ELEX 7820

Lab 2 - ADC and real time emulation mode

Prepared by:	Prep Date:
Peter Li	October 9, 2017

1 Introduction

Over the course of this lab we explored the Analog to Digital Converter on the TI TMS320F28027 Piccolo Microcontroller, as well as the Real-Time Emulation mode. We recorded part Specific Information such as PARTID, CLASSID, REVID, ADCREFTRIM, ADCOFFTRIM, the temperature sensor slope, the temeeerature sensor offset. And gained expereince using the on-chip A-D converter by writing code to initialize the A-D.

The other major feature we explored was the Real-Time Emulation mode; We observed global variables in the expressions window update on-the-fly. observed the graph of signal vs time;

2 Part Specific information

What are the addresses of these registers - where can you look to find them? What do the values of the above registers signify?

When a project is built and the program is runnning on the board, we are able to see various information regarding the device itself in the registers. These are as follows

- **PARTID:** located at 0x003D7FFF with a value of 0xCF
- **CLASSID:** located at 0x0882 with a value of 0xCF
- **REVID:** located at 0x0883 with a value of 0x01

These register signifiy important information about the microcontroller, namely the the particiular identifying part number, the type of microcontroller that it is, and the version of the microcontroller that it is. These pieces of information can be found by looking in the Texas instruments Datasheet sprs523k (specifically page 53-54).

3 ADC registers

Looking at ADCREFTRIM and ADCOFFTRIM we are able to get the following values these are teh ADC registers that are the reference trim value and the offset trim value for the ADC.

- **ADCREFTRIM:** 10136
- **ADCOFFTRIM:** 7

ADCREFTRIM is further broken into several bit fields that have the following values

- **BG_FINE_TRIM:** 24
- **BG_COARSE_TRIM:** 12
- **EXTREF_FINE_TRIM:** 19

What are the addresses of these registers - where can you look to find them? What do the values of the above registers signify?

These are the values of the trim for the ADC

What are the addresses of these registers - where can you look to find them? What do the values of the above registers signify?

With The offset and the slope of the temperature sensor can be acquired by dereferencing 0x3d7e80 and 0x3d7e83 respectively. we see a value of $0.174286\frac{C^{\circ}}{LSB}$ for the temp slop (when converted into Q15) and an offset value of 1621. According to the datasheet, typical values are 0.18 deg/LSB and 1700 for the offset these values are reasonably close. We can convert temp_slope into floating point format by performing a Q to Float conversion on it (convert int value to float, then divide by $2^{1}5$)

4 Configuring the ADC

We configred the adc by first powering up the ADC's analog circuitry, bandgap and reference buffer. We also elected to generate the interupt pulse on the end of conversion as opposed to the start of the conversion then enabling the ADC. After waiting 1 ms for the ADC to be powered on, we configured it to measure the temperature sensor.

Where can you look to read up on the meaning of these registers?

The registers in particular that we needed to configure, had two places where info needed to be pulled from. We can see the actual definition of the registers in the DSP2902x_ADC.h and we can see detailed information regarding the functionality of the Microcontroller in the TI ADC reference datasheet http://www.ti.com/lit/ds/sprs523k/sprs523k.pdf

With all of the registers configured we were able to get a tempearture reading of the temperature sensor s

(x)= temp_offset	int	1622	0x00000406@Data
(x)= temp_slope	int	0.174286 (Q-Value(15))	0x00000407@Data
(x)= temp_reading	int	1803	0x00000405@Data
(x)= (temp_reading-temp_offset)*(0.174286)	double	31.545766	
(x)= (double)(temp_slope)	double	5711.0	

Figure 1: Temperature Reading

With Real time emulation we are able to view the continuously updating value without a need for breakpoints or suspension of the processor.

With this we can now use graphs to view data from the micro.

Given the measured case temperature, how does the measured junction temperature compare to the theoretical junction temperature?

From the resulting graph we can estimate a peak to peak variation of (6)*0.174 = 1.04 degrees Celcius variation.

However you can see where i put my cold finger on the package of the TI TMS320F28027 Piccolo Microcontroller, where the temperature dropped to (1785-1622)*0.174 = 28.362 degrees.

It is mentioned in the documentation by TI that there is a chance that the first reading from the sensor may be inaccurate. But the second read from the sensor is. According to the ADC documentation, it provides an example where the temperature sensor is read by two adc channels and only the result of the second channel is stored. This should result in a significantly more accurate answer.

How does the peak-to-peak variation in the reading versus time compare to the previous code where only one conversion is done? To better measure this, Export the Graph data to Excel and compute the rms values for

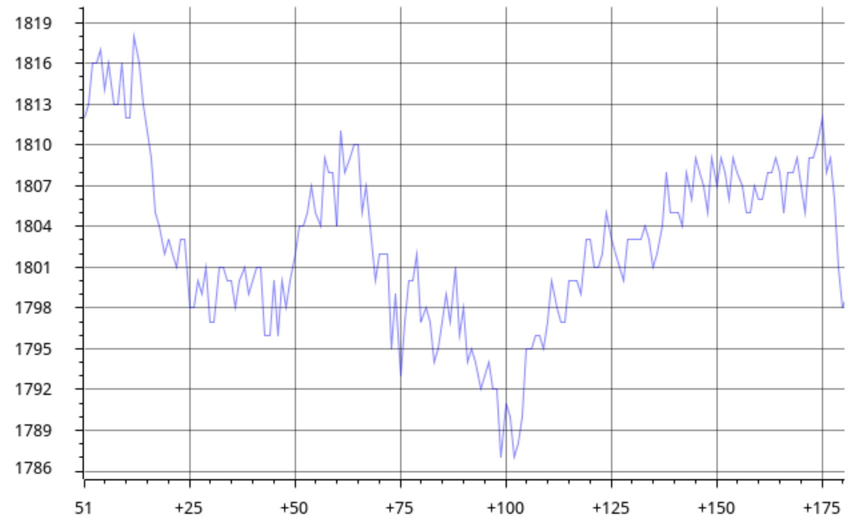

Figure 2: Temperature Reading Graph

the two cases.

Recording the values of Two channel's measurement, we found an RMS error value of 0.1327005185. Recording One channel's measurement we Found an RMS 0.220961796. So in this case it does appear that there is a better accuracy and less jitter, but not by much.

We can apply the calibration data to the temperature reading and produce a human readable result. We can do this by first applying the offset calibration then multiplying by the slope value to convert to a human readable Celsius value.

You might have to think carefully about how to convert your integer value to a floating-point value in your C calculation statement. Is the Celsius value in the right ballpark for the physical temperature?

```
temp_celcius = (temp_reading - temp_offset) * ((float)temp_slope/32768)
```

32768 is 2^{15} which is the factor needed to convert temp_slope from fixed point Q15 to floating point.

Is the Celcius value in the right ballpark for the physical temperature

We get a value hovering around 1834, which when combined with an offset of 1622 and a slope value of 5711 we get a value of 36.9486 degrees celcius which is in the correct ballpark.