PIC32MX ADC Sample Time Discrepancy

When developing and testing a new ADC Callback library for the PIC32MX150F128D Microcomputer Chip by Microchip, I discovered a discrepancy between the theoretical calculated sample time and the measured sample time. The difference was so huge that I made a closer investigation.

In this case I used a PIC32MX150F128D processor but I think it’s safe to say that this applies to all PIC32MX processors. Then developing a new ADC library with callback functionality I discovered a rather huge time difference between the theoretical calculated sample time and what I actually did measured with my oscilloscope. First I believed I was making some mistake in my calculations or timing setup. However after done an investigation it is my conclusion that the calculation of the total ADC sample time given in the manuals and data sheets is not the whole truth. Due to that I think this can be of general interest I decided to share my findings with other PIC32 users.

Theoretical Background

The total sample time is the sum of two parts. If we check the PIC32MX Family Reference Manual, section 17. 10 Bit A/D Converter (DS61104D) we can find the following figure:

Figure 17-2: ADC Sample/Conversion Sequence

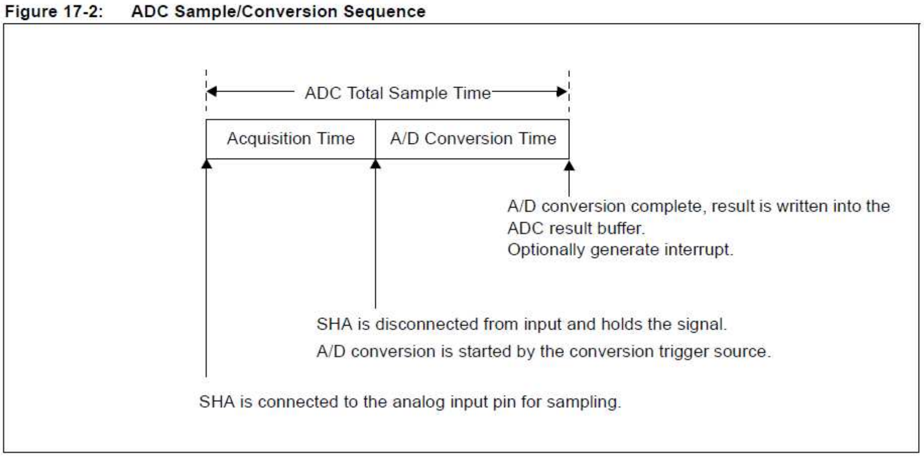

ADC Total Sample Time
Acquisition Time
A/D Conversion Time
A/D conversion complete, result is written into the ADC result buffer.
Optionally generate interrupt.
SHA is disconnected from input and holds the signal.
A/D conversion is started by the conversion trigger source.
SHA is connected to the analog input pin for sampling.

Figure 1, ADC Total sample time

The total sample time is the sum of:

ADCST (ADC Sample Time) = ADCAT (ADC Acquisition Time) + ADCCT (ADC Conversion Time)

During the Acquisition time the ADC connects the sample and hold (SH) amplifier to the source to be measured. This charges a small capacitor, thereafter the SH amplifier disconnects the input and the stored voltage in the capacitor is connected to the Analog to Digital Conversion logic. Then the conversion is complete a flag is raised and optionally an interrupt is generated.

In the case presented in this report the clock to the ADC was taken from the Peripheral clock. In my case the processor was running on 48 MHz and the Peripheral clock divisor was set to one so the peripheral clock was also 48 MHz. According to the manuals and data sheets for the processor we have the following equations that govern the ADC total sample time ADCST.

We have the following equations:

$$PB\_CLKF \;=\frac{SYS\_CLKF}{PB\_DIV} \qquad (1)$$

Where:
SYS_CLKF = System clock frequency (MHz)
PB-DIV = Peripheral clock divisor (#)
PB_CLK F= Peripheral clock frequency (MHz)

$$TPB=\frac{1}{PB\_CLKF} \qquad (2)$$

Where:
TPB = Peripheral clock period (micro seconds/µS)

$$TAD = 2 * (TPB(ADCS + 1) \qquad (3)$$

Where:
TAD = ADC conversion clock period
ADCS = ADC clock prescaler register, 8 bit wide can be 0-255 in value

$$TACQ = SAMC * TAD \qquad (4)$$

Where:
TACQ = ADC Acquisition Time (micro seconds/µS)
SAMC = Auto Sample register value, can be 1 to 31 (0 is not allowed)

$$TCONV = 12 * TAD \qquad (5)$$

Where:
TCONV = ADC conversion time (Convert analog to digital value)

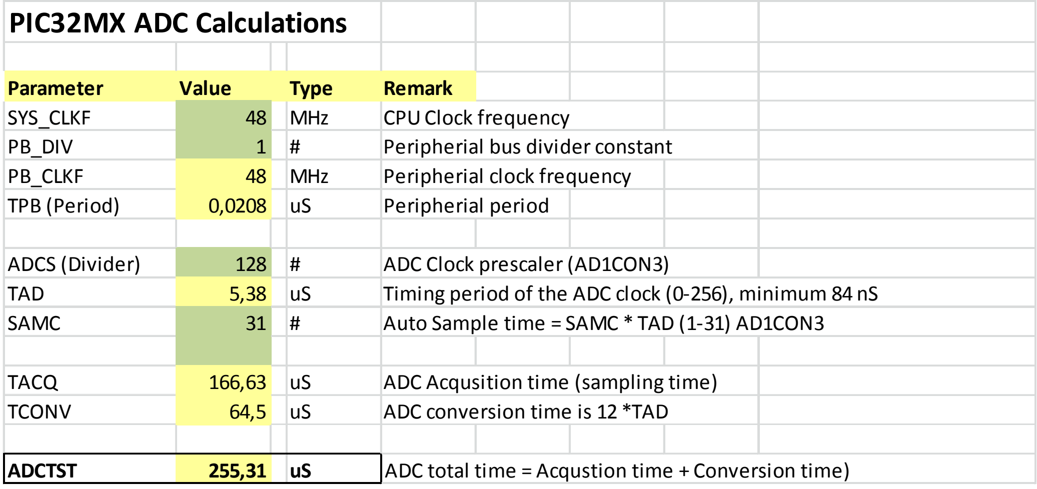
$$ADCTST = TACQ + \; TCONV \qquad (6)$$

Where:
ADCTST = ADC Total sample time (from start to finish) (micro seconds/µS)

Doing some substation we get:

$$ADCTST = (SAMC * TAD) + (12 * TAD) \qquad (7)$$

Calculations
Now if we take the equations above and put them into an Excel sheet to calculate the theoretical values and also measure this to confirm the timing we will get the following, se figure 2.

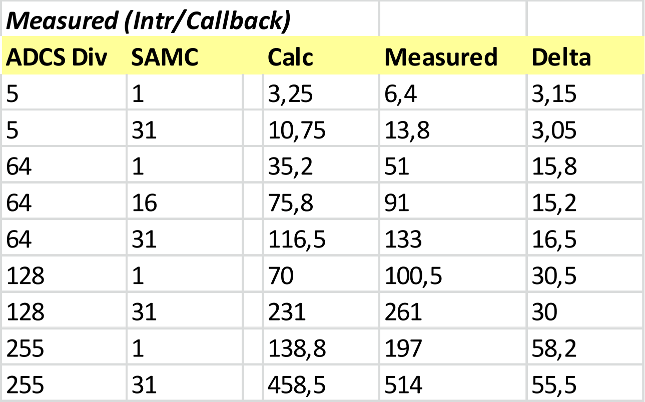


PIC32MX ADC Calculations			
Parameter	**Value**	**Type**	**Remark**
SYS_CLKF	48	MHz	CPU Clock frequency
PB_DIV	1	#	Peripherial bus divider constant
PB_CLKF	48	MHz	Peripherial clock frequency
TPB (Period)	0,0208	uS	Peripherial period
ADCS (Divider)	128	#	ADC Clock prescaler (AD1CON3)
TAD	5,38	uS	Timing period of the ADC clock (0-256), minimum 84 nS
SAMC	31	#	Auto Sample time = SAMC * TAD (1-31) AD1CON3
TACQ	166,63	uS	ADC Acqusition time (sampling time)
TCONV	64,5	uS	ADC conversion time is 12 *TAD
ADCTST	**255,31**	**uS**	ADC total time = Acqustion time + Conversion time)

Figure 2, Excel calculation

Measurements

In figure 2 we can calculate the expected theoretical ADCTST (ADC Total Sampling Time) for different values of the parameters like System clock, Peripheral divisor, value of ADCS (ADC prescaler) and SAMC. Now if we make a table for different values of ADCS we will get the following table, se figure 3.



Measured (Intr/Callback)				
ADCS Div	**SAMC**	**Calc**	**Measured**	**Delta**
5	1	3,25	6,4	3,15
5	31	10,75	13,8	3,05
64	1	35,2	51	15,8
64	16	75,8	91	15,2
64	31	116,5	133	16,5
128	1	70	100,5	30,5
128	31	231	261	30
255	1	138,8	197	58,2
255	31	458,5	514	55,5

Figure 3, calculated and measured values

As can be seen in figure 3 we have for example a time difference of about 55 µS for an ADCS value of 255 and a SAMC value of 31. This corresponds to about 12 %. This is much more than expected. A graph of the result can be seen in figure 4.

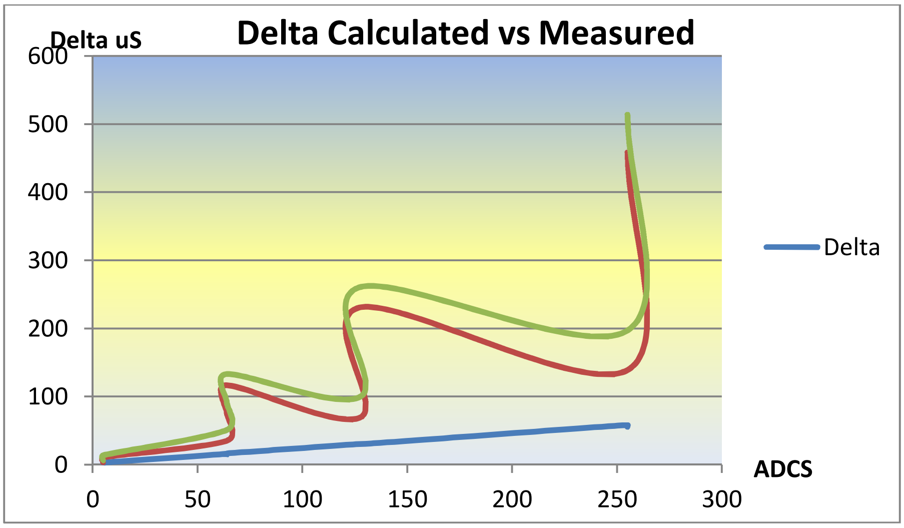


Figure 4, Graph of Calculated vs. Measured

At first I expected that there was something wrong with my code or system clock frequency. I did check and rechecked but couldn't find anything wrong. An interesting fact was that the delta in the timing was linear to the ADCS value, e.g. TAD. Here is an oscilloscope picture of the measurement, see figure 5.

I did rewrite my code to be as simple as possible using a polling method to minimize any code overhead in the measurement. But even so I ended up with about the same discrepancy between calculation and measurement.

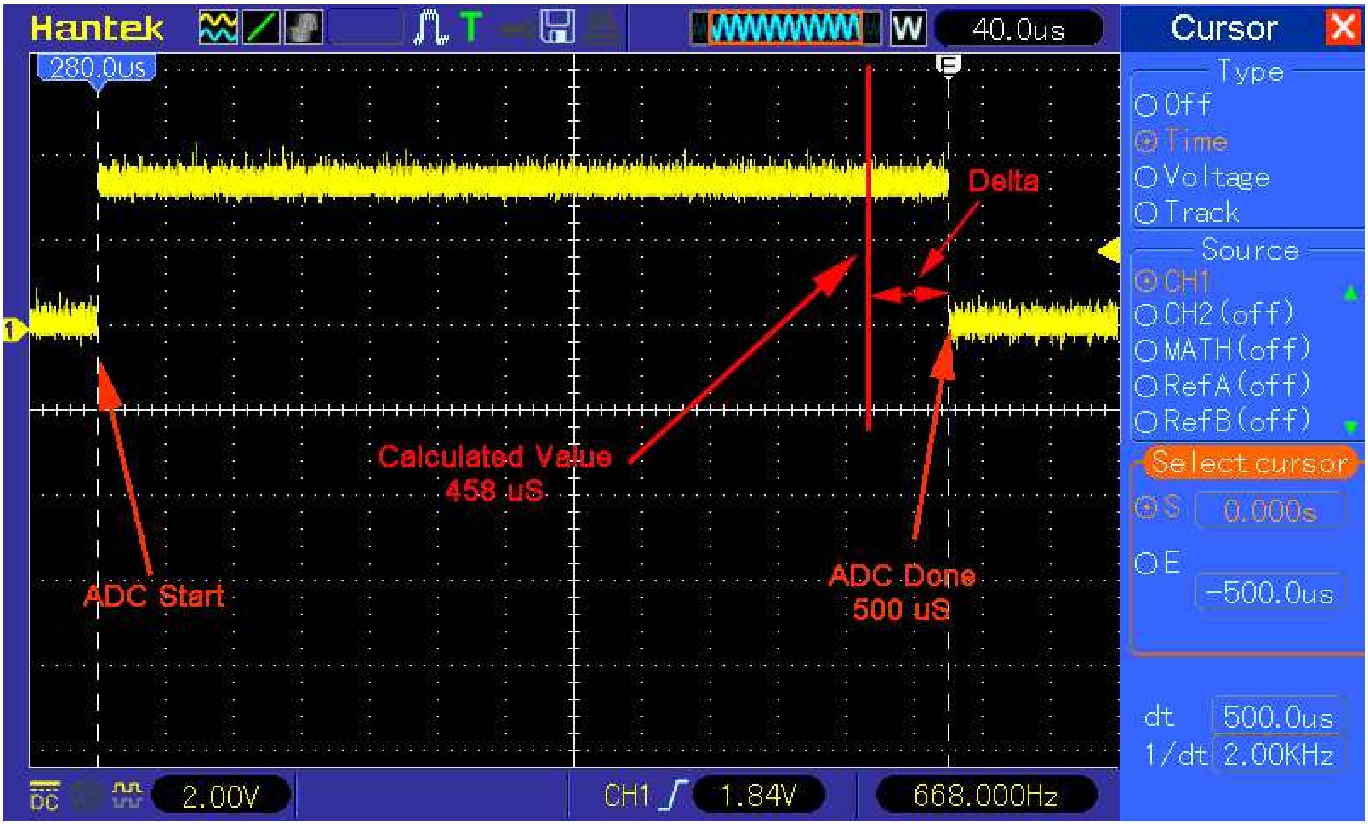


Figure 5, Oscilloscope measurement

Explanations and solution

At first I thought that it could be an "errata" problem but I could not find any hint or explanation there. I did an Internet search; I could find some leads but no explanation. I downloaded the latest documentation and read it in deep and carefully. In the section "29.0 ELECTRICAL CHARACTERISTICS" in the manual "PIC32MX1XX/2XX" (DS60001168F-page 275) I could find some leads. There are some timing parameters that influences the ADC total sample time. In figure 6 I have summarized the information.

1. It takes 1 TAD to discharge the sample and hold capacitor
2. There is an delay of up to 1.5 TAD before the sampling starts after setting the start sample bit
3. There is a Conversion start delay of up to 1 TAD
4. There is a Buffer transfer delay of 1 TAD (check timing diagram)

	TAD's
Discharge 1 TAD	1
ADC Samp Start Dly (AD61)	1,5
Buff Transfer Dly	1
Conversion Start Dly (AD60)	1
Summa:	**4,5**

Figure 6, ADC delays

So we should add an delay to and rewrite equation (7). We have the following

$$TDLY = (TDIS + AD61 + TBUFF + AD60) * TAD \tag{8}$$

Where:

Values are according to figure 6. So we get:

$$TDLY = 4.5 * TAD \tag{9}$$

So the total ADC Total sample time (ADCTST), from start to finish is:

$$ADCTST = (SAMC * TAD) + (12 * TAD) + TDLY \tag{10}$$

Or

$$ADCTST = (SAMC * TAD) + (12 * TAD) + 4.5 * TAD \tag{11}$$

Or (as pointed out by ric/Microchip Forum):

$$\boxed{ADCTST = (SAMC + 16.5) * TAD \qquad (12)}$$

Confirmation

Now if we do a new calculation with the revised equation (11) and do a measurement to get the difference we get the table in figure 7.

ADCS Div	SAMC	TAD	TACQ	TCONV	TDLY	ADCTST	Measured	Delta
5	1	0,25	0,3	3,0	1,1	4,4	5,2	0,8
5	16	0,25	4,0	3,0	1,1	8,1	9,1	1,0
5	31	0,25	7,8	3,0	1,1	11,9	12,7	0,8
64	1	2,71	2,7	32,5	12,2	47,4	48,0	0,6
64	16	2,71	43,3	32,5	12,2	88,0	88,0	0,0
64	31	2,71	84,0	32,5	12,2	128,6	128,5	-0,1
128	1	5,38	5,4	64,5	24,2	94,1	94,5	0,4
128	16	5,38	86,0	64,5	24,2	174,7	175,0	0,3
128	31	5,38	166,6	64,5	24,2	255,3	255,0	-0,3
254	31	10,63	329,4	127,5	47,8	504,7	506	1,3

Figure 7, Calculation vs. measurement, revised equation

The difference between the calculated value and the measured value are now small and reasonable. There is a small difference due to measurement errors and the execution time of the code loop, but they are negligible. Bottom line is that equation (12) seems to be the correct way of calculating the total sampling time.

It interesting to note, that obviously, the ADC-circuit uses the TAD for the internal work. As the ADCS value increases and the TAD becomes bigger this has a notable effect on the total conversion time of the ADC.

/BG 2015-05-03 23:40