

# INTERLEAVED ADC MODE ON STM32F1

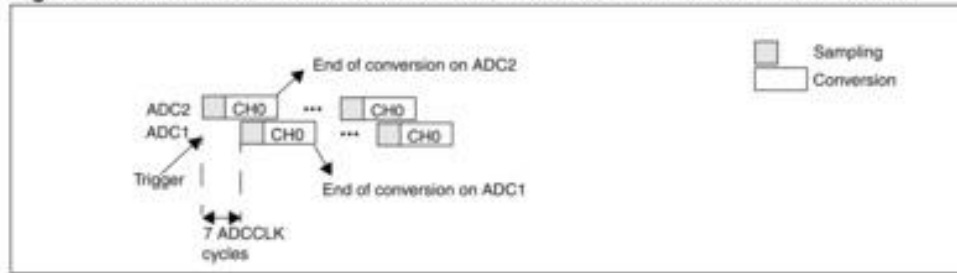
Another feature of VESNA's microcontroller I discovered recently.

STM32F103's datasheet says that the chip includes three analog-to-digital converters. So far I never used more than one and while I never really thought about it, I kind of assumed that they are not really three independent hardware components.

As it turns out they really can work independently and even better, you can use two of them in tandem to achieve twice the sample rate of a single one. The documentation calls this *fast interleaved dual ADC mode*. Interestingly, this is something STMicroelectronics advertises on the first page of datasheets for their signal-processing Cortex-M4 series. But for Cortex-M3 line you have to dig through the reference manual to find a mention of this mode and even then there's not much specific information on how to use it.

In interleaved mode two converters work simultaneously with half of the conversion period out of phase. This means that first converter provides even samples while the second provides odd samples, as this figure from the manual shows:

Figure 32. Fast interleaved mode on 1 channel in continuous conversion mode



Regarding practicalities that are left out of the manual: both ADC1 and ADC2 must be configured for continuous conversion mode. They have to be set for external triggering (but you can use the *SWSTART* option to simulate it from software through *ADC\_CR2* register). This means that you can't trigger the conversion in the usual way by writing 1 to the *ADON* bit while the converter is already powered up.

The DMA transfer reads only from the ADC1 *DR* data register. You have to set 32-bit word length which will actually read out both ADC1 and ADC2 12-bit samples from the same register in kind of hackish way. Because of this the odd and even samples will have switched places in the DMA buffer. This is not really convenient if you want to process the data further on the microcontroller where you usually need samples ordered by time.

How well does it work? I was kind of skeptical, especially regarding trigger accuracy and calibration agreement between both converters.

I did some preliminary measurements at 2 Msamples/s though and so far it looks very promising. Offset difference between converters appears to be around 0.6 digits. Trigger error is on the order of  $10^{-3}$  times conversion rate which is excellent.

I've created a repository on GitHub that contains the code I used to test this (needs libopenm3). The repository also includes an iPython notebook if anyone wants to repeat the measurements or double-check my calculations.

Source:

[https://www.tablix.org/~avian/blog/archives/2013/10/interleaved\\_adc\\_mode\\_on\\_stm32f1/](https://www.tablix.org/~avian/blog/archives/2013/10/interleaved_adc_mode_on_stm32f1/)