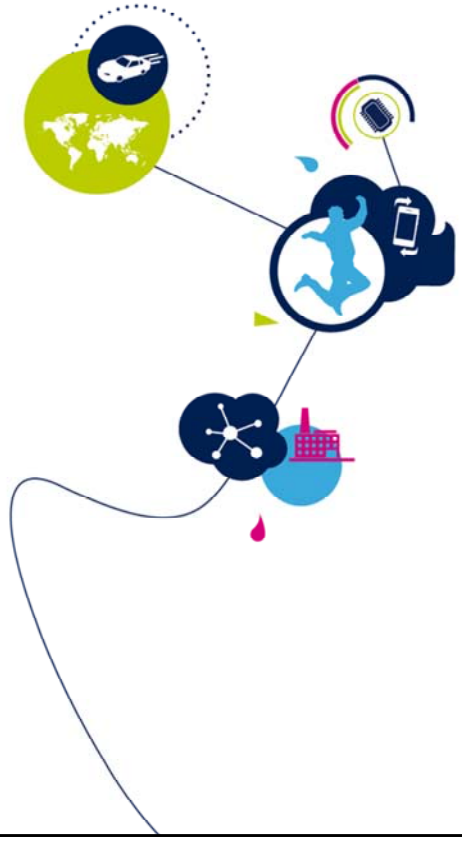
STM32H7 – ADC

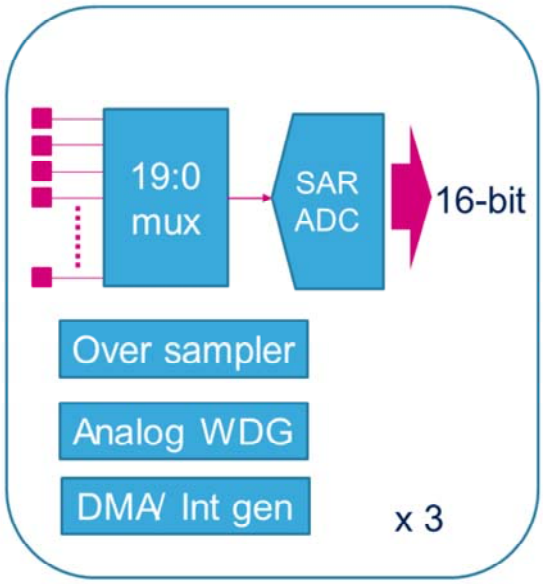
Analog-to-Digital Converter
Revision 1.0



life.augmented

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
  - Three ADCs with up to 20 input channels
  - 16-bit structure, up to 21 bits with oversampling
  - ENOB limited to 14 bits due to noise level
  - 5 Msample/s max. (14-bit resolution)
  - Three analog watchdogs per ADC
  - DMA request generation
  - Interrupt generation

Application benefits

- Ultra-low power consumption: 1.6 mA @ 5 Msample/s
- Flexible trigger, data management to offload CPU

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 20 analog inputs available across the three ADCs. The ADC module itself is a 16-bit successive approximation converter with additional oversampling hardware. Due to the noise level, only 14-bit equivalent performance is achieved. To have more than 16-bit performance, it is necessary to use oversampling methodology. Under certain conditions, the oversampled output can have a 21-bit result. The sampling speed is 5 mega samples per second for 14-bit resolution. Each ADC module integrates an analog watchdog. The data can be made available either through DMA movement or interrupts. This ADC is designed for low power and high performance. There are a number of triggering mechanisms and the data management can be configured to minimize the CPU workload.

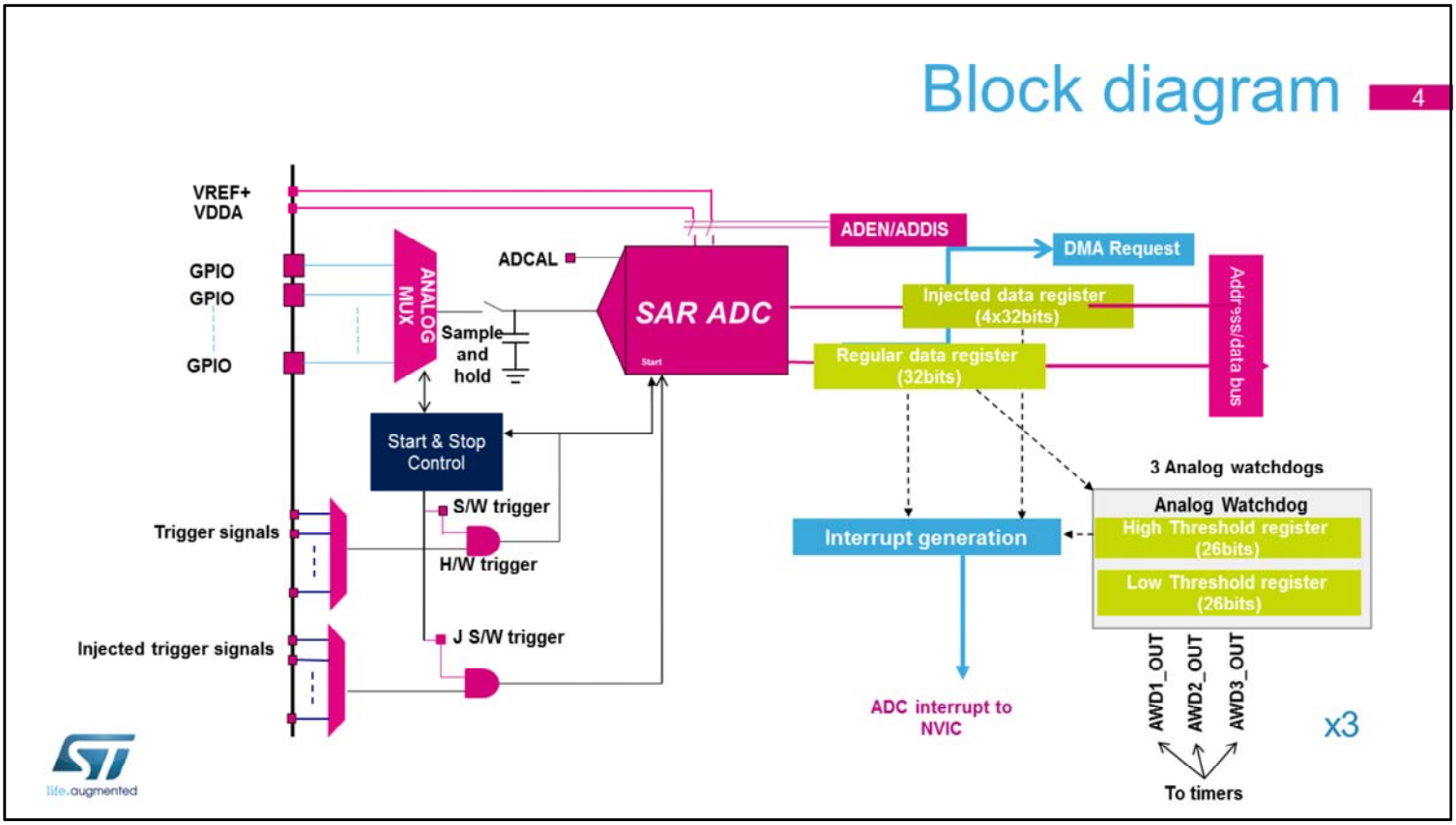
Key features

3

Features	Description
ADC units	3 modules
Input channel	Up to 20 external (GPIOs) or internal channels per ADC
	16-bit successive approximation
Conversion time	200 ns, 5 Msamples/s (when f_{ADC_CLK} = 50 MHz, 14 bits)
Functional mode	Single, Continuous, Scan, Discontinuous, or Injected
Triggers	Software or external trigger (for 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 analog-to-digital converters are integrated inside STM32H7 products. The input channel is connected to up to 20 channels capable of converting signals in either Single-end or Differential mode. The ADCs can convert signals in 5 mega samples per second in 14-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.

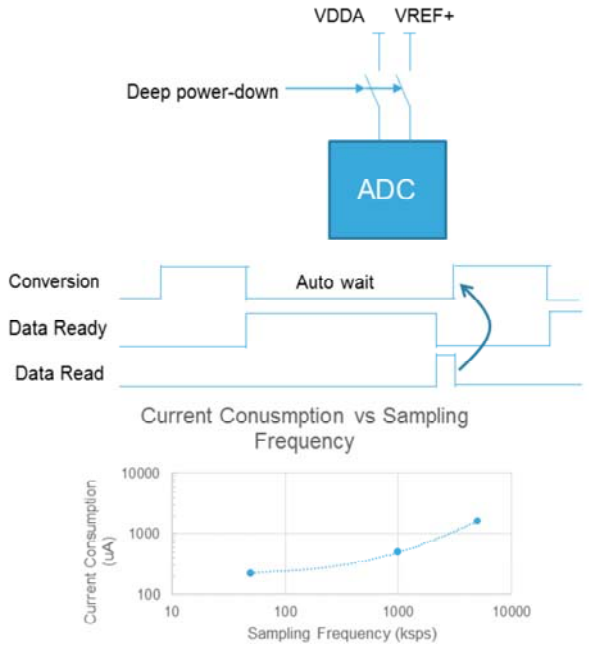


This slide shows the general block diagram for the 3 analog-to-digital converters embedded in the STM32H7.

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
  - 1.6 mA @ 5 Msamples/s, 500 µA @ 1 Msample/s, 220 µA @ 50 ksamples/s

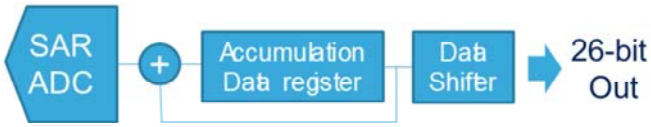


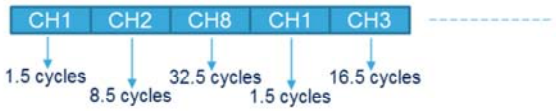
The STM32H7's ADCs support a Deep power-down mode. When the ADC is not used, it can be disconnected by a power switch to further reduce the leakage current. Auto-delayed mode makes the ADC wait until the last conversion data is read before starting the next conversion. This avoids unnecessary conversions 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 50 MHz ADC clock @14-bit conversion
- Interleave mode can support up to 10 Msamples/s
- Hardware oversampling
  - Accumulator and bit shifter can output 26-bit data without CPU support
- Flexible sequencer
- Auto-calibration to reduce offset, better linearity





The ADC supports up to 5 mega samples per second of 14-bit conversion. By using Dual interleaved mode, it can be extended to 10 mega samples par second. The ADC includes the oversampling hardware which accumulates data and then divides without CPU help. The oversampler can accommodate from 2 to 1024 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 period. The ADC offers an auto calibration mechanism for the offset and the linearity. It is recommended to run the calibration on the application if the reference voltage changes more than 10% so this would include emerging from RESET or from a low-power state where the analog voltage supply has been removed and reinstated.

ADC conversion speeds

7

- **Conversion speed is resolution dependent**

- ADC needs minimum 1.5_{ADC_CLKs} for sample period and 7.5_{ADC_CLKs} for conversion (14 bits).
- 50 MHz maximum clock with 10-cycle (it needs 2.5 clk of sample period) results 5M samples/s (14 bits)
- Speed up by low resolution
  - 12-bit : 6.5_{ADC_CLKs}(+1.5) => 6.25 Msamples/s
  - 10-bit : 5.5_{ADC_CLKs} (+1.5) => 7.1 Msamples/s
  - 8-bit : 4.5_{ADC_CLKs} (+1.5) => 8.3 Msamples/s

Resolution	$t_{Conversion}$
16 bits	8.5 Cycles
14 bits	7.5 Cycles
12 bits	6.5 Cycles
10 bits	5.5 Cycles
8 bits	4.5 Cycles

life.augmented

The ADC needs a minimum of 1.5 clock cycles for the sampling and 7.5 clock cycles for conversion for 14-bit mode. With a 50 MHz ADC clock, it can achieve 5 mega samples per second. For higher sampling speed, it is possible to reduce the resolution down to 8 bits then the sampling speed can go up to 8.3 mega samples per second.

Sampling times

Programmable sampling time

- The following sampling times can be selected:
  - 1.5 cycles
  - 2.5 cycles
  - 8.5 cycles
  - 16.5 cycles
  - 32.5 cycles
  - 64.5 cycles
  - 387.5 cycles
  - 810.5 cycles
- 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.

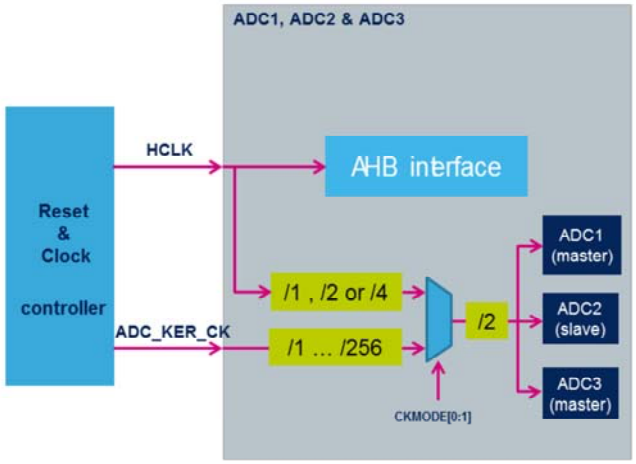
life.augmented

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 sample times ensure that signals having a higher impedance are correctly converted.

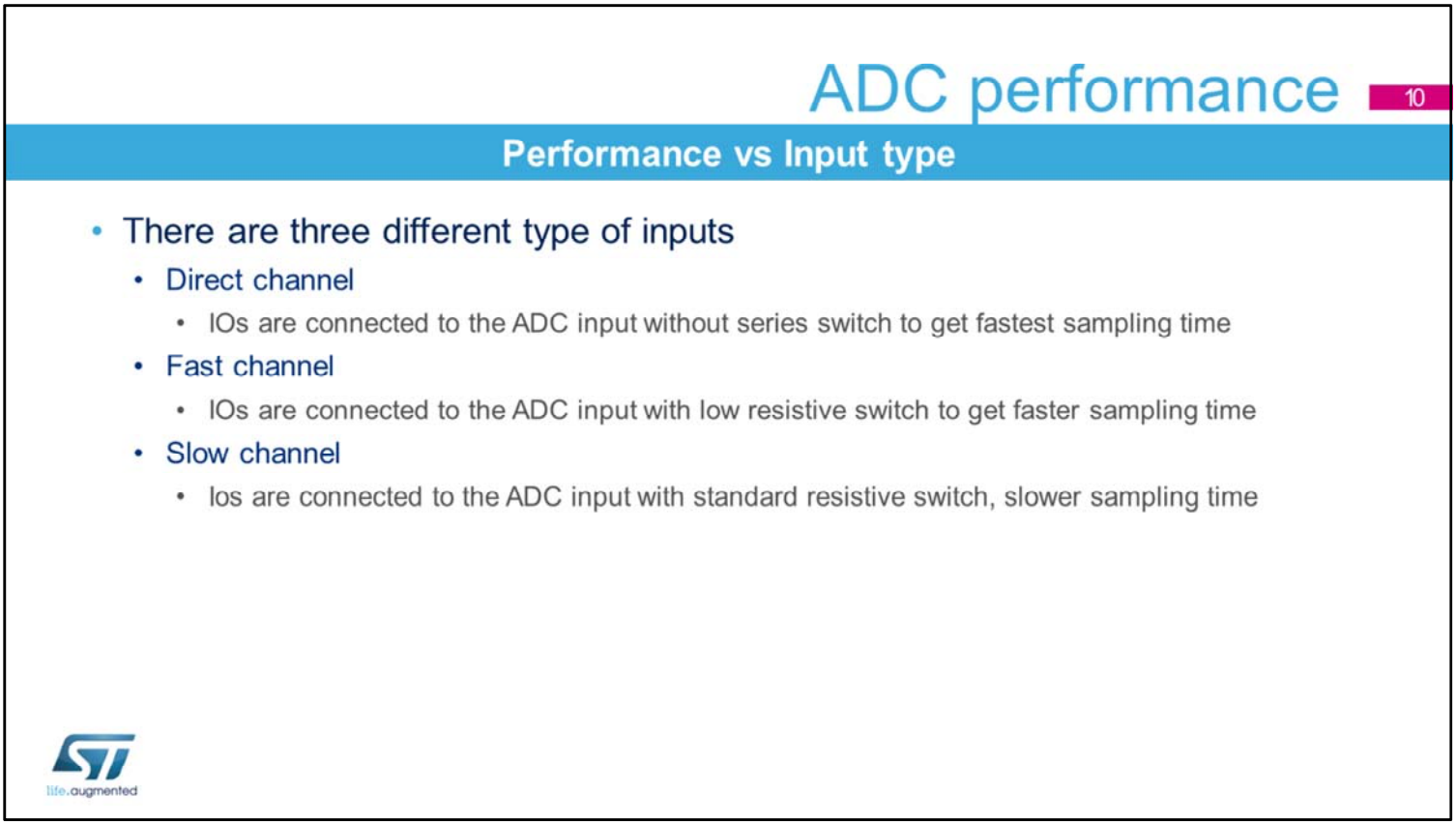
ADC clocks

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.



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.



There are three different type of inputs:

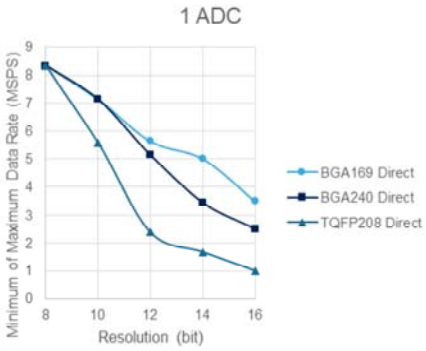
- Direct channel where IOs are connected to the ADC input without series switch to get fastest sampling time
- Fast channel where IOs are connected to the ADC input with low resistive switch to get faster sampling time
- Slow channel where IOs are connected to the ADC input with standard resistive switch, slower sampling time

ADC performance

Performance vs Package

- ADC performance is dependent on the package type and on the number of channels working at the same time.
  - The type of package impacts the ADC performance.
    - Depends on the package, voltage reference stability changed, resulting performance impact.
  - Following order shows better ADC performance
    - BGA package
    - WLCSP package
    - TQFP176 package
    - TQFP208 package

Better performance



The ADC performance is dependent on the package type and on the number of channels working at the same time. The figure shows that the BGA package technology is offering the better performance especially at high resolution.

ADC performance

Performance vs Setting

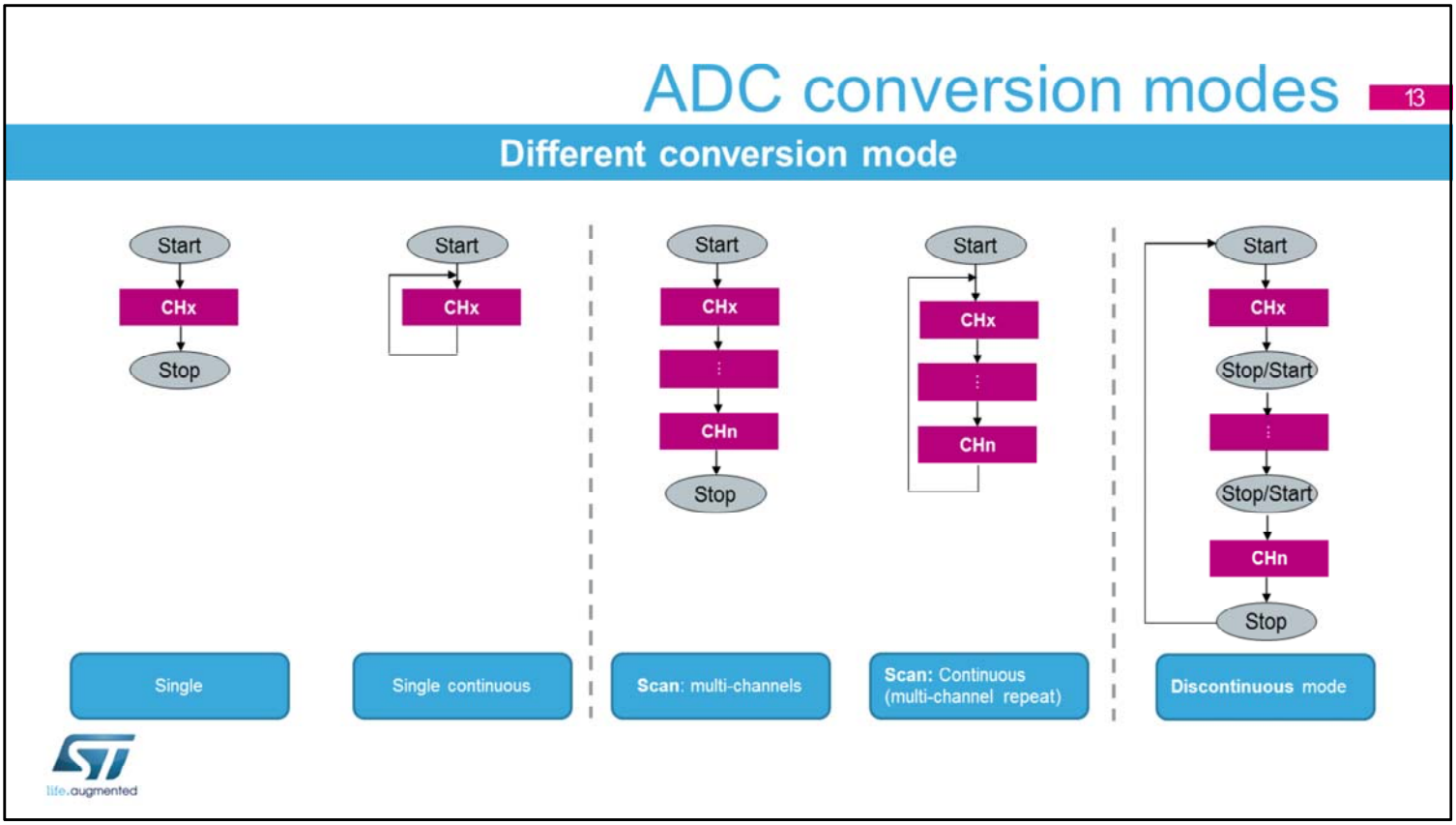
- ADC speed depends on the sampling time and the conversion time.
  - During the sampling time, hold capacitor needs to be charged inside the ½LSB error voltage.
    - 16 bit@3.3V, 1LSB =~ 50uV
    - During sampling time, input should be stable at +/-25uV
    - 12 bit@3.3V, 1LSB=~ 800uV
    - During sampling time, input should be stable at +/-400uV
  - Due to this, 16 bit requires longer sampling time than lower resolution.
    - In some case, to get higher sampling rate, lower clock frequency with low sampling period makes faster

      16 bit mode, Fadc=37MHz, sampling period=2.5 cycle resulting 3.35Msps

      16 bit mode, Fadc=36MHz, sampling period=1.5 cycle resulting 3.60Msps

life.augmented

The ADC speed also depends on the sampling time and the conversion time. During the sampling time, the hold capacitor must be charged to the proper LSB voltage with an error lower than a half of the LSB voltage to ensure a correct accuracy.
And the higher the resolution, the longer the sampling time. This is why it is sometimes more efficient to use a lower clock frequency with a low sampling period to get a higher sampling rate.



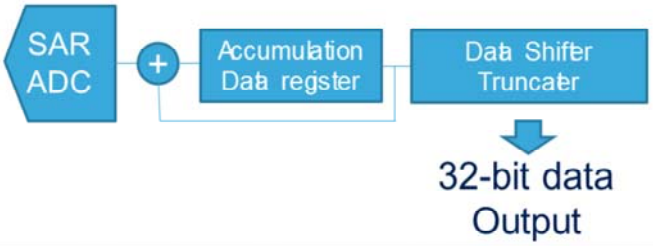
The ADC supports several conversion modes:

- 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, 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 x1024
- Programmable data shifter & truncater Left shift of 0 to 15 bits, right shift of 0 to 11 bits.
- Up to 32-bit data width
- Averaging, data rate reduction, SNR improvement, and basic filtering



Oversampling ratio	Output resolution	Equivalent sampling frequency (max.)
x1(none)	16 bits	3.6 Msamples/s
x16	18 bits	225 ksamples/s
x256	20 bits	14 ksamples/s
x1024	21 bits	2.9 ksamples/s

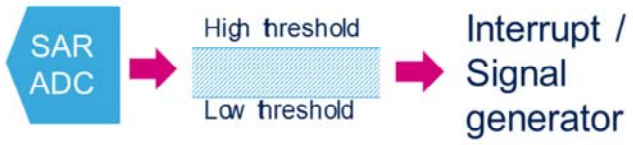
The ADCs support hardware oversampling. They can sample by 2 to 1024 times without CPU support. The converted data is accumulated in a register and the output can be processed by the data shifter and the truncater. 16-bit data can be extended to be presented as 32-bit data register. This functionality 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

15

Reduced software overhead

- Each of the 3 ADCs has three Window comparators
  - One 26-bit analog watchdog can monitor one selected channel or all enabled channels
  - Two 26-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 window threshold, 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

16

Reduced software overhead

- Regular conversion data is stored in a 32-bit data register
  - Software polling, interrupts or DMA requests can be used to move 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 32-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 32-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 of the 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 32-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 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

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

Mode	Description
Run	Active.
Sleep	Active. Peripheral interrupts cause the device to exit Sleep mode.
Stop	Not available. Peripheral registers content is kept
Standby	Powered-down. The peripheral must be reinitalized 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.
- ADC3 (D3 domain) can run while D1 & D2 domains are in Standby mode.

life.augmented

The ADCs are active in Run and Sleep modes. In 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.
Note the ADC3, located in the D3 domain, can run while other domains are in Standby mode.

Performance

	Condition	Data (typ.)	Unit
Sampling rate	16-bit mode	3.6	Msamples/s
	14-bit mode	5.0	Msamples/s
	8-bit mode	8.3	Msamples/s
DNL	(single end)	+3/-1	LSB
INL	16-bit mode	±11.0	LSB
ENOB	16-bit mode (single end)	12.2	bits
	16-bit mode (differential)	13.2	bits
Consumption	5 Msamples/s(14-bit)	1.6	mA
	50 ksamples/s(14-bit)	220	µA

Note : The data indicated here is for the Direct channel

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

The ENOB of 16-bit mode is saturated when less than 14 bits due to the noise level of the system. By using Over-sample mode, the ENOB can be extended further.

Related peripherals

- Refer to these trainings 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	ADC3
Dual mode	Master	Slave	-
Interconnect / Domain	AHB1/ Domain 2	AHB1/ Domain 2	AHB4/ Domain 3
Internal channel connection		DAC1 Out DAC2 Out	Bandgap (VREFINT) Temp sensor VBAT/4

life.augmented

The STM32H7 embeds three ADCs. ADC 1 and ADC 2 can be configured to work together in Dual mode, so that each analog-to-digital conversion can be synchronized between the two modules. ADC 3 works as a standalone converter.

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.