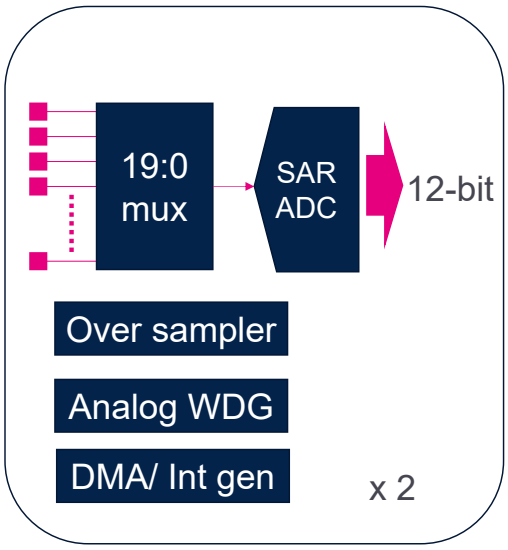
Hello and welcome to this presentation of the STM32 Analog-to-Digital Converter block. It will cover the main features of this block, which is used to convert the external analog voltage-like sensor outputs to digital values for further processing in the digital domain.

Overview



- Provides analog-to-digital conversion
  - Two ADCs with up to 19 input channels
  - 12-bit structure, up to 16 bits with oversampling
  - 5.0 Msample/s max. (12-bit resolution)
  - Three analog watchdogs per ADC
  - DMA request generation
  - Interrupt generation

Application benefits

- Ultra-low power consumption: 244 µA @ 1 Msample/s
- Flexible trigger, data management to offload CPU

life.augmented

The analog-to-digital converters inside STM32 products allow the microcontroller to accept an analog value like a sensor output and convert the signal into the digital domain. There are up to 19 analog inputs available across the two ADCs. The ADC module itself is a 12-bit successive approximation converter with additional oversampling hardware. To have more than 12-bit performance, it is necessary to use oversampling methodology. Under certain conditions, the oversampled output can have a 16-bit result. The sampling speed is 5 mega samples per second for 12-bit resolution. Each ADC module integrates an analog watchdog. The data can be made available either through DMA request or interrupt base. This ADC is designed for low power and high performance. There are numbers of triggering mechanism, and the data management can be configured to minimize the CPU workload.

Key features

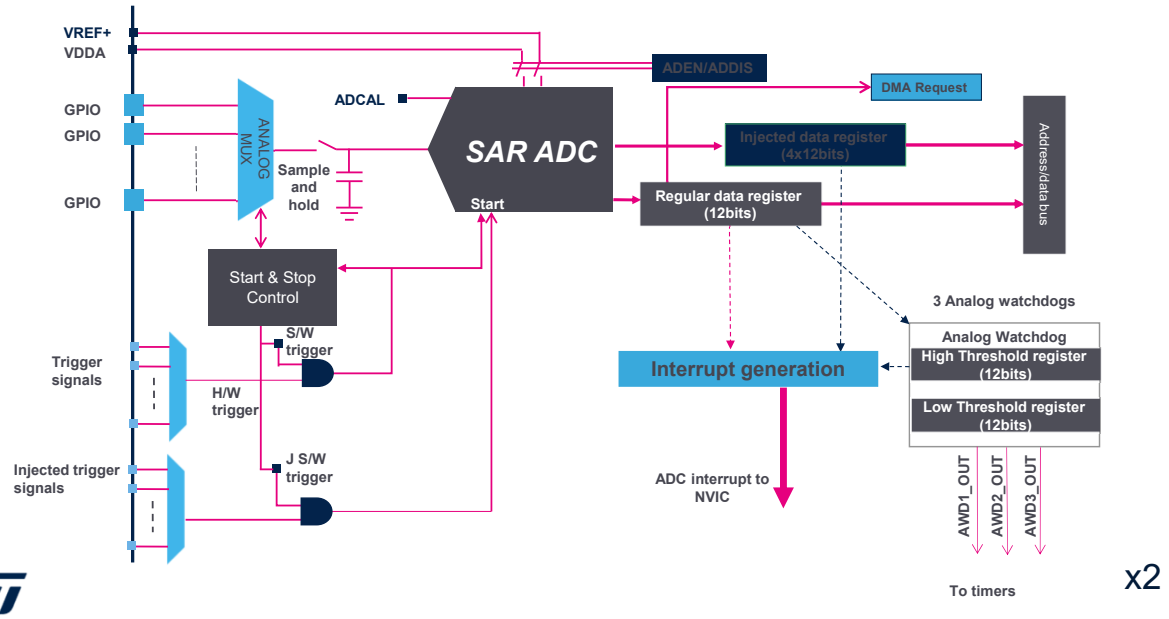
Features	Description
ADC units	2 modules
Input channel	Up to 19 external (GPIOs) or internal channels per ADC
	12-bit successive approximation
Conversion time	200 ns, 5.00 Msamples/s (when f_{ADC_CLK} = 75 MHz, 12 bits)
Functional mode	Single, Continuous, Scan, Discontinuous, or Injected
Triggers	Software or external trigger (by Timers & IOs)
Special functions	Analog watchdogs, Hardware oversampling, Self-calibration
Data processing	Interrupt generation, DMA requests
Low-power modes	Deep power-down, auto delay, power consumption dependent on speed

life.augmented

3

2 analog-to-digital converters are integrated inside STM32MP13 products. The input channel is connected to up to 19 channels capable of converting signals in either Single-end or Differential mode. The ADCs can convert signals in 5 mega samples per second in 12-bit mode. There are several functional modes which will be explained later. There are also several different triggering methods. In order to offload the CPU, the ADC has an analog watchdog for monitoring thresholds. The ADC also offers oversampling to extend the number of bits presented in the final conversion value. For power-sensitive applications, the ADC offers a number of low-power features.

Block diagram

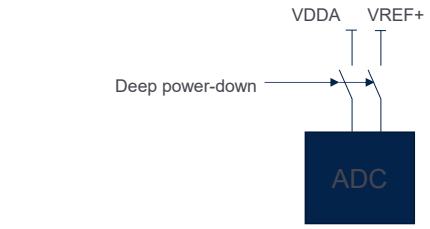


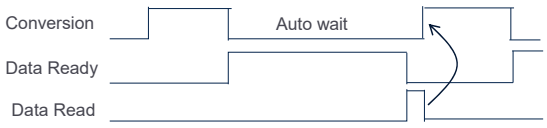
This slide shows the general block diagram for the 2 analog-to-digital converters embedded in the STM32MP13 microprocessor.

Low-power features

Several low-power features are implemented

- Deep power-down mode
  - Internal supply for ADC can be disabled by power switch for leakage reduction
- Auto-delayed conversion
  - ADC can automatically wait until last data is read.
- Power consumption depends on sampling time
  - 810 µA @ 5 Msamples/s, 244 µA @ 1 Msample/s





life.augmented

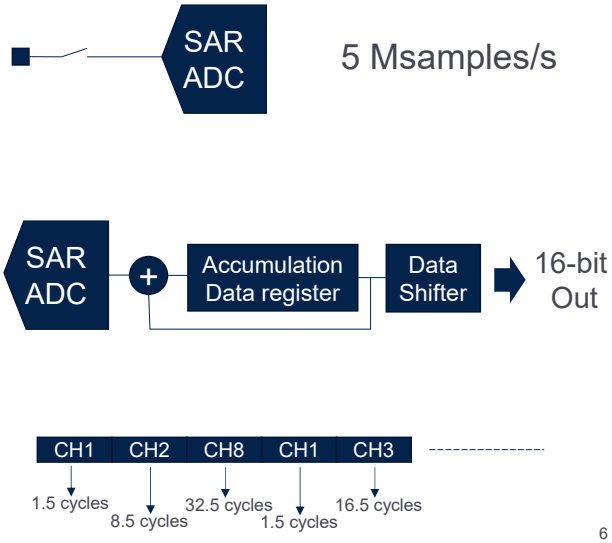
5

The STM32MP1's ADCs support a Deep power-down mode. When the ADC is not used, it can be disconnected by a power switch to further reduce the leakage current. Auto-delayed mode makes the ADC wait until the last conversion data is read before starting the next conversion. This avoids unnecessary conversions and thus reduces power consumption. The power consumption is in function of the sampling frequency. For low sampling rates, the current consumption is reduced almost proportionally.

High performance features

Several high performance features are implemented

- 5 Msamples/s for 75 MHz ADC clock @12-bit conversion
- Hardware oversampling
  - Accumulator and bit shifter can output 16-bit data without CPU support
- Flexible sequencer
- Auto-calibration to reduce offset



life.augmented

6

The ADC supports up to 5 mega samples per second of 12-bit conversion. The ADC includes the oversampling hardware which accumulates data and then divides without CPU help. The oversampler can accommodate from 2 to 256 times samples and right shift from one to eight binary digits. The sequencer allows the user to convert up to 16 channels in any desired order. Also each channel can have a different sampling time. The ADC offers an auto-calibration mechanism for the offset. It is advisable to run calibration on the application if the reference voltage changes by more than 10%.

ADC conversion speeds

Conversion speed is resolution dependent

- ADC needs minimum 2.5_{ADC_CLKs} for sample period and 12.5_{ADC_CLKs} for conversion (12 bits).
- 75 MHz maximum clock with 15-cycle results in 5 Msamples/s
- Speed up by lower resolution
  - 12-bit : 12.5_{ADC_CLKs}(+2.5) => 5 Msamples/s @75MHz
  - 10-bit : 10.5_{ADC_CLKs} (+1.5) => 5.77 Msamples/s @75MHz
  - 8-bit : 8.5_{ADC_CLKs} (+1.5) => 6.82 Msamples/s @75MHz
  - 6-bit : 6.5_{ADC_CLKs} (+1.5) => 8.33 Msamples/s @75MHz

Resolution	$t_{Conversion}$
12 bits	15 Cycles
10 bits	13 Cycles
8 bits	11 Cycles
6 bits	9 Cycles

life.augmented

7

The ADC needs a minimum of 2.5 clock cycles for sampling and 12.5 clock cycles for conversion for 12-bit mode. With a 75 MHz ADC clock, it can achieve 5 mega-samples per second. For higher sampling speed, it is possible to reduce the resolution down to 6 bits in which case the conversion speed can go up to 8.33 mega-samples per second.

Sampling times

Programmable sampling time

- The following sampling times can be selected:
  - 2.5 cycles
  - 6.5 cycles
  - 12.5 cycles
  - 24.5 cycles
  - 47.5 cycles
  - 92.5 cycles
  - 247.5 cycles
  - 640.5 cycles

  Note: Sampling time can vary depends on the conversion mode by ± 0.5 cycle. Please refer to the reference manual.
- If Scan mode is selected, each input channel can have a different sampling time
  - One ADC can scan the different input source with various source impedance.

life.augmented

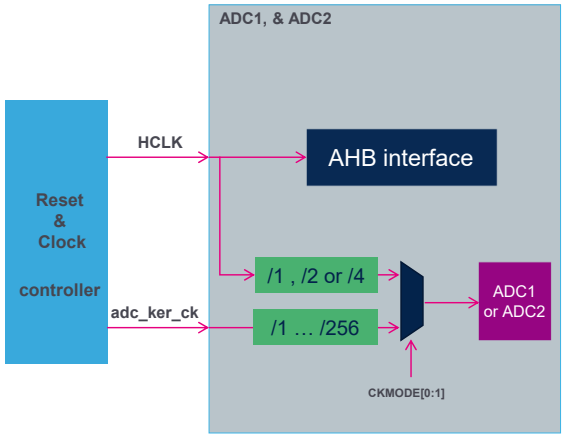
8

The sampling time can be programmed individually for each input channel of the analog-to-digital converters. The sampling times listed in this slide in ADC clock cycles are available. Longer sampling times ensure that signals with a higher impedance are correctly sampled and converted.

ADC clocks

Flexible clock selection

- ADC clock can be selected from
  - AHB clock divided by 1, 2 or 4.
    If a trigger event depends on the AHB clock, the latency between the event and start of conversion is deterministic.
  - Dedicated ADC clock
    Independent and asynchronous to the system clock (AHB). The CPU can run slowly even if the ADC is running full-speed. The adc_ker_ck source can be connected from independent PLL.
  - When two ADCs need to perform ADC conversion at same time it is mandatory to choose same clock source by adc_ker_ck

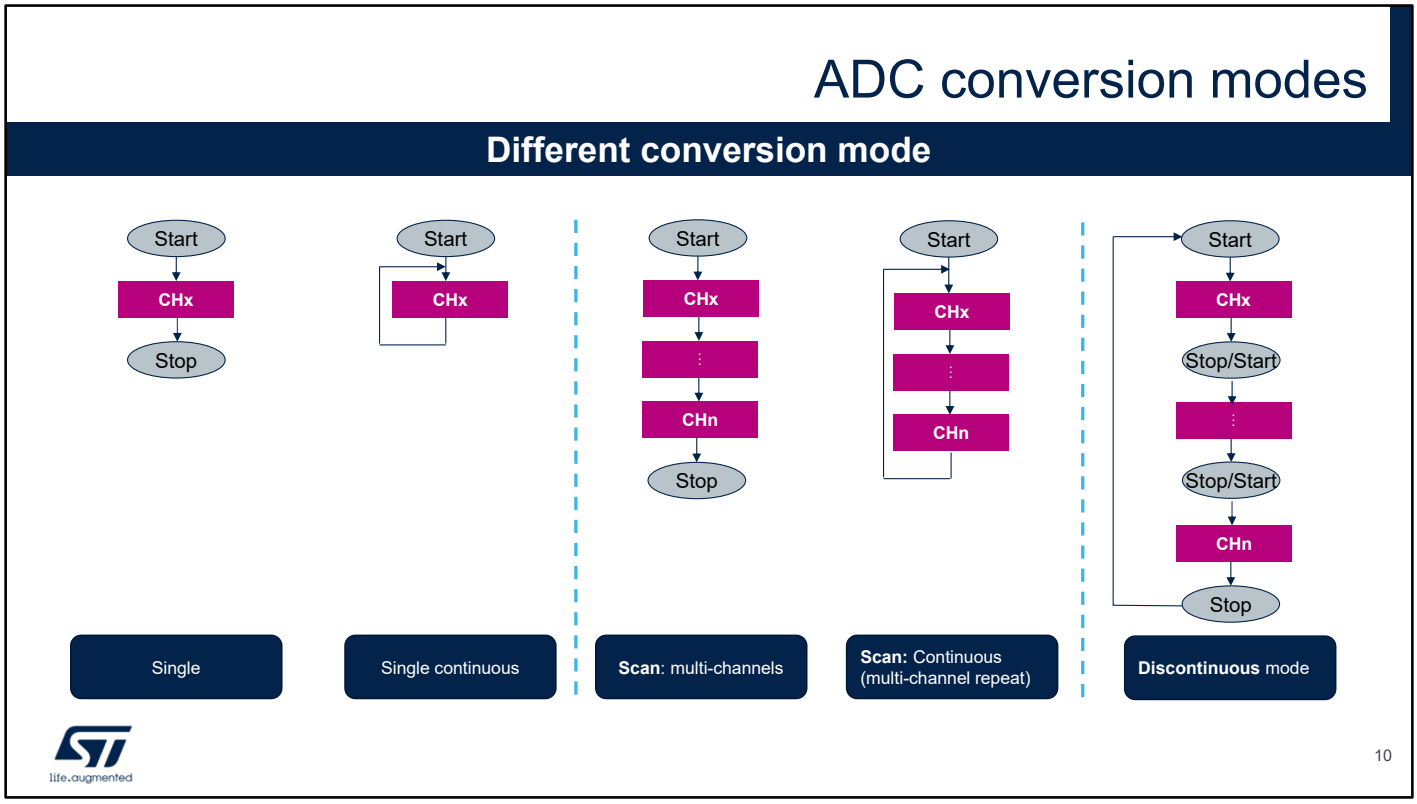


life.augmented

9

The ADCs have a selectable clock source. When the system needs to run synchronously, the AHB clock source is the best selection. If a slow CPU speed is required, but the ADC needs a higher sampling rate, the dedicated ADC clock can be selected. The adc_ker_ck source can be selected from the independent PLL.

ADC conversion modes



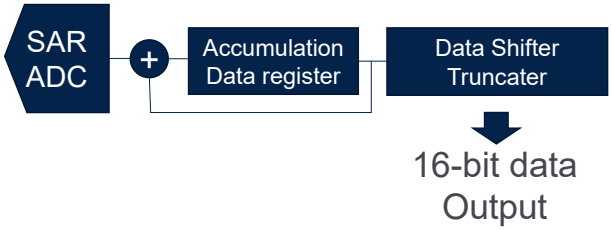
The ADC supports several conversion modes:

- Single mode, which converts only one channel, in Single-shot or Continuous mode.
- Scan mode, which converts a complete set of pre-defined programmed input channels, in Single-shot or Continuous mode.
- Discontinuous mode, which converts only a single channel at each trigger signal from the list of pre-defined programmed input channels.

Hardware oversampling

Data pre-processing to offload the CPU

- Programmable oversampling ratios: x2 to x256
- Programmable data shifter & truncator right shift of 0 to 8 bits.
- Up to 16-bit data width
- Averaging, data rate reduction, SNR improvement, and basic filtering



Oversampling ratio	Output resolution	Equivalent sampling frequency (max.)
x1(none)	12 bits	5 Msamples/s
x4	13 bits	1.25 Msamples/s
x16	14 bits	312.5 Ksamples/s
x256	16 bits	19.5 Ksamples/s

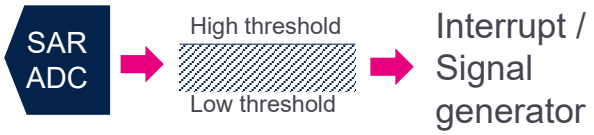
life.augmented

The ADCs support hardware oversampling. They can sample by 2 to 256 times without CPU support. The converted data is accumulated in a register and the output can be processed by the data shifter and the truncator. 12-bit data can be extended to be presented as a 16-bit data register. This functionality can be used as an averaging function or for data rate reduction and signal-to-noise ratio improvement as well as for basic filtering.

Analog watchdog

Reduced software overhead

- Each of the 2 ADCs has three Window comparators
  - One 12-bit analog watchdog can monitor one selected channel or all enabled channels
  - Two 8-bit analog watchdogs can monitor several selected channels
- Each watchdog continuously monitors an over- and/or under-threshold condition, then generates either an interrupt or an external signal, or stops a timer.



life.augmented

Each ADC has an integrated analog watchdog with high and low threshold settings. The ADC conversion value is compared to this threshold window. If the result exceeds the threshold, an interrupt or external signal can be generated or a timer can be immediately stopped without CPU intervention.

Data transfers

Reduced software overhead

- Regular conversion data is stored in a 16-bit data register
  - Software polling, interrupts or DMA requests can be used to move the data
  - The OVERRUN flag is set when previously converted data is overwritten by current data
  - For the analog watchdogs, it is not necessary to process each data. The OVERRUN flag can be disabled.
- Injected conversion data is stored in four 16-bit data registers
  - Injected conversion data is stored in dedicated registers. The regular data sequence can be kept even if injected conversion occurs.

life.augmented

The ADC conversion result is stored in a 16-bit data register. The system can use CPU polling, interrupts or DMA to make use of the conversion data. An overrun flag can be generated if data is not read before the next conversion data is ready. For injected channel conversions, 4 dedicated data registers are available.

Injected conversions

Interruption during ADC conversion

- ADC can accept injected triggers even if a regular conversion is running
  - A trigger will stop the regular conversion then start the injected conversion.
    Up to 4 injected conversions are available by a single trigger.
  - Auto-resume occurs once the injected conversion finishes.
  - Four dedicated 16-bit data registers are available for the injected conversion result.
  - Creates the interrupt, or flags for use by the user's firmware.
  - Queue of injected conversion can be reprogrammable on the fly.

life.augmented

An injected conversion is used to interrupt the regular conversion, then insert up to 4 channel conversions. Once an injected conversion is finished, the regular conversion sequence can be resumed. The injected conversion result is stored in dedicated data registers. Flags and interrupts are available for the end of conversion or end of sequence. The choices for an injected channel can be reprogrammed on the fly. Even if a regular or injected conversion is in progress, you can add a different channel to the queue so that the next injected channel can be different from the previous one.

Interrupts and DMA

Interrupt event	Description	Interrupt event	Description
ADRDY	The ADC is ready to convert	**AWDx**	An analog watchdog threshold breach detection occurs
EOC	The end of regular conversion	**EOSMP**	The end of a sampling phase
EOS	The end of sequence for regular conversion group	**OVR**	A data overrun occurs
JEOC	The end of injected conversion	**JQOVF**	The injected sequence context queue overflows
JEOS	The end of sequence of an injected conversion group		

- DMA requests can be generated after each end of conversion of a channel.

life.augmented

15

Each ADC can generate 9 different interrupts: ADC Ready, end of conversion, end of sequence, end of injected conversion, end of injected sequence, analog watch dog, end of sampling, data overrun and the overflow of the injected sequence context queue.
DMA requests can be generated at each end of conversion when the ADC output data is ready.

Low-power modes

MPU / MCU domain state	Description of the peripheral allocated to the domain
CRun	**Active**.
CSleep	**Active**. Peripheral interrupts cause the device to exit Sleep mode.
Stop + LP Stop	**No operation**. Peripheral registers content is kept.
LPLV Stop	**No operation**. Peripheral registers content is kept.
Standby or Shutdown	**Powered-down**. The peripheral must be reinitialized after exiting Standby mode.

- In Deep power-down mode, the analog part of each ADC is switched off by an on-chip power switch. Calibration data is kept.

life.augmented

The ADCs are active in CRun and CSleep modes. In Stop, LP Stop and LPLV Stop mode, the ADCs are not available, but the contents of their registers are kept. In Standby mode, the ADCs are powered-down and must be reinitialized when returning to a higher power state. There is a Deep power-down mode in each ADC itself which reduces leakage by turning off an on-chip power switch. This is the recommended mode whenever an ADC is not used.

Performance

	Condition	Data (typ.)	Unit
Sampling rate	12-bit mode	5	Msamples/s
	10-bit mode	5.77	Msamples/s
	8-bit mode	6.82	Msamples/s
DNL	(single ended)	±0.75	LSB
INL	12-bit mode	±2	LSB
ENOB	12-bit mode (single ended)	11.0	bits
	12-bit mode (differential)	11.5	bits
Consumption	5 Msamples/s	810	µA
	1 Msamples/s	244	µA

life.augmented

17

The following table shows performance parameters for the ADC. All values are preliminary.

Related peripherals

- Refer to training modules linked to this peripheral, if needed:
  - DMA – Direct memory access controller
  - Interrupts
  - GPIO – General-purpose inputs and outputs
  - RCC – Clock module
  - DAC – Digital-to-analog converter
  - TIM – Timers for triggering interrupts and events
  - DFSDM – Digital filter for sigma delta modulators

life.augmented

These peripherals may need to be specifically configured for correct use with the ADCs. Please refer to the corresponding peripheral training modules for more information.

Features for each individual ADC

ADC features	ADC1	ADC2
Interconnect	MLAHB	MLAHB
Internal channel connection	VREFINT	VSENSE, VREFINT, VDDCORE, VBAT/4, VDDCPU, VDDQ_DDR

life.augmented

The STM32MP1 embeds two ADCs. ADC 1 and ADC 2 are individual instances.

References

- For more details, please refer to the following resources:
  - Application note AN2834: How to get the best ADC accuracy in STM32Fx Series and STM32L1 Series devices
  - Application note AN4073: How to improve ADC accuracy when using STM32F2xx and STM32F4xx microcontrollers
  - Application note AN2668: Improving STM32F1x and STM32L1x ADC resolution by oversampling
  - Application note AN4629: ADC hardware oversampling for microcontrollers of the STM32 L0 and L4 series

life.augmented

Several application notes dedicated to analog-to-digital converters are available. To learn more about ADCs, you can visit a wide range of web pages discussing successive approximation analog-to-digital converters.

Thank you

life.augmented