 kHz
Pass-band lower frequency:	2.85 kHz
Pass-band ripple:	3 dB (target) / 3 dB (reached)
Stop-band attenuation:	40 dB (target) / 36.8 dB (reached)

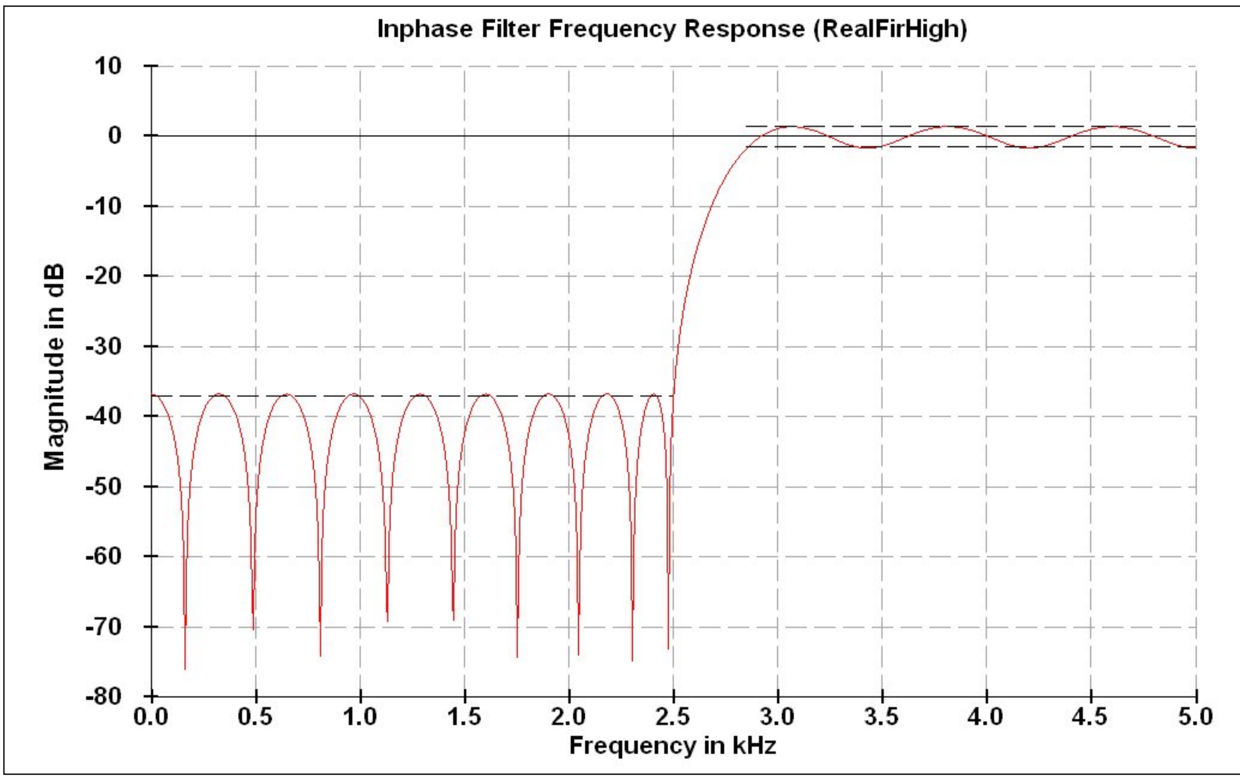

Figure 24: Inphase Filter Frequency Response of High-pass

Coefficients H[0] ... H[31]:

-0.009804315383, -0.010385248537, 0.042259644377, -0.049506892894, 0.011013822528,
0.027985710422, -0.011772500212, -0.033008029138, 0.025074365462, 0.036986648178,
-0.047669829299, -0.040627538214, 0.095908116969, 0.043102799652, -0.314852599728,
0.456128215779, -0.314852599728, 0.043102799652, 0.095908116969, -0.040627538214,
-0.047669829299, 0.036986648178, 0.025074365462, -0.033008029138, -0.011772500212,
0.027985710422, 0.011013822528, -0.049506892894, 0.042259644377, -0.010385248537,
-0.009804315383, 0

8.3.3 Band-pass

Filter order:	32
Band-pass center:	2.5 kHz
Pass-band bandwidth at top:	1 kHz
Pass-band bandwidth at bottom:	1.8 kHz
Pass-band ripple:	3 dB (target) / 3 dB (reached)
Stop-band attenuation:	40 dB (target) / 40.1 dB (reached)

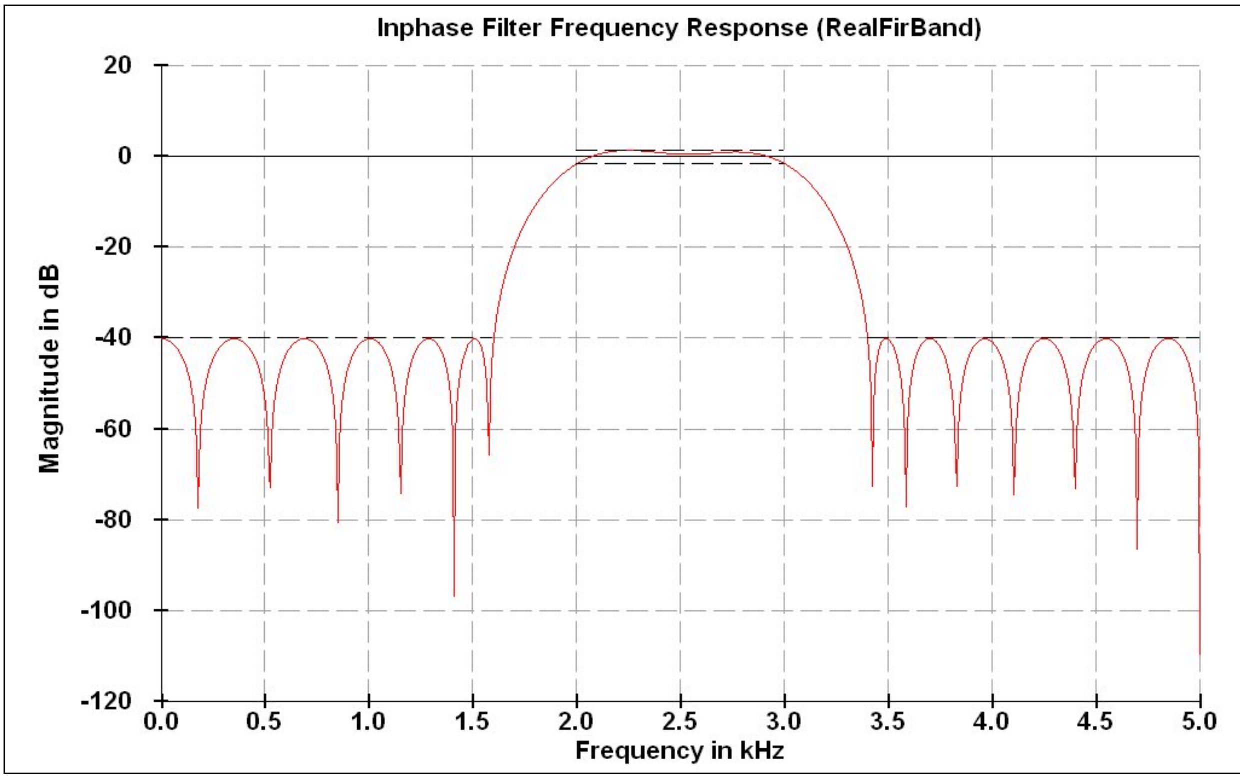

Figure 25: Inphase Filter Frequency Response of Band-pass

Coefficients H[0] ... H[31]:

-0.007724050458, 0.013007745983, 0.020557085731, -0.021066029949, -0.027525597674,
0.022485085025, 0.019803021106, -0.002863633441, 0.013029545738, -0.041645477089,
-0.068680448229, 0.102497099799, 0.131491438542, -0.159125130471, -0.177916170397,
0.188597334734, 0.188597334734, -0.177916170397, -0.159125130471, 0.131491438542,
0.102497099799, -0.068680448229, -0.041645477089, 0.013029545738, -0.002863633441,
0.019803021106, 0.022485085025, -0.027525597674, -0.021066029949, 0.020557085731,
0.013007745983, -0.007724050458

8.3.4 Band-stop

Filter order:	31
Band-stop center:	2.5 kHz
Band-stop bandwidth at top:	1.8 kHz
Band-stop bandwidth at bottom:	1 kHz
Pass-band ripple:	3 dB (Target) / 2.3 dB (reached)
Stop-band attenuation:	40 dB (Target) / 42.2 dB (reached)

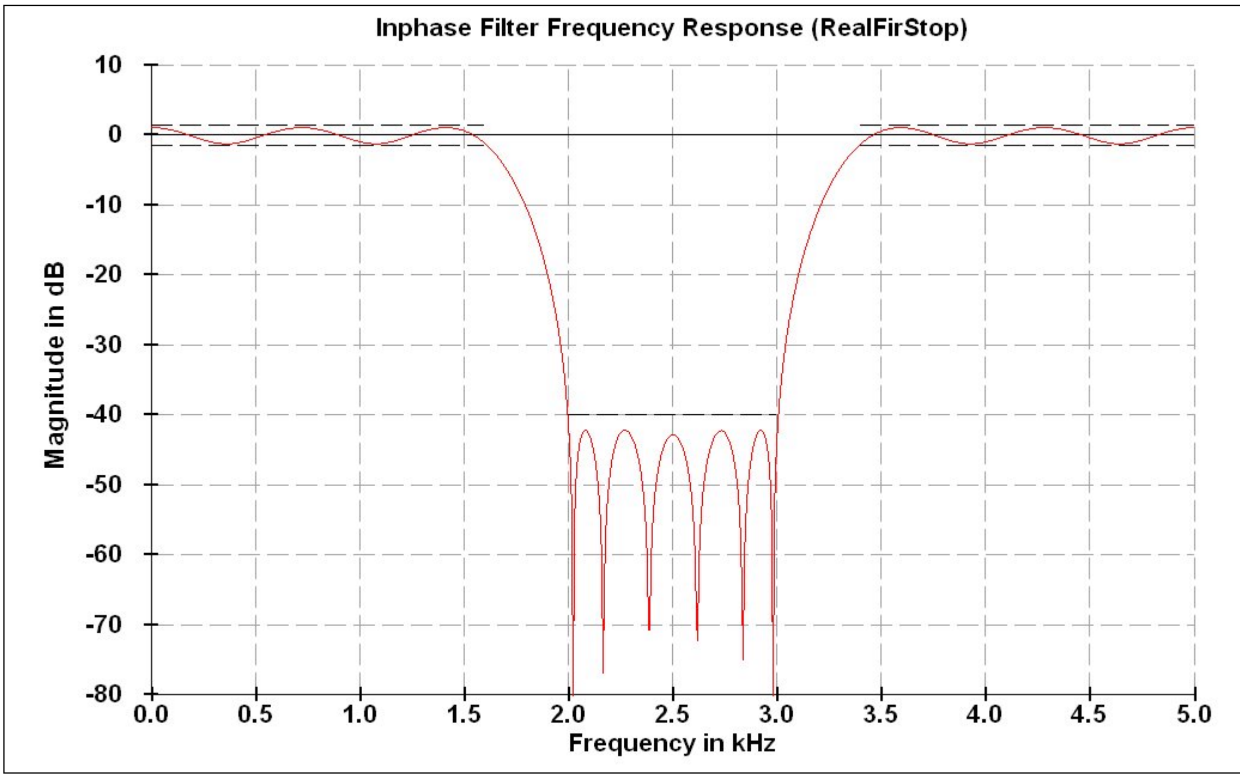

Figure 26: Inphase Filter Frequency Response of Band-stop

Coefficients H[0] ... H[31]:

-0.000009463398, 0.054457304841, -0.000000237427, 0.036093051590, 0.000001362987,
-0.055299910656, -0.000002548611, 0.042303044667, -0.000010887254, 0.029461509608,
0.000004758049, -0.146310522412, 0.000008216615, 0.256303699049, 0.000008799039,
0.698463119613, 0.000008799039, 0.256303699049, 0.000008216615, -0.146310522412,
0.000004758049, 0.029461509608, -0.000010887254, 0.042303044667, -0.000002548611,
-0.055299910656, 0.000001362987, 0.036093051590, -0.000000237427, 0.054457304841,
-0.000009463398, 0

9 Literature

1. TriLib User's Manual, V1.2, Jan. 2001
2. User's Manual TC1775, System Units + Peripheral Units; V2.0, Feb. 2001
3. Data Sheet TC1775, V1.1, Aug. 2001
4. TriBoard TC1775B Hardware Manual, V1.0, Jan. 2001
5. ScopeFIR Help, V3.7a, Copyright 1998 Iowegian International Corporation
6. Halbleiter-Schaltungstechnik, Tietze + Schenk, 10. Auflage, Springer-Verlag

Infineon goes for Business Excellence

“Business excellence means intelligent approaches and clearly defined processes, which are both constantly under review and ultimately lead to good operating results.
Better operating results and business excellence mean less idleness and wastefulness for all of us, more professional success, more accurate information, a better overview and, thereby, less frustration and more satisfaction.”

Dr. Ulrich Schumacher

http://www.infineon.com

Published by Infineon Technologies AG