

Analog to Digital Conversion

Natalie Hannon

Team 9: Safety Enhancement Features for the 21st Century Automobile
ECE 480: Capstone Design
Michigan State University

November 14, 2008

Contents

1	Executive Summary	1
2	Objective	1
3	Development	1
4	Issues	4
5	Results	5
6	Conclusion	7
	References	7

1 Executive Summary

Analog to digital conversion (ADC) is the process of representing continuous signals in binary so that the data can be stored and manipulated. Digital information is different from analog information in two distinct ways - it is sampled and quantized. The number of bits used to represent the signal determines the resolution. In the case of the “alerting system, A/D conversion is required to transfer the external acoustic environment into digital data that can be used for matched filtering.

2 Objective

This document covers the use of the 12-bit ADC module for Microchips dsPIC30F series. It explains how the ADC function is coded in C within the context of how an A/D converter works in general.

3 Development

The first step in coding the ADC function is to configure the A/D module. There are three control registers within the A/D module. The first register, ADCON1, has several bits that must be set for operation:

- ADON is the operating mode bit that must be set to use the A/D module. It is typically set after all other bits in the A/D module are set.
- FORM determines the data output format. In this application, a fractional data type is used.
- SSRC is the conversion trigger source select that determines how the A/D module is controlled. There are several options - in this case, Timer3 on the microcontroller is

used to end sampling and start conversion.

- ASAM is the A/D sample auto-start bit which allows for continuous sampling. This bit is set for this application.

The second control register, ADCON2, has several bits which determine what happens with the sampled data:

- VCFG sets the voltage reference configuration. The voltage reference is directly related to the resolution in the A/D conversion. For the 12-bit ADC module, the resolution is equal to $2^{12} = 4096$ bits. With a reference voltage of 5Vdc, the data can be resolved into $\frac{5V}{4096} = 1.22mV$. In this application, the line and ground voltages are specified as unique pins so that the reference voltage can be changed. This is advantageous to using the microcontroller line and ground pins as the voltage reference, which can not be changed.
- BUFS tells the A/D which buffer to fill with data, which is necessary for continuous sampling, as in this application.
- SMPI determines how often to interrupt at the completion of each sample/convert sequence. Currently, this bit is set at maximum but could be lowered to perform matched filtering quicker.

The third control register, ADCON3, has several bits which deal with timing. It is important to know how long the A/D conversion will take.

- For a sample rate of 5 KHz, the total conversion time is equal to $\frac{1}{5KHz} = 200\mu s$. This microcontroller has up to 30 million instructions per second (MIPS) operation, as specified in the data sheet. This means that the instruction cycle time T_{cy} is equal to $\frac{1}{30 \times 10^6} = 33.3ns$. For correct A/D conversions, the conversion clock T_{AD} minimum

is 668ns for a speed of up to 100ksps and Vdd range from 3.0V to 5.5V. Ultimately, if the A/D conversion clock select bit ADCS is set to its maximum value of 63, then: $T_{AD} = T_{cy}(ADCS + 1)/2 = 1.088\mu s$. The A/D module will take $14T_{AD} = 15.232\mu s$ to convert each sample.

- From this, the total time for each sample/convert sequence can be determined. Since Timer3 controls the sampling, it has to timeout after every $200\mu s$. As a result, the module will stop sampling and trigger a conversion on every Timer3 timeout. At that time, the conversion process starts and completes $15.232\mu s$ later. Effectively, the module samples for $184.768\mu s$ and converts for $15.232\mu s$.

The last two registers of importance to this application are the ADCHS register and the ADPCFG register. ADCHS is the A/D input select register and specifies which of the 16 available input pins will be used for ADC. In this application, it is important that the A/D module does not share an input pin with anything else so that the microcontroller can not confuse the data. ADPCFG is the A/D port configuration register. Here, all channels are set to digital mode except for the input pin specified in ADCHS.

Ultimately for operation, the ADC function has to initialize Timer3 and its interrupt flag and interrupt enable bits to zero. After the registers are configured, the A/D interrupt flag bit should be cleared and the A/D interrupt enable bit should be set. Finally, the A/D module can be started along with Timer3.

An interrupt service routine (ISR) moves converted data to the main code to perform matched filtering. The commented code is provided.

```
{//_ADCInterrupt() is the A/D interrupt service routine (ISR)
// that moves converted data to main

void __attribute__((interrupt, no_auto_psv)) _ADCInterrupt(void)
{
    IFS0bits.T3IF = 0; //Clear the Timer3 Interrupt Flag
    IFS0bits.ADIF = 0; //Clear the A/D Interrupt Flag

    int i = 0;

    adcPtr = &ADCBUF0 ; //pointer is set to the address of buffer
//Copy the A/D conversion results to variable "inputSignal"
    for (i=0;i<16;i++)
    {
        *iPtr++ = *adcPtr++;
    }
    if (iPtr > &inputSignal[256]) { //if address does not exist, stop
doFilterFlag = 1; //send A/D data to main
}
}
}
```

4 Issues

One primary concern with ADC is the integrity of the converted data. The voltage reference ceiling and floor must be set such that all values are captured. If the voltage references are not high or low enough, values will be missed. In the example shown in Figure 1, the A/D module will seemingly miss voltage values above and below 4V.

Another primary concern with ADC is the speed at which it can perform. For a sampling rate of 5 KHz, this should not pose a problem for this application.

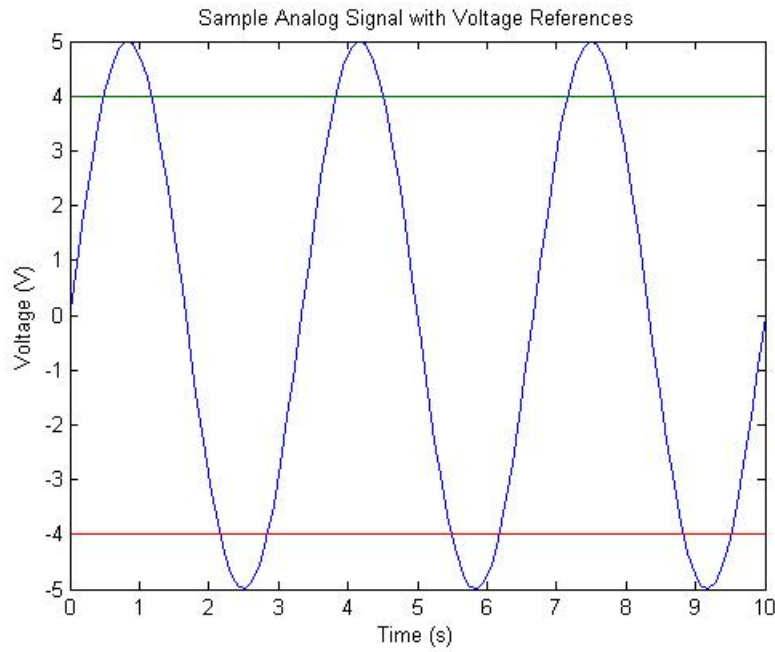


Figure 1: Voltage Reference Example

5 Results

The ADC function was tested by connecting a 2000 Hz, 5V p-p sinusoid waveform with a 2.5V DC offset from the function generator to the analog input pin. The Watch window in the MPLAB software was used to verify that hex and binary values of the converted data appeared at each address. This can be seen in Figure 2 for the first 40 samples.

Watch

Add SFR ACCA Add Symbol _SP

Update	Address	Symbol Name	Value	Binary
	85C	inputSignal		
	85C	[0]	0xCFC0	11001111 11000000
	85E	[1]	0xFFFF	11111111 11110000
	860	[2]	0xB470	10110100 01110000
	862	[3]	0x3780	00110111 10000000
	864	[4]	0x06C0	00000110 11000000
	866	[5]	0x53B0	01010011 10110000
	868	[6]	0xD1C0	11010001 11000000
	86A	[7]	0xFFFF	11111111 11110000
	86C	[8]	0x1080	00010000 10000000
	86E	[9]	0x1E40	00011110 01000000
	870	[10]	0x9130	10010001 00110000
	872	[11]	0xF7C0	11110111 11000000
	874	[12]	0xE8D0	11101000 11010000
	876	[13]	0x74F0	01110100 11110000
	878	[14]	0x1010	00010000 00010000
	87A	[15]	0x1F90	00011111 10010000
	87C	[16]	0x9470	10010100 01110000
	87E	[17]	0xF900	11111001 00000000
	880	[18]	0xE6D0	11100110 11010000
	882	[19]	0x0690	00000110 10010000
	884	[20]	0x4FF0	01001111 11110000
	886	[21]	0xCE30	11001110 00110000
	888	[22]	0xFFFF	11111111 11110000
	88A	[23]	0xB620	10110110 00100000
	88C	[24]	0x38B0	00111000 10110000
	88E	[25]	0x0730	00000111 00110000
	890	[26]	0x5290	01010010 10010000
	892	[27]	0xD0F0	11010000 11110000
	894	[28]	0xFFE0	11111111 11100000
	896	[29]	0xB430	10110100 00110000
	898	[30]	0x1D40	00011101 01000000
	89A	[31]	0x8FD0	10001111 11010000
	89C	[32]	0xF670	11110110 01110000
	89E	[33]	0xEA80	11101010 10000000
	8A0	[34]	0x7650	01110110 01010000
	8A2	[35]	0x1050	00010000 01010000
	8A4	[36]	0x1F00	00011111 00000000
	8A6	[37]	0x92D0	10010010 11010000
	8A8	[38]	0xF7F0	11110111 11110000
	8AA	[39]	0xE780	11100111 10000000

Watch 1 Watch 2 Watch 3 Watch 4

Figure 2: Watch Window A/D Data

6 Conclusion

Analog to digital conversion is vital to the “alerting system” application and numerous other applications that require digital representation of analog signals. Built in modules like that of Microchip’s dsPIC30F series make implementation relatively easy. However, knowledge of each register and bit of the A/D module is required for proper functionality. Further, additional code is required to utilize the data after conversion. Ultimately, ADC is an important topic for this application.

References

- [1] “Analog to Digital Conversion” IOtech. 2008. 12 Nov 2008.
http://www.iotech.com/handbook/chapt_2.html
- [2] “Microchip dsPIC30F3014/4013 Data Sheet” Microchip. 2008. 04 Nov 2008.
<http://ww1.microchip.com/downloads/en/DeviceDoc/70138F.pdf>
- [3] Smith, Steven W, Ph.D. “ADC and DAC”
The Scientist and Engineer’s Guide to Digital Signal Processing. 1997. 12 Nov 2008.
<http://www.dspguide.com/CH3.PDF>
- [4] “Section 18. 12-bit A/D Converter” Microchip. 2008. 04 Nov 2008.
<http://ww1.microchip.com/downloads/en/DeviceDoc/70065D.pdf>