Introduction:

After testing the analog-to-digital converters (ADCs) on the Ko Brain board and the Ranger robot (Freescale Semiconductor 56F8347 and Analog Devices AD7490), it was determined that using an ADC chip with more than 12 bits of accuracy and features such as adjustable gain, multiplexed inputs, and low power would be best suited to gathering high quality data from a robot's sensors.

Since our applications only require relatively low sampling rates, in the kilosamples per second range, the sigma-delta converter architecture was considered as an alternative to more traditional converter architectures. This type of ADC has the benefits of low cost, high resolution, and low power, at the expense of limited bandwidth. Basically, a sigma-delta converter operates by oversampling the data (sampling at a much higher rate than the desired output rate), shaping the quantization noise so that most of the noise lies outside of the frequency range of interest, using a digital filter to remove the noise at high frequencies outside of the frequency band of interest, and using a decimator to reduce the data rate from the original sampling rate to the desired output data rate.

To simplify data acquisition, an ADC with user-friendly features, such as adjustable data rates, programmable input gain and offset, multiplexed inputs, and built-in filtering, was desired. This would simplify circuit design when incorporating ADCs into a robot's electronics system, because most of the circuitry needed to maximize the usefulness of an ADC is already included in this type of ADC chip.

A search of the primary analog device manufacturers' websites yielded four potential sigma-delta converters with the desired feature set. These included the Maxim MAX1402/3, the Texas Instruments ADS1174/8, the Texas Instruments ADS1112, and the Analog Devices AD73360L. Based on their available output data rates, adjustable input gain, number of available inputs, and power consumption, the Maxim MAX1402 was selected as the best option.

The Maxim MAX1402 offers 18 bits of resolution, with 16 bits of accuracy for data rates up to 480 samples per second (sps). It can scan between three fully differential or 5 pseudo-differential input channels, which would allow one ADC to gather data from multiple sensors. The input gain and offset are programmable to spread the input signal over the full input range. It also has input buffering to isolate the inputs from the capacitive loading of the gain and modulator circuitry. The chip utilizes a +5V analog supply, a +3V or +5V digital supply, and is SPI compatible. The converter's output data rate is programmable from 50 sps to 4800 sps.

Purpose:

The purpose of this testing was to determine the "real-world" noise levels present in the Maxim MAX1402 analog-to-digital converter as well as the chip's measured power consumption. The tests included several variables, including output data rate, input voltage gain, digital filtering, pseudo-differential or fully-differential inputs, input buffering, and clock frequency.

Created 5/14/08 by Nicole Rodia Cornell University

Experimental apparatus:

The experiments were conducted using the Maxim MAX1402 Evaluation Kit with the Maxim 68HC16 microcontroller module. These two boards were plugged together for the testing (see Figure 1). The 68HC16 module was powered by a DC voltage supply at 7.1 V. The tests were executed and output data collected via a serial (RS232) connection. Figure 1 shows the evaluation board configured for pseudo-differential inputs, using input pins AIN3 and AIN6. Figure 2 shows the resistor network and connectors used to create an AC signal with DC offset from the function generator and +5V DC power supply. A schematic of this set-up is shown in Figure 3. After some of the testing was completed, it was noticed that changing the frequency output by the function generator was affecting the DC offset of the signal, so a 0.015 uF capacitor was added in series with the input resistance, as shown in Figure 4. The HP 3300A function generator used in the testing is pictured in Figure 6. Since this function generator was used for gathering the 2.5 GHz data.



MAX1402 Testing Set-up

Figure 1. This photograph shows the basic testing setup for all experiments done using the MAX1402 analog-to-digital converter. It includes the Maxim 68HC16 module on the left, plugged into the Maxim MAX1402 Evaluation Kit (right). The 68HC16 module is connected to a laptop via a serial connection (upper left).

The same evaluation board set-up was used for the testing with fully-differential inputs, which utilized input pins AIN1 and AIN2. These differential analog inputs were from a Transcell Model BSA-250 Load Cell. The load cell's white and green signal wires were connected to the analog input pins, and the load cell power and ground wires were connected to the MAX1402 Evaluation Board OUT1 and GND pins. The OUT1 pin provided a 200 uA supply current for the load cell. Since this current was rather small to

Created 5/14/08 by Nicole Rodia Cornell University Modified 5/18/08 11:32 PM by Nicole Rodia MAX1402 Noise Test Report.doc power the load cell, the input it provided to the MAX1402 was essentially a ground signal.



Resistor Network for Function Generator Input

Figure 2. This photograph illustrates the resistor network used to create the input to the MAX1402 from the function generator.



Resistor Network Schematic

Figure 3. This schematic shows the resistor network used to create the pseudo-differential input signal used to test the MAX1402.

Created 5/14/08 by Nicole Rodia Cornell University Resistor and Capacitor Network Schematic



Figure 4. This schematic shows the resistor network, plus a capacitor, used to create the pseudo-differential input signal used to test the MAX1402. This configuration was used for the 2.5 MHz and DC input signals.

Modified 5/18/08 11:32 PM by Nicole Rodia MAX1402 Noise Test Report.doc



Figure 5. This HP 3300A Function Generator with variable output frequency, amplitude, and waveform type was used to create an AC test signal at varying frequencies for the MAX1402.



Function Generator Output and MAX1402 Test Signal Input on Oscilloscope

Figure 6. This shows the function generator's AC output and the AC test signal with DC offset used as the test input for the MAX1402.

Procedure:

Data Acquisition from the MAX1402 Evaluation Board:

1. Supply 7.0-7.2 V to the 68HC16 module and flip the power switch to the on position.

Created 5/14/08 by Nicole Rodia Cornell University Modified 5/18/08 11:32 PM by Nicole Rodia MAX1402 Noise Test Report.doc

Function Generator

- 2. Connect the serial cable from the 68HC16 module to a computer and open the Maxim MAX1402 Evaluation Kit software. Wait for the computer to initialize its connection with the board.
- 3. Select the appropriate data inputs from the "Inputs" drop down menu.
- 4. Real time data should appear in the Graph window (see Figure 7).
- 5. To collect a data sample, click on the "Sample" button, and select the desired parameters.



MAX1402 Evaluation Kit Software

Figure 7. This screen shot of the MAX1402 Evaluation Kit software shows the voltage and equivalent number of bits for each analog input in the bottom window. The graph at the top tracks the input value over time.

Load Cell Noise Tests:

Data sets of 256 samples were collected for several combinations of output data rate, input gain, input buffering, and digital filtering. Output Data Rates: 60, 200, 600, 1200, 4800 sps Input Gain: +1, +16, +128 Input Buffering: on, off Digital Filtering: on (low noise mode), off (low power mode)

Function Generator Noise Tests:

Data sets of 256 samples were collected for several combinations of input frequency, output data rate, input gain, input buffering, and digital filtering. These included the same range of operating modes as the Load Cell Noise Tests. Input Frequencies: 0 Hz (DC), 20 Hz, 80 Hz, 200 Hz, 800 Hz, 2 kHz, 4 kHz, 8 kHz, 50

kHz, 100 kHz, 2.5 MHz

Function Generator Power Consumption Tests:

For various combinations of the MAX1402 operating parameters, the voltages across resistors R7 and R8 on the MAX1402 Evaluation Board, both 10Ω , were measured. The current through R8 can be used to calculate the total power consumption of the chip, and the current through R7 can be used to calculate the analog power consumption of the chip. The difference between them is the digital power consumption. An additional parameter considered for these tests was the clock frequency.

Clock Frequencies: 1.024, 2.048, 2.4576, 4.9152 MHz

Data:

All tests were conducted using a clock frequency of 2.4576 MHz, except where noted for the power consumption tests, with a sample size of 256 samples per test. The device's input voltage range was 0 to 2.5 V. The chip's 18 bits of resolution provide an output range of 0 to 262144 levels. The register settings for the adjustable gain are shown in Table 1.

Gain	G0	G1	G2	
1x	0	0	0	
16x	1	0	0	
128x	1	1	1	

Register Configurations for Adjustable Gain

Table 1.

To quantify the amount of noise in a given data set, the standard deviation of the data points was used. The value was calculated using the Matlab command std(X). This command uses the definition of standard deviation given in Figure 8. The Matlab function data_analysis(), used to process the data collected in the noise tests, can be found in the Appendix.

Standard Deviation

$$s = \left(\frac{1}{n-1}\sum_{i=1}^{n} (x_i - \bar{x})^2\right)^{\frac{1}{2}}$$

Figure 8. The definition of standard deviation used in the Matlab command std(X). This definition was used to calculate the standard deviation of the collected data sets.

Load Cell Noise Tests:

The nominal voltage output from the load cell was -0.00078V = 131032 bits. A plot of data from the load cell noise tests is shown in Figure 9. This demonstrates the low noise levels of the chip at low output data rates. The output value is only changing by a few bits at a data rate of 60 sps. Figure 10 shows a comparison of the noise levels for varying data output rates, output modes, gain, and input buffering. All of the data from this set of tests is shown in Table 2.



Figure 9.



Figure 10.

Load Cell Tes	t Data	4/18/2008			
Unbuffered, 1x ga	in	Buffered, 1x gain			
Data Rate (sps)	Std Dev (bits)	Data Rate (sps)	Std Dev (bits)		
60	0.68885	60	0.73247		
200 low noise	4.42940	200 low noise	4.71687		
600 low noise	8.10397	600 low noise	8.30974		
1200 low noise	8.48667	1200 low noise	8.60406		
200 low power	26.82460	200 low power	28.89060		
600 low power	48.80230	600 low power	49.18580		
1200 low power	49.80060	1200 low power	50.12610		
4800	48.62370	4800	46.87230		
Unbuffered, 16x g	ain	Buffered, 16x gain			
Data Rate (sps)	Std Dev	Data Rate (sps)	Std Dev		
	(bits)		(bits)		
60	1.30016	60	1.31/96		
600 low noise	8.96012	600 low noise	8.38106		
1200 low noise	8.45769	1200 low noise	8.05575		
600 low power	48.21050	600 low power	48.13940		
1200 low power	50.66640	1200 low power	48.03570		
4800	51.86920	4800	49.35360		
Unbuffered, 128x	gain	Buffered, 128x gai	n		
Data Rate (sps)	Std Dev	Data Rate (sps)	Std Dev		
	(bits)		(bits)		
60	7.39412	60	9.21599		
600 low noise	18.18370	600 low noise	20.61750		
1200 low noise	18.73880	1200 low noise	19.72790		
600 low power	48.15740	600 low power	54.84550		
1200 low power	51.59180	1200 low power	59.32190		
4800	50.50860	4800	54.11740		

Table 2.

Function Generator Noise Tests:

The input circuit for these tests provided a theoretical input voltage of 1.46 ± 0.643 V. This translated into a nominal voltage of 1.44308 V = 204916 bits output from the MAX1402. Input frequencies below about 10 kHz can't be considered "noise" frequencies because the AC signal begins to appear on the digital output – it is no longer filtered out. Thus, only data at 8 kHz and higher frequencies was considered in these results. Figures 11 and 12 show the raw data for output data rates of 60 sps and 4800 sps with a DC input, which show the least and most amounts of noise, respectively. A comparison of noise due to varying gain and data rate is shown in Figure 13. The function generator test data is shown in Table 3. All of these tests were conducted with input buffering off.

Further testing showed that noise at the sampling frequency is not filtered out and appears on the output. For an output data rate of 60 sps with a clock frequency of



Figure 11.





Created 5/14/08 by Nicole Rodia Cornell University Modified 5/18/08 11:32 PM by Nicole Rodia MAX1402 Noise Test Report.doc 2.4576 MHz, the sampling frequency is 38.4 kHz. With an input noise at this frequency, there was up to 0.3 V of noise on the output. The noise level when the input frequency was adjusted to be slightly offset from the sampling frequency was less than 0.2 mV. Input frequencies that were at harmonics of the sampling frequency, such as 76.8 kHz, also appeared on the output, but at a somewhat lower amplitude. However, setting the input frequency to the output data rate frequency (e.g. 600 Hz) had no effect on the noise levels.

Function Generator Power Consumption Tests:

These tests were conducted at a variety of clock frequencies and data output rates. Another variable considered was the digital filtering – "low power" or "low noise" mode. Note that changing the gain did not affect the power consumption. A plot of noise versus power consumption is shown in Figure 14. The data points are for different data output rates, and the noise and power for 600 and 1200 sps were measured in both low noise and low power modes. Note that the power consumption was measured with an input frequency of 20 kHz while the noise levels were measured with an input frequency of 50 kHz. Since these high frequencies are filtered out by the chip, however, the noise and power for each frequency should be comparable.



Figure 13.

Function Gen					
04/27/2008, 04					
Input Freq (Hz)	Data Rate (sps)	Mode	Gain	Std Dev (bits)	
0 (DC)	60		1	2.1656	
0 (DC)	60		128	2.1360	
0 (DC)	600	low noise	1	13.4587	
0 (DC)	600	low noise	128	13.7798	
0 (DC)	600	low power	1	66.3367	
0 (DC)	600	low power	128	63.1516	
0 (DC)	1200	low noise	1	14.1654	
0 (DC)	1200	low noise	128	13.2721	
0 (DC)	1200	low power	1	67.2389	
0 (DC)	1200	low power	128	71.9104	
0 (DC)	4800		1	67.5594	
0 (DC)	4800		128	66.4122	
2.00E+03	60	low noise	1	1.7742	
2.00E+03	600	low noise	1	15.0999	
2.00E+03	1200	low noise	1	76.0251	
4.00E+03	60		1	1.5654	
4.00E+03	1200	low noise	1	13.3888	
4.00E+03	2400	low noise	1	74.7930	
4.00E+03	2400	low power	1	115.3320	
8.00E+03	60		1	1.7284	
8.00E+03	600	low noise	1	15.3934	
8.00E+03	1200	low noise	1	15.2707	
8.00E+03	4800		1	107.5470	
5.00E+04	60		1	1.5548	
5.00E+04	60		16	1.6283	
5.00E+04	60		128	1.7279	
5.00E+04	600	low noise	1	14.8411	
5.00E+04	600	low noise	16	15.2876	
5.00E+04	600	low noise	128	14.6958	
5.00E+04	600	low power	1	72.7320	
5.00E+04	600	low power	16	68.5494	
5.00E+04	600	low power	128	68.4082	
5.00E+04	1200	low noise	1	12.7666	
5.00E+04	1200	low power	1	64.4905	
5.00E+04	4800		1	71.2521	
1.00E+05	60		1	2.0263	
1.00E+05	600	low noise	1	15.1620	
1.00E+05	600	low power	1	66.3712	
1.00E+05	1200	low noise	1	14.5099	
1.00E+05	1200	low power	1	65.7870	
1.00E+05	4800		1	66.1363	
2.50E+06	60		1	2.1957	
2.50E+06	600	low noise	1	13.5365	

2.50E+06	600	low power	1	71.7638
2.50E+06	1200	low noise	1	13.1928
2.50E+06	1200	low power	1	70.8079
2.50E+06	4800		1	66.9399

Table 3.



Figure 14.

Figure 15 shows a comparison of the power consumption in low power versus low noise mode at 600 and 1200 sps. A plot of power consumption versus output data rate gives a comparison of power consumption for a 2.4576 MHz clock frequency and a 4.9152 MHz clock frequency in Figure 16. The complete power consumption data can be found in Table 3. The power consumption was calculated by measuring the voltage drop across two 10 Ω resistors, R7 and R8, in order to find the total current and the analog current consumed by the device. This current was multiplied by the input voltage, V_{dd} = 5 V, to find the power consumption.



Figure 15.



Figure 16.

MAX1402 Power Consumption								
4/28/2008	20 kHz	input signal						
				Total		Analog		
Clock Freq. (MHz)	Data Rate (sps)	Mode	V Total R8 (mV)	V Analog R7 (mV)	Current (mA)	Power (mW)	Current (mA)	Power (mW)
1.024	80	16 bit	8.90	4.20	0.89	4.45	0.42	2.10
1.024	400	low noise	9.00	4.20	0.90	4.50	0.42	2.10
1.024	400	low power	6.50	2.50	0.65	3.25	0.25	1.25
1.024	800	low noise	24.20	17.70	2.42	12.10	1.77	8.85
1.024	800	low power	9.10	4.20	0.91	4.55	0.42	2.10
1.024	1600	N/A	24.40	17.70	2.44	12.20	1.77	8.85
2.048	80	16 bit	8.00	4.10	0.80	4.00	0.41	2.05
2.048	400	low noise	8.00	4.10	0.80	4.00	0.41	2.05
2.048	400	low power	6.00	2.40	0.60	3.00	0.24	1.20
2.048	800	low noise	22.30	17.50	2.23	11.15	1.75	8.75
2.048	800	low power	8.10	4.10	0.81	4.05	0.41	2.05
2.048	1600	N/A	22.30	17.50	2.23	11.15	1.75	8.75
2.4576	60	16 bit opt power	5.90	2.40	0.59	2.95	0.24	1.20
2.4576	600	low noise	8.10	4.10	0.81	4.05	0.41	2.05
2.4576	600	low power	6.00	2.40	0.60	3.00	0.24	1.20
2.4576	1200	low noise	22.30	17.50	2.23	11.15	1.75	8.75
2.4576	1200	low power	8.10	4.10	0.81	4.05	0.41	2.05
2.4576	4800	N/A	24.40	17.70	2.44	12.20	1.77	8.85
4.9152	60	16 bit opt power	5.70	2.40	0.57	2.85	0.24	1.20
4.9152	600	low noise	7.50	4.00	0.75	3.75	0.40	2.00
4.9152	600	low power	5.70	2.40	0.57	2.85	0.24	1.20
4.9152	1200	low noise	21.30	17.40	2.13	10.65	1.74	8.70
4.9152	1200	low power	7.60	4.00	0.76	3.80	0.40	2.00
4.9152	4800	N/A	22.40	17.50	2.24	11.20	1.75	8.75
					Min	2.85	Min	1.20
					Max	12.2	Max	8.85

Table 4.

Conclusions:

The tests using the load cell for an input show that the inherent noise in the MAX1402 at low output data rates is minimal. Changing the input voltage gain and buffering at higher data output rates only caused a slight increase in the noise levels; however, this is inconclusive due to the fact that there was no input "signal" in this case. At lower data rates, the input buffering and especially the higher voltage gain substantially increased the noise levels. This could be due to the fact that the circuitry for the input buffering and voltage gain adds some noise into the system.

The function generator tests show that the MAX1402 effectively filters out high frequency noise at all frequencies that are at least twice the data output rate, other than the sampling frequency and its harmonics. Increasing the input voltage gain had little effect on the output noise levels. For the 60 sps and 600 sps low noise configurations,

the noise levels were only slightly higher for the 16x and 128x gain than those of the 1x gain. For the 600 sps low power configuration, the noise levels for the 16x and 128x gain were lower than those of the 1x gain.

The power consumption tests give an idea of the MAX1402 total power consumption and its relationship to different configuration variables. The total power usage of the chip ranges from 2.85 to 12.2 mW depending on configuration. The output data rate, digital filtering, and clock frequency all had an effect on the power consumption. As data rate increased, so did power consumption. Turning on more digital filtering, the socalled low noise mode, consumed more power than turning it off in low power mode. Figure 14 shows that the low power mode for both 600 and 1200 sps output data rates had comparable noise and power levels. The low noise mode for 600 sps, however, consumed much less power than low noise mode at 1200 sps. though they had comparable noise levels. This plot shows that the 60 sps and the low noise 600 sps configurations best optimize power and noise. The comparison between clock frequency and power consumption in Figure 16 shows that operation at the 4.9152 MHz clock frequency consumes slightly less power for the same data output configuration than the 2.4576 MHz clock frequency.

Execution of the testing has also revealed that configuring the MAX1402 is very straightforward. The function of all of the registers is clearly explained in the data sheet and they can be easily configured to change the mode of the device as desired.

Discussion:

Overall, the MAX1402 analog to digital converter is an excellent option for acquiring data from various analog sensors on a robot. Since noise levels and power consumption increase substantially with increasing data rate, it is important to sample the data at as low a rate as possible, based on how guickly the analog input signal is changing. Most high frequency noise above 2x of the output data rate will be filtered by the chip, except for at frequencies corresponding to the sampling rate. Shielding the input wires or using a simple low-pass filter could remedy this issue if there is noise present at the sampling frequency. Care must be taken to eliminate noise frequencies less than 2x of the output data rate, because these will appear on the output as a signal.

References:

- 1. Kester, W. (Ed.). (2005). Data Conversion Handbook. Boston, MA: Elsevier.
- 2. Maxim. MAX1402 Datasheet, Rev. 2. http://datasheets.maximic.com/en/ds/MAX1402.pdf.
- 3. Maxim. MAX1402 Evaluation Kit/Evaluation System Datasheet, Rev. 2. http://datasheets.maxim-ic.com/en/ds/MAX1400EVKIT-MAX1402EVKIT.pdf.
- 4. Maxim. Application Note 1108: Understanding Single-Ended, Pseudo-Differential and Fully-Differential ADC Inputs. http://www.maximic.com/appnotes.cfm/an pk/1108.
- 5. Maxim. Application Note 1870: Demystifying Sigma-Delta ADCs. http://www.maximic.com/appnotes.cfm/appnote number/1870.

Appendix:

%Data Analysis Program for ADC Noise Testing %Nicole Rodia %April 30, 2008

function [] = data_analysis()
[FileName,PathName] = uigetfile('*.*', 'MultiSelect','on');

%clear variables clear datapoints %close plot windows close all

%get output file name outfile = input('Please enter an output file name: ', 's');

```
%open output file
fout = fopen(outfile, 'wt');
```

```
%handle case of one input file
if(ischar(FileName))
FileName = cellstr(FileName);
end
```

```
%go through each file in the list for k=1:length(FileName)
```

```
%read the entire file in, just as strings
fid = fopen([PathName char(FileName(k))], 'r');
index=0;
running = 1;
while running
    index = index+1;
    input_text(index) = textscan(fid, '%s', 1, 'delimiter', ");
    if isempty(input_text{index})
        running = 0;
    end
end %while running
fclose(fid); %done reading the file in
```

```
%go through the lines and process them
index=0;
for i=1:length(input_text)
tmpstring = char(input_text{i});
%make sure the line is not zero length
if (length(tmpstring))
```

```
%see if the line contains a valid number
     val = str2num(tmpstring);
     %if val is zero length, then it wasn't a number
     if (length(val))
        %it was a number, so add it to the datapoints
       index=index+1;
       datapoints(k, index) = val;
     end
  end
end %done reading in lines
%print test info
fprintf(fout, '%s\n', char(FileName(k)));
%calculate overall stats
num pts = length(datapoints(k, :));
avg = mean(datapoints(k, :));
sdev = std(datapoints(k, :));
vare = var(datapoints(k, :));
noise = (max(datapoints(k, :)) - min(datapoints(k, :))) / 2^18;
mVrange = (max(datapoints(k, :)) - min(datapoints(k, :))) * 0.02;
%output stats
fprintf(fout,'\nTotal data points: %d\n',num pts);
fprintf(fout,'Average of data points: %g\n',avg);
fprintf(fout, 'Standard Deviation of data points: %g\n',sdev);
fprintf(fout,'Variance of data points: %g\n',vare);
fprintf(fout,'Percentage Noise of data points: %g%%\n',noise*100);
```

fprintf(fout,'Range of data points: %g mV\n\n\n',mVrange);

end %done going through files

%close the output file fclose(fout);