AN1903
APPLICATION NOTE

A First Order Sigma Delta Converter Using ST52x440

1 INTRODUCTION

This application note shows the implementation of a Sigma-Delta (ΣΔ) converter using a member of ST52 microcontroller family.

Although many devices of the ST52 family already have a built-in A/D Converter, another kind of converter (precisely a ΣΔ converter) is implemented via sofware.

In this specific case, an ST52x440 is used. This device includes an Analog Comparator among its peripherals, which can also be used as a single slope ADC, thus allowing an easy implementation in firmware of the SD conversion algorithm using few additional external elements.

The ΣΔ conversion algorithm is easily implemented in firmware using few additional external elements.

Attached you can find the firmware package.

2 SIGMA-DELTA THEORY

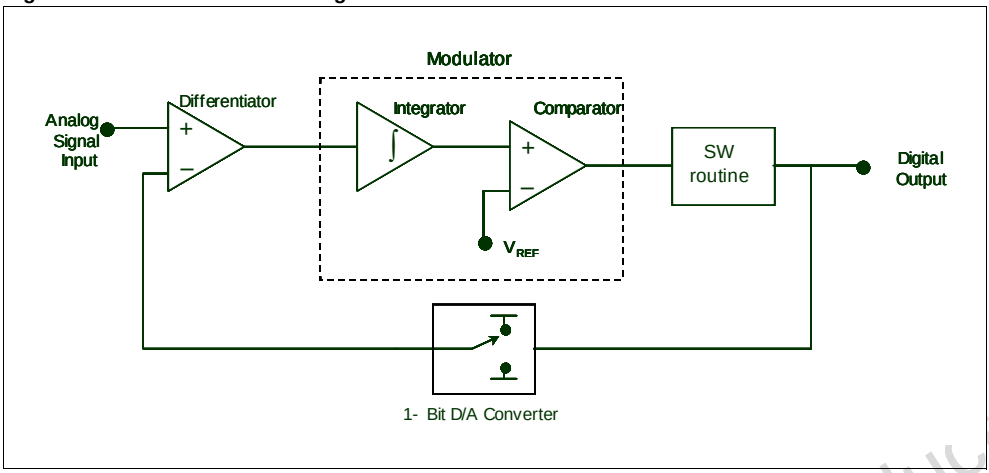
Sigma-Delta modulation has emerged as the architecture of choice for high-resolution analog-to-digital conversion using simple and low-precision analog components. In this kind of converter, an increased resolution is obtained reducing the conversion bandwidth or increasing the clock speed, over sampling the signal to convert.

Over sampling has become popular because it avoids many difficulties encountered with conventional methods for A/D and D/A conversion, especially in those applications requiring high-resolution and using relative low-frequency signals. Moreover, sampling at high frequency eliminates the need for abrupt cut-offs in the analog anti-aliasing filters.

Classical Sigma-Delta circuits are composed of a ΣΔ Modulator which provides a high-speed one-bit data string followed by a digital filter and decimator to produce high-resolution data. The lowpass ΣΔ modulator consists of one or more integrators and a one-bit quantizer. The basic concept underlining Sigma-Delta Converters is the use of feedback in order to improve the effective resolution of the quantizer.

Figure 1 shows a ΣΔ Converter block diagram. The input signal is added to the signal coming from the feedback loop. This differentiated signal is then sent to the modulator. The comparator acts like a one-bit quantizer, its output is sent to the D/A converter of the feedback branch and to a digital filter which provides with time the multi-bit Digital Output.

Figure 1. ΣΔ Converter Block Diagram

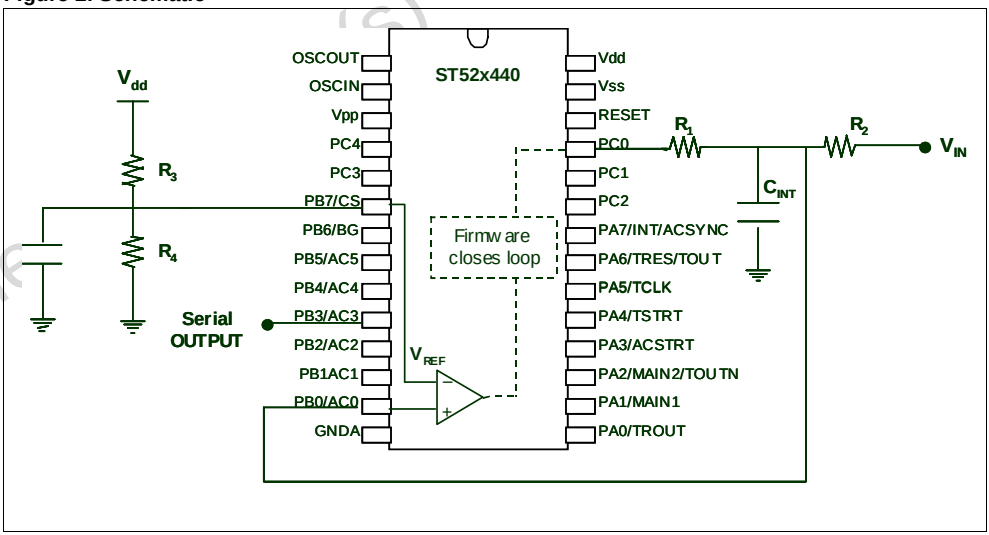


3 IMPLEMENTATION WITH ST52X440 MICROCONTROLLER

In Figure 2 a scheme of how the ΣΔ Converter is implemented with the ST52x440 microcontroller is shown. If compared to a classical Sigma-Delta Converter, the external capacitor C_{INT} absolves the role of integrator; the comparator is built inside; voltage reference V_{REF} is supplied by partitioning the voltage V_{SS} by means of two equal resistances R_3 and R_4; while the one-bit D/A Converter and the digital filter are implemented by the firmware. By means of the two equal resistances R_3 and R_4 (center scale set to 2.5V) to a conversion of signals in the range 0 divided by 5V is allowed.

When pin PC0 is set high by the feedback loop, the voltage at pin PB0 increases in magnitude until it becomes higher than V_{REF}. Once this has occurred, pin PC0 is set low and the voltage at pin PB0 decreases consequently until the comparator sets PC0 high. In the meantime, the number of zeros occurring at the output of the comparator (which means integrated signal higher than reference signal) are counted. This number is strictly related to the input voltage V_{IN}.

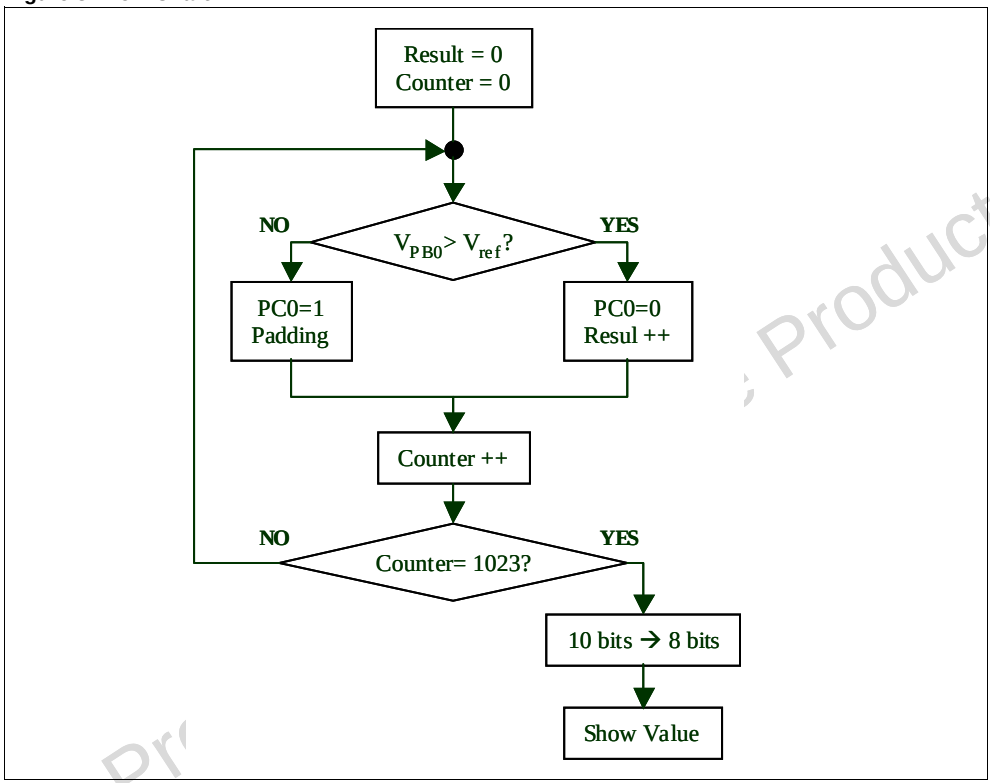
Figure 2. Schematic



When the signal to be converted comes from a high impedance source, a buffer has to be placed between this signal source and the input terminal of R2 in order to decouple impedances.

In Figure 3 a flow chart of the firmware used is shown. When V_{PB0} (which is the input to the comparator) is higher than the reference signal pin PC0 is set low and the variable result is incremented; otherwise PC0 is set high and padding instructions are executed in order to keep all paths through the code equal. As soon as the variable *Counter* reaches value 2^N-1 (with N = num of bits) the result is stopped and the result obtained is filtered and then serially shown through pin PB3. Serial communication is implemented by software by means of the internal timer, which fixes the BAUD rate.

Figure 3. Flow Chart



4 FIRMWARE

The implementation used in this application performs the comparison between the integrated signal and the reference signal, the calculation of the result and closes the feedback loop. Moreover, it processes the results and gives them serially to the output pin PB3 so that they can be acquired, displayed and eventually processed by a PC.

The software was developed in Visual FIVE, the visual Development Tool of the ST FIVE Family of micros and it is displayed in the following section.

Figure 4 and 5 are two screens taken from the attached firmware developed in VisualFIVE describing respectively the flow chart of the main program and the result computation algorithm.

Figure 4. Main Loop

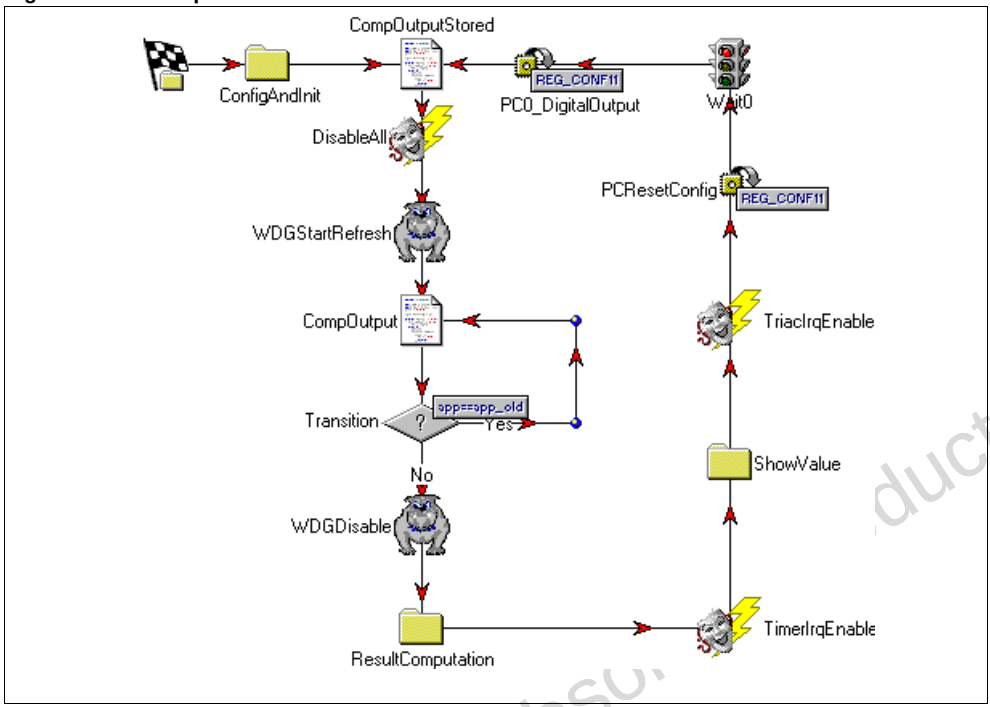
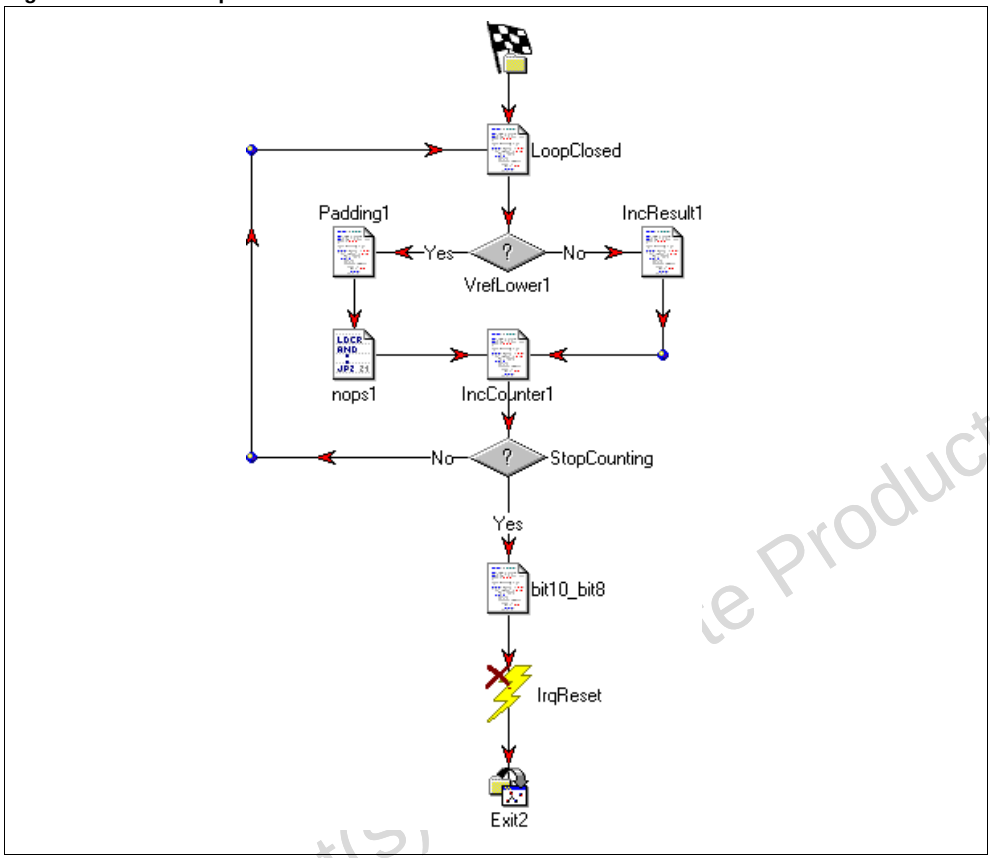


Figure 5. Result Computation



The following steps refer to the attached firmware too.

4.1 Peripheral Configuration and Micro Initialization

In this step, the micro is initialized, in particular with:

– timer: it is configured to work @ 1.25 MHz, interrupt source on rising edge, Timer Count = 131(in order to have BAUD rate equal to 9600 bits/s);
– triac driver: configured to work in PWM mode;
– A/D : set in comparator mode;
– watchdog: counting period equal to 15.6 ms
– initialization of the ports (analog input, digital input/output).

4.2 Result Computation and Loop Closing

Each cycle consists of two phases: during the first cyle, data is acquired, while during the second cycle, data is shown.

While converted data is processed and shown, the capacitor C_{INT} is charged depending on the input voltage and on the last comparator output. In order to keep initial conditions independent from the input voltage for each conversion, data is acquired as soon as transition (0/1) is detected at the output of the comparator. This is achieved by using two variables. One is used to store the old value of the comparator output and the other one stores the current value. As soon as these two variables become different, computation starts. Comparator output is sent to pin PC0 closing the feedback loop. If this value is high the variable *result* is incremented; otherwise padding instructions are executed.

4.3 Result Shown

The digits of the result are changed into ASCII and are serially sent to pin PB3. The timer synchronizes the transmission permitting a serial communication at 9600 baud. Sampling frequency is set by triac driver peripheral used as a timer; in fact it generates an interrupt each 15.6 ms and the conversion starts as soon as this interrupt occurs.

Output data can be acquired by Windows HyperTerminal.

5 CONCLUSION

Some results are shown here.

Two different signals were used as input signals: a saw tooth waveform and a triangular wave form both signals have a 100mHz frequency and a 0 ÷ 5 V range (thus with an output signal in 0\255 range). In Figures 6 and 7 the respective output signals are plotted.

Figure 6. Sawtooth Waveform - Output

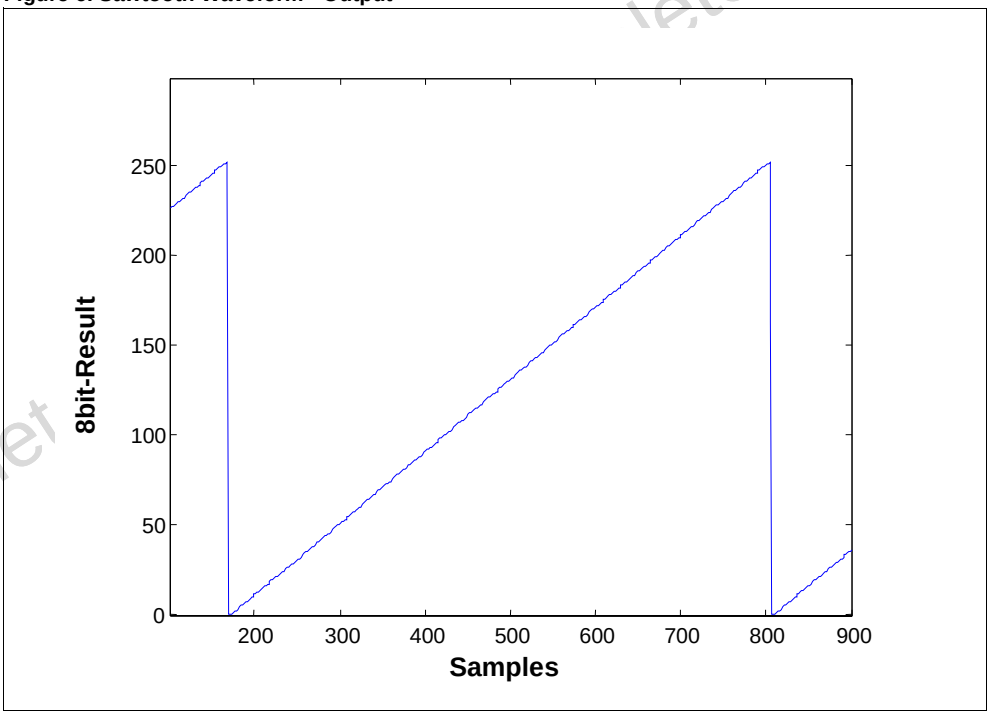
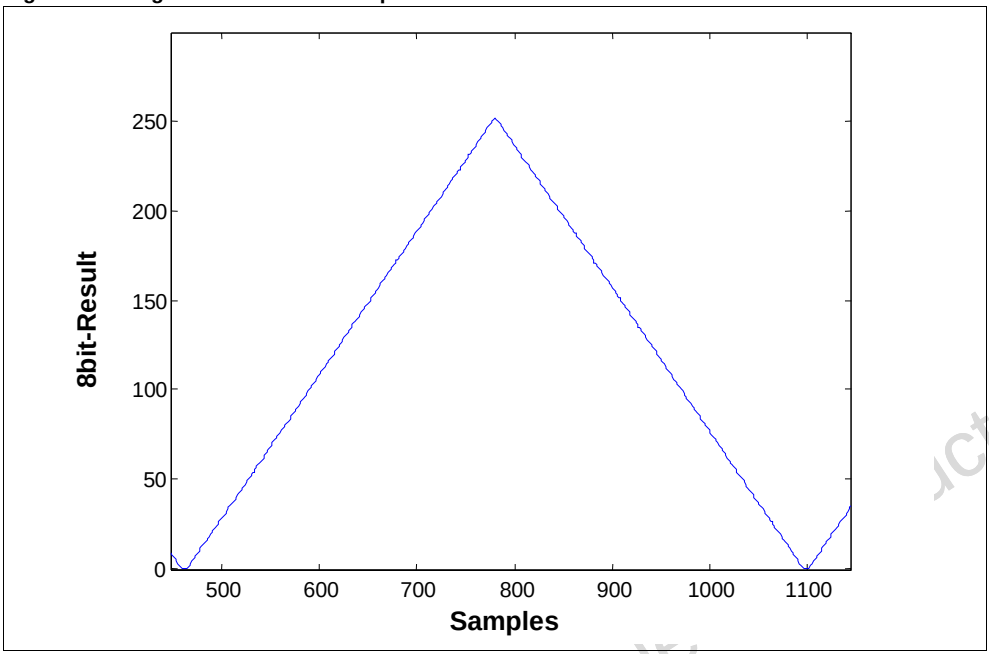


Figure 7. Triangular Wave Form - Output



6 REFERENCES

1. S.R. Norsworthy, R. Shreier, G.C. Temes “Delta-Sigma Data Converters, Theory, Design and Simula tion”, New York - IEEE PRESS, 1996
2. J.C. Candy, G.C. Temes “Oversampling Delta-Sigma Converters”, New York - IEEE PRESS, 1991