AN4173
Application note

10-bit ADC - Configuration Guide for SPC56xBxx and SPC56xCxx products

Introduction

The aim of this document is to clarify the usage of the ADC 10-bit SAR peripheral timing parameters in order to help the user to find the best configuration to optimize the conversion time and the quality/precision of the sampled signal considering its input impedance.

This document describes an application able to acquire sinusoidal signals of up to 500 kHz using two DMA channels triggered by PIT and EOC signal.

The quality of the signal is estimated using an offline computation of the FFT of the sampled values.

Contents

List of tables

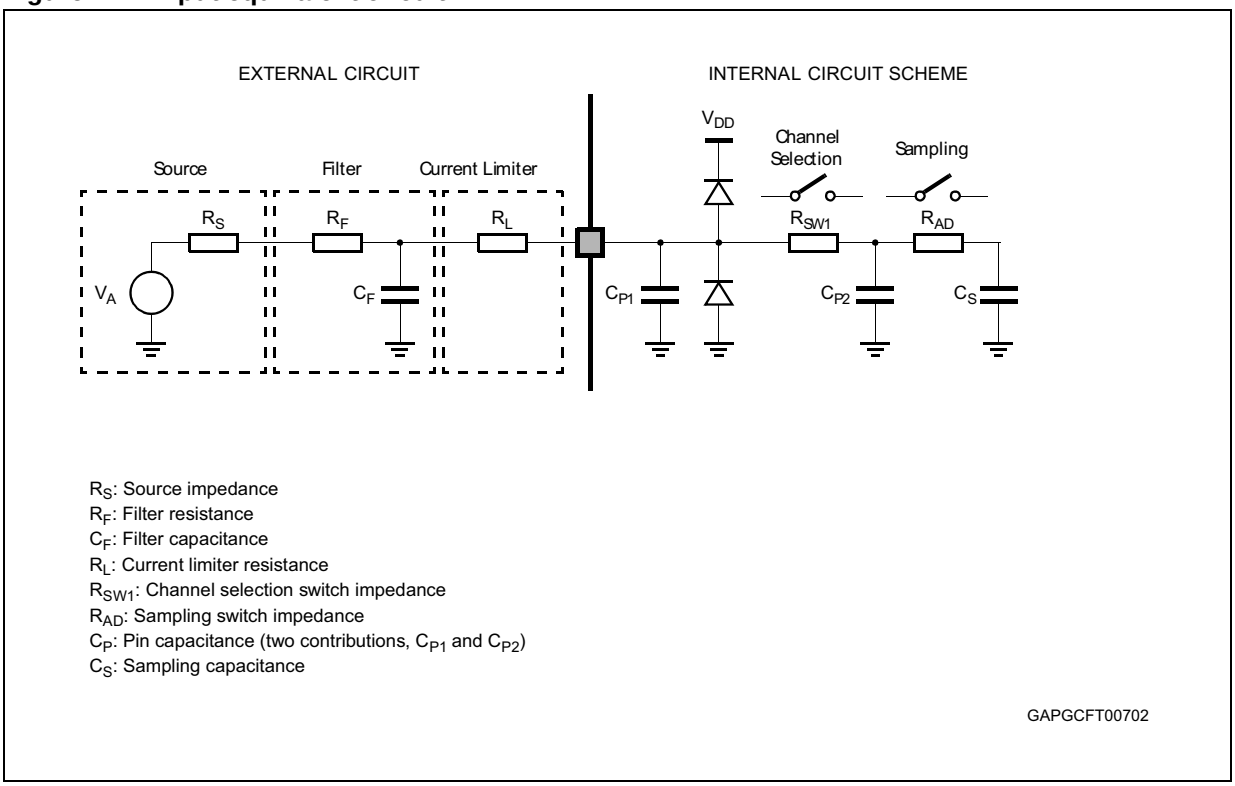
List of figures

1 Considerations on the input impedance of the signal source

The impedance relative to the signal source can limit the ADC's sample rate. Furthermore a current limiter resistance and an RC filter are often necessary to minimize the current request and to attenuate the noise present on the input pin. This external network can generate accuracy problems for the ADC converter and for this reason it is important to invest time in reaching the right adaptation.

The capacitance C_S associated with the internal ADC circuitry, combined with the resistance to the input signal source R_S creates a low-pass filter. This RC time constant can increase the time needed to stabilize the incoming signal before the ADC can sample it, and if the input signal has a high frequency, which means that it changes quickly over time, this low-pass filter can distort the ADC's acquisition.

Figure 1. Input equivalent circuit



Moreover, considering a sinusoidal signal as input, if the source has a low impedance, the RC time constant will not be relevant and the voltage on the capacitance will quickly match the input voltage.

Furthermore, all the other internal ADC components, such as the pin capacitance C_{PX}, the sampling, and the channel selection switch impedance, can limit the maximum frequency applicable to the ADC input.

The internal resistance of the source must allow the capacitance to reach its final value before the end of sample time.

If this does not happen, that is, if the source resistance is mis-matched to the sample time, a voltage loss will occur at the sample and hold stage. This voltage loss causes an accuracy loss when increasing or decreasing the input voltage from 0 to V_{refH}.

The error is calculated by the formula:

Equation 1

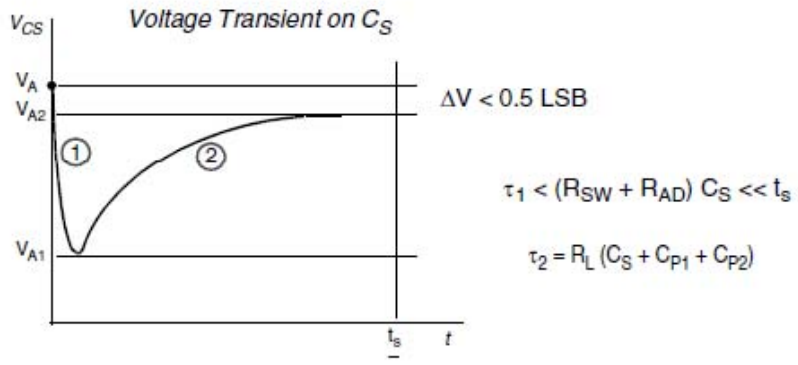
$$Maxerror_{LSB} = 1024 \times e^{\left(-\frac{TS}{RC}\right)}$$

where: t_S = sample time in µs, R = R_{SOURCE} + R_L in Ω, C = C_S in µF.

For example:

Since the error is proportional to the difference between V_{IN} and V_{refH}, the effect produces a non-linearity in the conversion of large-amplitude signals. In practice, if t_S > 7RC, the maximum error is reduced to less than half LSB (< 0.05%).

Figure 2. Transient behavior during sampling phase



A possible workaround in order to avoid the distortion is compensate the high source resistance adjusts the ADC's sample rate. A slower sample rate can provide more time for the ADC acquisition to overcome the distortion.

To minimize the error induced by voltage partitioning betweeen the equivalent resistance $R_{EQ,}$ (sampled voltage on C_S + C_{P2}) and the sum of R_S and R_F, the external circuit must be designed to satisfy the following relation:

Equation 2

$$v_A \times \left(\frac{R_S + R_F}{R_{EQ}}\right) < \frac{1}{2} \times LSB$$

where R_{EQ} = 1/($f_C \times (C_S + C_{P2})$) and f_C is the conversion rate.

Consider:

Equation 3

$$T_S \gg C_S \times (R_{SW} + R_{AD})$$

where t_S = sample time,

and

Equation 4

$$T_S > 8.5 \times R_L \times (C_s + C_{p1} + C_{p2})$$

where R_L shall be sized also according to the current limitation constraints in combination with R_S and R_F. See Data Sheet for further details.

2 ADC 10-bit timing configuration parameters

In order to control the conversion time of the ADC_0 10-bit SAR peripheral, there are three registers CTR0, CTR1, and CTR2.

The CTR0 register is associated with internal precision channels (from 0 to 15).

The CTR1 register is associated with internal standard channels (from 32 to 63).

The CTR2 register is associated with external multiplexed channels (from 64 to 95).

The three parameters involved in the evaluation of the conversion time are:

- CTR.INPSAMP: used to configure the duration of the analog input signal sampling operation. It represents the number of the clock cycles that this operation will last. Minimum value is 3. If it is less, it is automatically set to 3 inside the ADC.
  - Range of values: [3:255]
- CTR.INPLATCH: used to configure the duration of the latching phase during the 1-bit evaluation of the SAR algorithm. 0 means half a clock cycle and 1 means one clock cycle. The 1 condition is possible only if INPCMP is bigger than one. Otherwise it is automatically set to 0 inside the ADC. Parameter sets the portion of $T_{BITEVAL}$ used to latch the comparator result. If 0, this time is equal to half the period of ADC clk. If 1, it is equal to 1 TCK.
  - Range of values: [0:1]
- CTR.INPCMP: used to configure the duration of the evaluation of each bit of the SAR algorithm. It represents the number of clock cycles that this operation will last. Minimum value is 1. If it is less, it is automatically set to 1 inside the ADC.
  - Range of values: [1:3]

ADC Clk: ADC peripheral input clock frequency from 6 to 32 MHz, depending on the parameters and constraints.

2.1 Conversion timing

When a conversion is started, the ADC connects the internal sampling capacitor to the respective analog input pin, allowing the capacitance to charge up to the input voltage value. The time to load the capacitor is referred to as sampling time. After completion of the sampling phase, the evaluation phase starts and all the bits corresponding to the resolution of the ADC are estimated to provide the conversion result.

The conversion times are programmed via the bit fields of the CTR. Bit fields INPLATCH, INPCMP and INPSAMPLE are used to define the total conversion duration (t_{CONV}) and in particular the partition between sampling phase duration (t_{SAMPLE}) and total evaluation phase duration (t_{EVAL}).

The sampling phase duration of ADC is given by:

Equation 5

$$T_{sample} = (INPSAMPLE - ndelay) \times T_{CK}$$

where INPSAMPLE $\geq$ 3 and ndelay = 0.5 if INPSAMPLE < 6, otherwise ndelay = 1

The total evaluation phase duration is given by:

Equation 6

$$T_{eval} = 10 \times T_{biteval} = 10 \times (INPCMP \times T_{CK})$$

where INPCMP ≥ 1 and INPLATCH < INPCMP

The total conversion duration of ADC is given by:

Equation 7

$$T_{conv} = (T_{sample} + T_{eval} + (ndelay \times T_{CK}))$$

where TCK depends on the peripheral set 3 clock divider.

ADC Clock Frequency:

In SPC560Cxx, SPC560Dxx and SPC560Bxx devices (256 KB, 512 KB and 1.5 MB) there is a bit in the ADC MCR register that allows to select if the ADC clock frequency is half or equal to the Peripheral Set Clock frequency. The bitfield is the ADCLKSEL.

In SPC564Bxx e SPC56ECxx devices the ADC clock frequency is always set as half of the Peripheral Set Clock.

2.2 SPC56xBxx and SPC56xCxx family ADC 10-bit timing properties

The integration of the ADC 10-bit module in the SPC56xBxx and SPC56xCxx products leads to some further constraints to guarantee the performance indicated in the datasheet/electrical characteristics. In particular sampling time and conversion times must respect minimum values, which are:

- sampling time must be equal or greater than 500 ns.
- the ADC clock range is between 6 MHz to 32 MHz.
- evaluation time must be equal or greater than 500 ns.
- the minimum conversion time reachable is 1.05 us.

Table 1 gives the ADC sampling and conversion timing, at 5V / 3.3 V, for the ADC, using the suggested parameters:

Table 1. ADC sampling and conversion timing

Clock (MHz)	TCK (µs)	INPSAMP	IMPCMP	INPLATCH	ndelay	t_{SAMPLE} (µs)	t_{EVAL} (µs)	t_{CONV} (µs)	t_{CONV}/ TCK	t_{SAMPLE}/ TCK
6	0.167	4	1	0	0.5	0.583	1.667	2.333	14	3.500
7	0.143	4	1	0	0.5	0.500	1.429	2.000	14	3.500
8	0.125	5	1	0	0.5	0.563	1.250	1.875	15	4.500

Table 1. ADC sampling and conversion timing (continued)

Clock (MHz)	TCK (µs)	INPSAMP	IMPCMP	INPLATCH	ndelay	t_{SAMPLE} (µs)	t_{EVAL} (µs)	t_{CONV} (µs)	t_{CONV}/ TCK	t_{SAMPLE}/ TCK
9	0.111	5	1	0	0.5	0.500	1.111	1.666	15	4.500
16	0.063	9	1	0	1	0.500	0.625	1.188	19	8.000
20	0.050	11	1	0	1	0.500	0.500	1.05	21	10.000
32	0.031	17	2	1	1	0.500	0.625	1.156	37	16.000

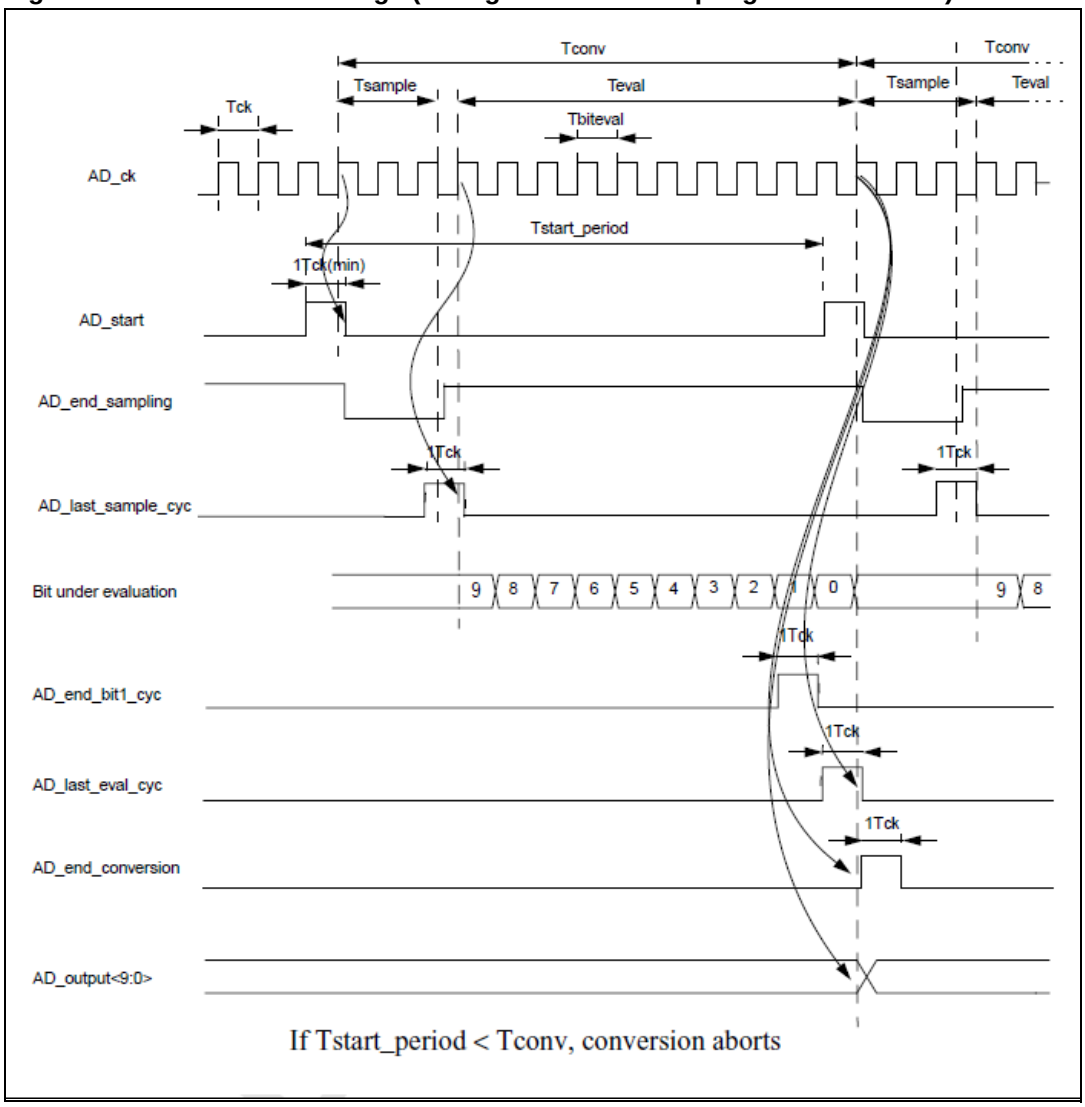
where

- INPSAMPLE ≥ 3
- ndelay = 0.5, if INPSAMPLE ≤ 6, otherwise ndelay = 1
- t_{SAMPLE} = (INPSAMP-ndelay)*TCK $\geq$ 500 ns

2.3 Latency of the ADC conversion.

The *Figure 3* shows the timing diagram in order to measure the time needed to complete successfully an ADC conversion from the begin to the end of conversion, considering every Tck needed for the sampling time (Tsample), for the evaluation time (Teval equal to 10 times of Tck for 10 bits) and the delay of 0.5 or 1 Tck.

Figure 3. Conversion timings (configuration for sampling rate at 20 MHz)



As described in the previous paragraph, the Tconv is the sum of the Tsample and Teval plus 0.5 or 1 Tck.

On the last AD_ck cycle of evaluation phase the signal AD_last_eval_cyc has a pulse of ‘1’.After the 10th bit evaluation is ended up, a pulse of ‘1’ on AD_end_of_conversion signal is generated from rising edge to rising edge of AD_ck to provide external logic the acknowledgement the conversion has completed and the AD_output bus contains the converted value.

3 Application code structure

This section gives an overview about the application code implemented to test the best configuration, in terms of conversion time and conversion precision, for the input signal used.

3.1 Peripherals used

- Device: SPC564xB/C.
- System clock: 120 MHz by PLL initialization.
- ADC frequency: 20 MHz.
- PIT configuration: a trigger is arisen every 10 µs. This means a sample time of 100 kHz.
- DMA and DMAMUX configuration: two channels used.
- STM configuration: to allow calculation of the t_{CONV} of the ADC on the device.
- Sinusoidal signal at 10 kHz with an amplitude of 4.6 V and offset of 2.3 V.

3.2 Code implementation

The application configures the device to:

- Disable the Software Watchdog
- Configure modes and activate all clocks for all peripherals
- Enable the Cache
- Enable all exceptions and the ADC_EOC, PIT and STM interrupts
- Set the GPIO in order to turn on some LEDs
- Configure the STM peripheral to calculate the ADC t_{CONV} performances.
- Configure the PLL to work at 120 MHz.
- Configure the ADC sample parameters setting, the sampling duration, the comparison duration and the latching phase time. The ADC is set to One Shot Mode with DMA feature enabled.
- Configure the PIT to create a periodic interrupt every 10 µs for a sampling time of 100 kHz.
- Initialize the DMA and DMAMUX :
  - Channel0: to move the ADC sampled values from the ADC ch0 to a buffer of 1024 elements, triggered by the EOC from DMAMUX channel 29
  - Channel1: to start the ADC acquisition from the ADC ch0, triggered by the PIT with DMAMUX channel 60 (always enabled).

3.3 Performance comparisons

A brief performance comparison has been made using a constant value in order to measure the conversion time of the ADC 10-bit on board.

Table 2. ADC sampling and conversion timing obtained by test code

Clock (MHz)	TCK (µs)	INPSAMP	IMPCMP	INPLATCH	ndelay	t_{CONV} (µs)
6	0.167	4	1	0	0.5	2.53
7	0.143	4	1	0	0.5	2.10
8	0.125	5	1	0	0.5	2.03
9	0.111	5	1	0	0.5	1.81
16	0.063	9	1	0	1	1.18
20	0.050	11	1	0	1	1.175
32	0.031	17	2	1	1	1.156

The discrepancy between the values on the RM and those from the experimental results is due to an effective phase shift of half AD_ck between peripheral clock and AD_ck. We must add 1.5 AD_ck clock cycles to the calculated t_{CONV}. This phase shift comes from the internal divider implemented within ADCDig. Of this 1.5 cycles, 1 is for storing the data and 0.5 due to phase shift.

In order to measure the time needed for the ADC conversion the system timer module (STM) was used. The STM clock frequency is set equal to the system clock frequency.

Figure 4. STM configuration function extracted code

```
/*****************************************************************
* FUNCTION : STM_Init
* DESCRIPTION : It initializes STM module
* INPUTS :
* OUTPUTS
*****************************************************************/
void STM_Init(void)
{
    STM.CR.B.CPS = 0;     //divide system clock by 1
    STM.CR.B.FRZ = 1;     //STM counter is stopped in debug mode

    STM_ResetCounter();

    /* enable compare of all channels */
    STM_ChannelDisable(STM_ALL);

    /* interrupt flag clear */
    STM.CH[0].CIR.B.CIF = 1;
    STM.CH[1].CIR.B.CIF = 1;

    STM_StartCounter();
}
```

The system clock is set equal to the PLL frequency by the PLL init function. This function permits to configure the PLL in order to switch from 40MHz of XOSC up to 120MHz.

The ADC initialization function allows to set different timing parameters for different tests.

Figure 5. ADC init function

```
/******** ADC_Init  *********/
//!  ADC initialization
/*!
*/
void ADC_Init (tU32 INPSAMP,tU32 INPCMP,tU32 INPLATCH,tU32 Divider) {

    tU8 adc_channel[1] = {20};

    SIU.PCR[adc_channel[0]].B.APC = 1;          //PB[4] = PCR[20]

    //Normal Conversion Configuration
    ADC_0.CTR0.B.INPSAMP    = INPSAMP;          // set the sampling duration
    ADC_0.CTR0.B.INPCMP     = INPCMP;           // set the comparision durati
    ADC_0.CTR0.B.INPLATCH   = INPLATCH;         // set the latching phase du

    ADC_0.CTR0.B.OFFSHIFT   = 3;        // offset shift not used
    CGM.SC_DC[2].R = 0x80 + Divider;    // Enabled system clock divider #2
    ADC 0 clock prescaler = div0 + 1
    ADC_0.MCR.B.PWDN    = 0;            // exit from power down state (offset
    cancellation is performed)

    ADC_0.NCMR0.B.CH0  = 1;         // One Shot Conversion in Normal Mode on
    channel 0

    /* Presampling Setting */
    ADC_0.PSR0.B.PRES0 = 0;         // Channel 0 Presampling  Enable
    ADC_0.PSCR.B.PREVAL0 = 0;           // Internal Voltage VDD_HV_ADC0 selec
    for Presampling
    ADC_0.PSCR.B.PRECONV = 0;           // The ADC will perform a sampling
    followed by a conversion (the sampling is not bypassed)

    ADC_0.MCR.B.MODE = 0;   //set ONE SHOT MODE

    //DMA configuration for one shot mode sampling
    ADC_0.DMAE.B.DMAEN = 1; //DMA feature enable
    ADC_0.DMAE.B.DCLR = 0; //DMA request cleared by Acknowledge from DMA cont
    ADC_0.DMAR0.R = 0x1; //enable bit for channel 0



} /* InitDADC */
```

The following ADC start conversion function starts the ADC acquisition and allows to measure the time elapsed from the begin of the conversion command till the end of conversion (EOC) bitfield of the ADC ISR register.

Figure 6. ADC start conversion function

```
/******** ADC_StartConversion  *********/
//!  ADC starts conversion
/*!
*/
trOutput ADC_StartConversion(void)
{
    tU32 start_time = 0;
    tU32 stop_time = 0;
    trOutput ADC_Result;
    ADC_0.MCR.B.NSTART = 1;              // Start conversion
    start_time = STM.CNT.R;              // Start the timer counter
    while(!(ADC_0.ISR.B.EOC))            // wait till EOC interrupt occurred
    stop_time = STM.CNT.R;               // Stop the timer counter
    ADC_0.ISR.R = 0x00000003;            // Clear ECH bit */
    ADC_Result.clock_cycle = stop_time-start_time;  // measuring of the Tconv
    ADC_Result.result_conv = ADC_0.CDR[0].B.CDATA;; // result of the conversion
    return ADC_Result;
}
```

3.4 How to choose the right parameters

The example presented in this application note shows a typical sinusoidal signal with a frequency of 10 kHz and amplitude of 4.6 V and offset of 2.3 V.

Consider the following data from the SPC564B/C Data Sheet:

ADC Input sampling capacitances: C_S = 3 pF

C_{P1} = 3 pF

C_{P2} = 1 pF

Internal resistance of analog source:

R_{SW1} = 3 kΩ

R_{SW2} = 2 kΩ

R_{AD} = 2 kΩ

ADC voltage V_{dd} = 5 V

Following the equations 3 and 4 of the previous paragraph:

$t_S > 8.5 \times R_L \times (C_S+C_{P2}+C_{P1}) = 0.500$ µs

$t_S >> C_S \times (R_{AD} + R_{SW1}) = 0.012$ µs

In order to calculate the resistance for current limitation the target is minimize the Tconv:

R_L = 0.500µs / (8.5 x (C_S+C_{P2}+C_{P1})) = 8.4 KΩ

Note: Considering the available resistance it has been chosen 8.2 KΩ with a 10% of tollerance.

Suggested parameters for the ADC acquisition as follows:

- ADC frequency = 20 MHz
- TCK = 0.05 µs
- INPSAMP = 11
- INPCMP = 1
- INPLATCH = 0
- ndelay = 1

This theoretically implies:

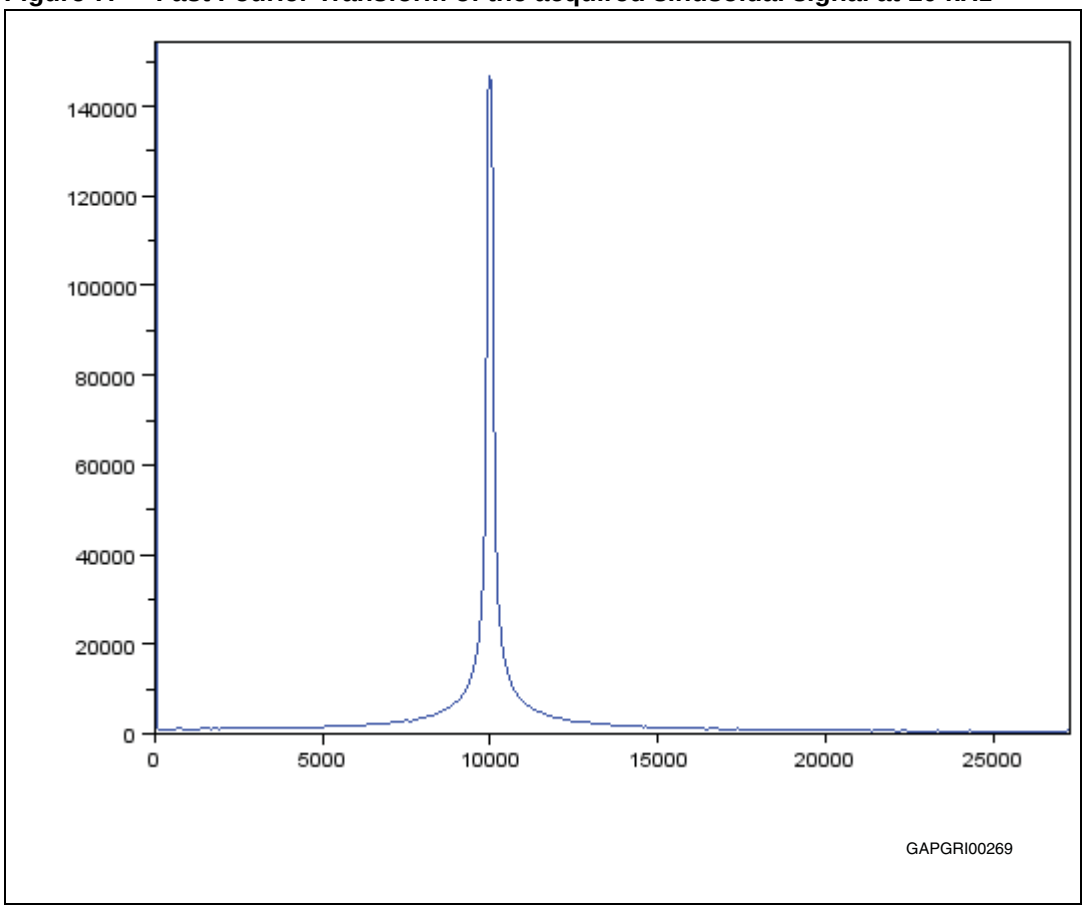
- t_{SAMPLE} = (INPSAMP - ndelay) x TCK = 0.5 µs
- t_{EVAL} = 10 x (INPCMP x TCK) = 0.5 µs
- t_{CONV} = t_{SAMPLE} + t_{EVAL} + (ndelay x TCK) = 1.05 µs

Choosing as sampling rate for ADC acquisition, a value 10 times greater than the signal frequency:

Sampling time = 1/(10 x 10 kHz) = 1/100kHz = 10 µs

Figure 7 shows the FFT for the sinusoidal signal acquired by the ADC:

Figure 7. Fast Fourier Transform of the acquired sinusoidal signal at 10 kHz



3.4.1 Degradation of the signal acquisition

When the sampling frequency is choosen only 2 times higher than the frequency of the input signal, as Nyquist-Shannon's sampling theorem suggests, the process of reconstruction of the acquired signal is guaranteed but not perfectly.

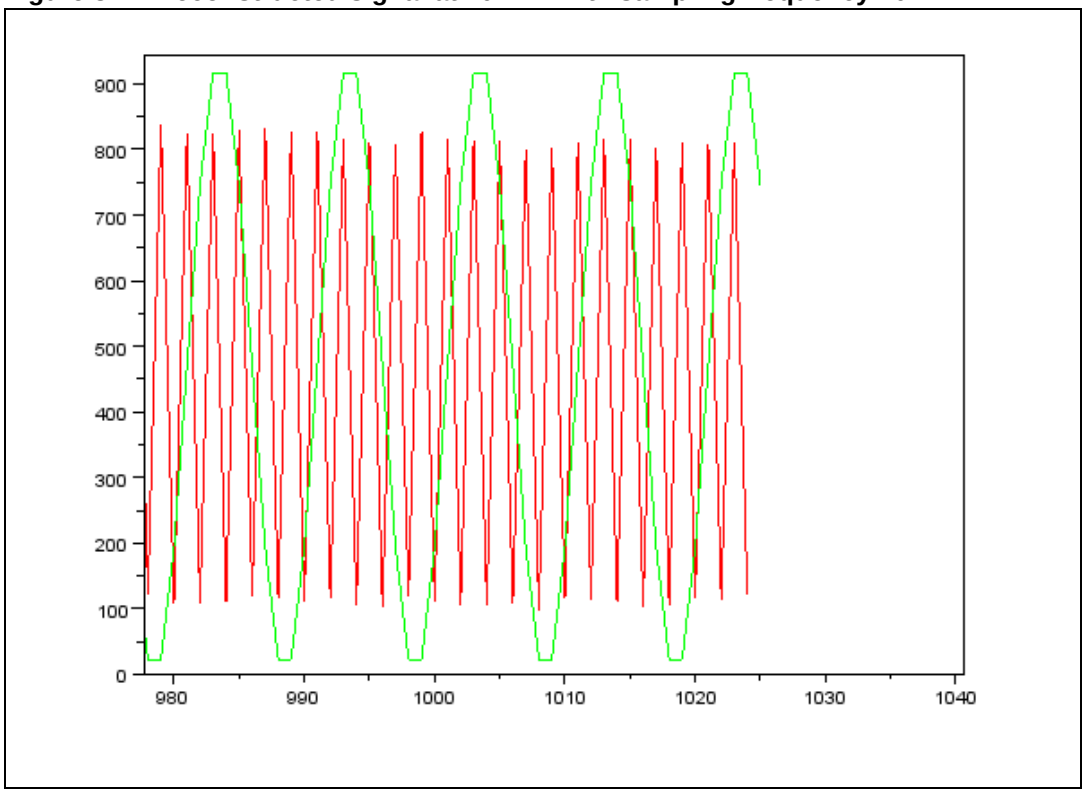
Indeed the sampling theorem provides only a sufficient condition but not a necessary one for a perfect reconstruction.

The condition is that the signal obtained from this reconstruction process cannot have any frequencies higher than one-half the sampling frequency. According to the theorem, the reconstructed signal will match the original one provided if the original signal contains no frequencies at or above this limit. This condition is called the Nyquist criterion.

if the condition is not satisfied, the resulting reconstructed signal may have a component at the frequency, but the amplitude and phase of the component generally will not match the original one.

The *Figure 8* shows the acquired sinusoidal signal using a sampling frequency only 2 times higher than the frequency of the input signal:

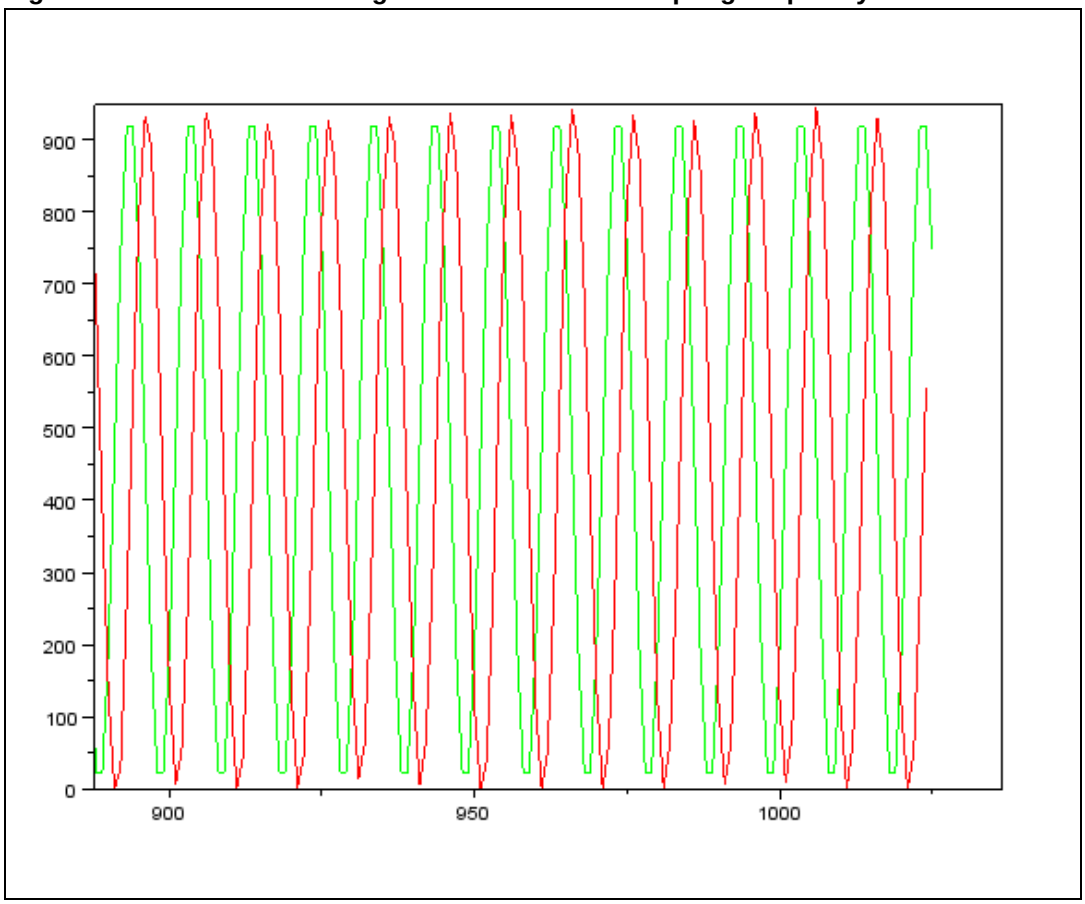
Figure 8. Reconstructed signal at 10 Khz with sampling frequency 20Khz



In this case the reconstructed signal in red cannot be considered a perfect copy of the original input signal in green.

The *Figure 9* shows a correct acquisition using a sampling frequency 10 times higher than the input signal.

Figure 9. Reconstructed signal at 10 Khz with sampling frequency 100Khz



The original input signal in green and the reconstructed signal in red can be considered equal with the same frequency and amplitude.

This implies that if theoretically it's possible to acquire input sinusoidal signal with a frequency up to 500KHz using a maximum sampling rate of 1MHz (1µs) , in order to reconstruct the original signal, only two samples per period will not provide the necessary amount of information to obtain a perfect matching.

Using the ADC with DMA feature to avoid a big amount of CPU workload and a period interrupt timer (PIT) to set a sampling rate, as suggested in this application note, it's possible to acquire sinusoidal signal with a good resolution of 5-10 samples for period only considering that the minimum time needed for a conversion is around 5-10 µs, then signals with a max frequency from 100KHz - 200 KHz.

3.5 Conclusions

There are four possible ways to optimise input signal acquisition:

1. Minimise the total source impedance. This means choosing sensors with low output impedance (not always easy for some types of sensor) and minimising the serial resistance of any protection devices between the analog source and the input pin (while still providing a voltage protection level compatible with the circuit specification).
2. Match the sample time to the analog source impedance. Using the formulas that relate the sample time to the source internal resistance to match the source resistance to one of the available sample times.
3. Match the sample time to the analog filter cut-off frequency, to remove high frequencies: the microcontroller sampling time (ADC silicon configuration) must be 5 to 10 times shorter than the period of the cut-off frequency of the low-pass filter on ADC input signal.
4. Reduce noise at the input pin. Add an external RC filter (with attention to the source internal resistance).

Appendix A Reference documents

A.1 Reference documents

SPC564Bxx, SPC56ECxx 32-bit MCU family built on the embedded Power Architecture® (Reference manual, RM0070, Doc ID 18196)

32-bit MCU family built on the Power Architecture® for automotivebody electronics applications (SPC564Bxx, SPC56ECxx Data Sheet, Doc ID 17478)

Revision history

Table 3. Revision history

Date	Revision	Changes
27-Sep-2012	1	Initial release.
19-Oct-2012	2	Added RPNs SPC56xBxx, SPC56xCxx SPC56xDxx.
13-Sep-2013	3	Updated disclaimer.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

ST PRODUCTS ARE NOT DESIGNED OR AUTHORIZED FOR USE IN: (A) SAFETY CRITICAL APPLICATIONS SUCH AS LIFE SUPPORTING, ACTIVE IMPLANTED DEVICES OR SYSTEMS WITH PRODUCT FUNCTIONAL SAFETY REQUIREMENTS; (B) AERONAUTIC APPLICATIONS; (C) AUTOMOTIVE APPLICATIONS OR ENVIRONMENTS, AND/OR (D) AEROSPACE APPLICATIONS OR ENVIRONMENTS. WHERE ST PRODUCTS ARE NOT DESIGNED FOR SUCH USE, THE PURCHASER SHALL USE PRODUCTS AT PURCHASER’S SOLE RISK, EVEN IF ST HAS BEEN INFORMED IN WRITING OF SUCH USAGE, UNLESS A PRODUCT IS EXPRESSLY DESIGNATED BY ST AS BEING INTENDED FOR “AUTOMOTIVE, AUTOMOTIVE SAFETY OR MEDICAL” INDUSTRY DOMAINS ACCORDING TO ST PRODUCT DESIGN SPECIFICATIONS. PRODUCTS FORMALLY ESCC, QML OR JAN QUALIFIED ARE DEEMED SUITABLE FOR USE IN AEROSPACE BY THE CORRESPONDING GOVERNMENTAL AGENCY.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.
Information in this document supersedes and replaces all information previously supplied.
The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2013 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com