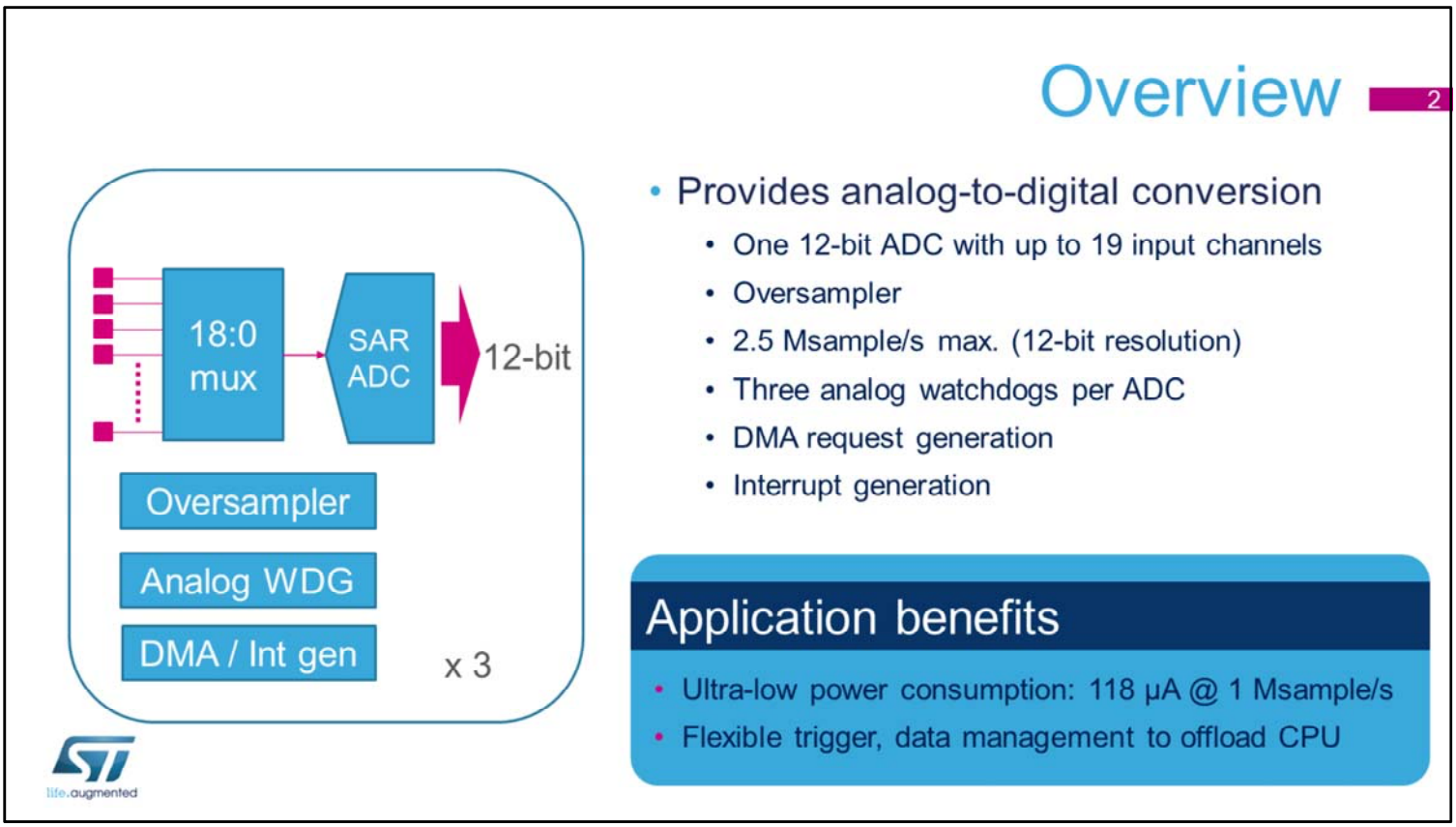
Hello and welcome to this presentation of the STM32 Analog-to-Digital Converter block. It will cover the main features of this block, which is used to convert the external analog voltage-like sensor outputs to digital values for further processing in the digital domain.



The analog-to-digital converter inside the STM32G0 microcontroller allows the microcontroller to accept an analog value like from a sensor output and convert the signal for use in the digital domain. There are up to 19 analog inputs. The ADC module itself is a 12-bit successive approximation converter with additional oversampling hardware.

The oversampling unit preprocesses the data to offload the main processor. It can handle multiple conversions and average them into a single data with an increased data width, up to 16 bits.

The sampling speed is 2.5 mega samples per second for 12-bit resolution. 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.
The ADC module also integrates an analog watchdog.

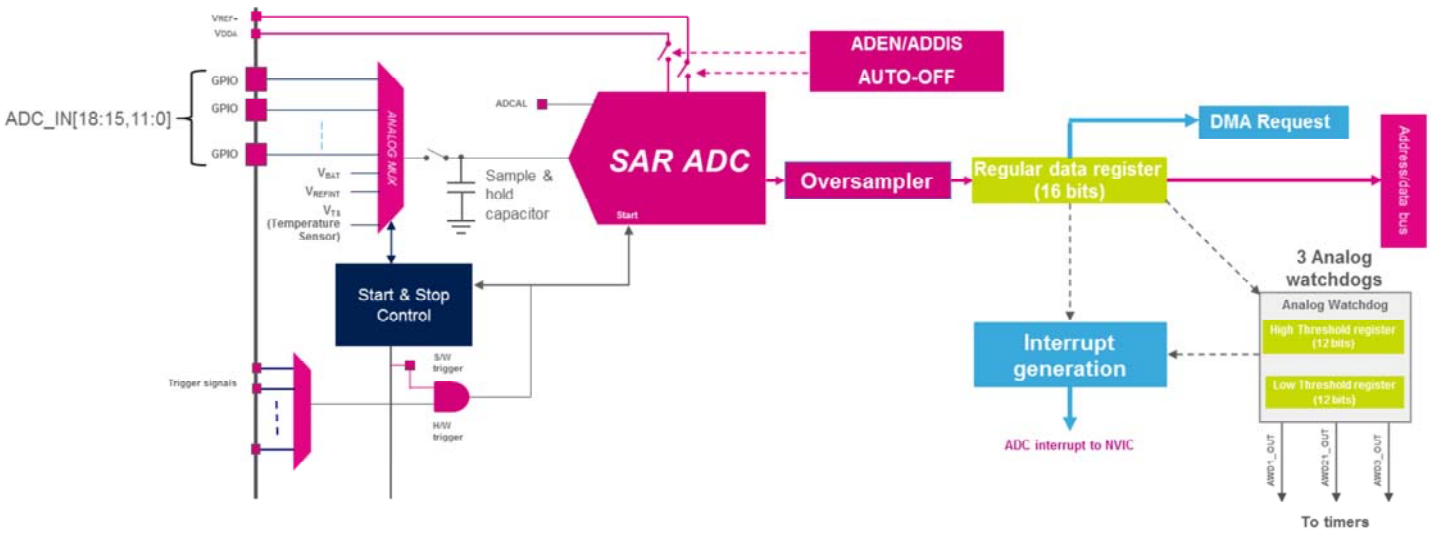
Key features

Features	Description
Input channel	Up to 16 external (GPIOs) and 3 internal channels
Type of conversion	12-bit successive approximation
Conversion time	400 ns, 2.5 Msamples/s (when f_{ADC_CLK} = 35 MHz, 12 bits)
Functional mode	Single, Continuous, Scan, and Discontinuous
Triggers	Software or external trigger (Timers & IOs)
Special functions	Analog watchdogs, Hardware oversampling, and Self-calibration
Data processing	Interrupt generation and DMA requests
Low-power modes	Wait, Auto-off, and Power-down

life.augmented

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 at 2.5 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

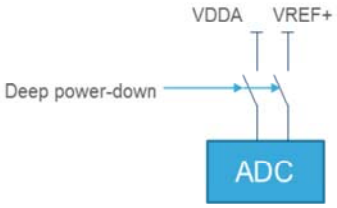


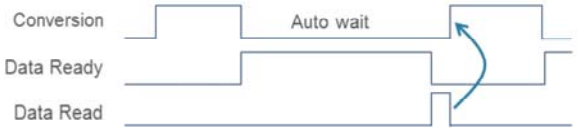
This slide shows the general block diagram of the ADC.
The left part describes the analog front-end, based on the analog multiplexer and the embedded sampling capacitor of the ADC. This sampling time must be enough for the input voltage source to charge the sample and hold capacitor to the input voltage level.
The right part is the digital back-end. Samples are stored into a data register that is either read by software or transferred to memory via a DMA channel.
3 analog watchdogs monitor the voltage of selected analog input with regard to high and low thresholds. When the voltage is not within the guard range, an interrupt may be generated and a trigger event may be signaled to the timer units.
Note that the analog front end can be power-gated when the ADC is not used and also between acquisitions by using the Auto-off function.

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
- Wait mode
  - A new conversion can start only if the previous data has been processed, once the ADC_DR register has been read or if the EOC bit has been cleared



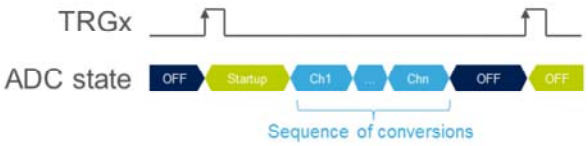


The ADC supports a Deep power-down mode. When the ADC is not used, it can be disconnected by a power switch to further reduce the leakage current. When Wait mode is active, the ADCs wait until the last conversion data is read or the end-of-conversion flag is cleared before starting the next conversion. This avoids unnecessary conversions and thus reduces power consumption.

Low-power features

Several low-power features are implemented

- Auto-off mode
  - The ADC is always powered off when not converting and automatically wakes up when a conversion is started (by software or hardware trigger). A startup time is automatically inserted between the trigger event which starts the conversion and the sampling time of the ADC



- Power consumption depends on sampling time
  - 475 µA @ 2.5 Msample/s
  - 15 µA @ 10 ksamples/s

life.augmented

The ADC has an automatic power management feature which is called Auto-off mode.
When Auto-off mode is enabled, the ADC is always powered off when not converting and automatically wakes up when a conversion is started (by software or hardware trigger). A startup time is automatically inserted between the trigger event which starts the conversion and the sampling time of the ADC. The ADC is then automatically disabled once the sequence of conversions is complete.
Auto-off mode can be combined with the Wait mode conversion (WAIT=1) for applications clocked at low frequency. This combination can provide significant power savings if the ADC is automatically powered off during the wait phase and restarted as soon as the ADC_DR register is read by the application.
The power consumption is in function of the sampling

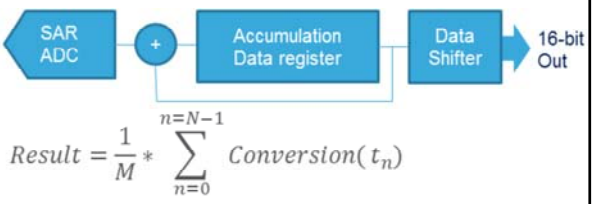
frequency. For low sampling rates, the current consumption is reduced almost proportionally.

High-performance features

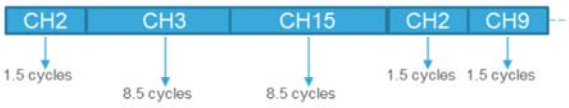
7

Several high-performance features are implemented

- Hardware oversampling
  - Accumulator and bit shifter can output 16-bit data without CPU support
- Flexible sequencer
  - Software can set a sequence of conversion by using two options
    - Ascending or descending selected channel order, up to 19 channels
    - User-defined order, up to 8 channels
- Self-calibration to reduce offset



$$Result = \frac{1}{M} * \sum_{n=0}^{n=N-1} Conversion(t_n)$$



life.augmented

The ADC includes the oversampling hardware which accumulates data and then divides without CPU help. The oversampler can accommodate from 2 to 256 time samples and right shift from one to eight binary digits.
The sequencer allows the user to convert either up to 19 channels in ascending or descending order or up to 8 channels in a user-defined order.
Two sampling periods can be programmed. Each channel is assigned one of these two values.
The ADC offers a self-calibration mechanism for the offset. It is recommended to run the offset 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. High temperature excursion may also require to run the offset calibration.

ADC conversion speeds

Conversion speed is resolution dependent

- ADC needs minimum 1.5_{ADC_CLKs} for sample period and 12.5_{ADC_CLKs} for conversion (12 bits).
- 35 MHz maximum clock with 14-cycle results in 2.5 Msamples/s
- Speed up by low resolution
  - 12-bit: 12.5_{ADC_CLKs}(+1.5) => 2.5 Msamples/s
  - 10-bit: 10.5_{ADC_CLKs} (+1.5) => 2.92 Msamples/s
  - 8-bit: 8.5_{ADC_CLKs} (+1.5) => 3.50 Msamples/s
  - 6-bit: 6.5_{ADC_CLKs} (+1.5) => 4.37 Msamples/s

life.augmented

Resolution	$t_{Conversion}$
12 bits	12.5 Cycles
10 bits	10.5 Cycles
8 bits	8.5 Cycles
6 bits	6.5 Cycles

The ADC needs a minimum of 1.5 clock cycles for the sampling and 12.5 clock cycles for conversion for 12-bit mode. With a 35 MHz ADC clock, it can achieve 2.5 mega samples per second. For a higher sampling speed, it is possible to reduce the resolution down to 10, 8 or 6 bits.

Sampling times

Programmable sampling time

- One ADC can scan the different input source with various source impedances
  - The following sampling times are supported:
    - 1.5 cycles, 3.5 cycles, 7.5 cycles, 12.5 cycles, 19.5 cycles, 39.5 cycles, 79.5 cycles, 160.5 cycles
- The user has to select two of these values
  - The ADC sampling time register ADC_SMPR is used to assign one of these two values to each analog channel

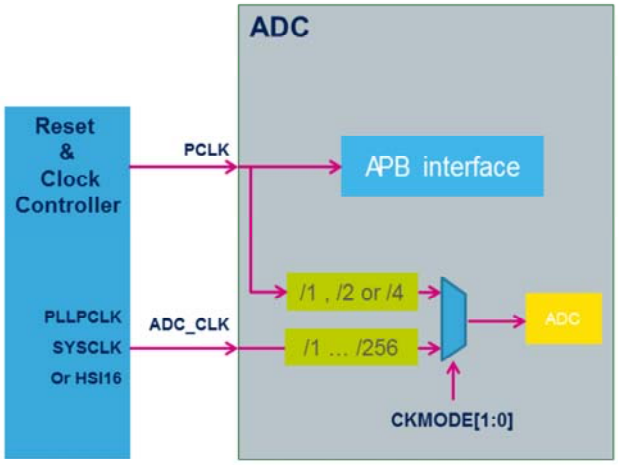
life.augmented

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.
At a given time, only a pair of possible sampling times is active. For each analog channel, the user is free to select one of these two selected values.

ADC clocks

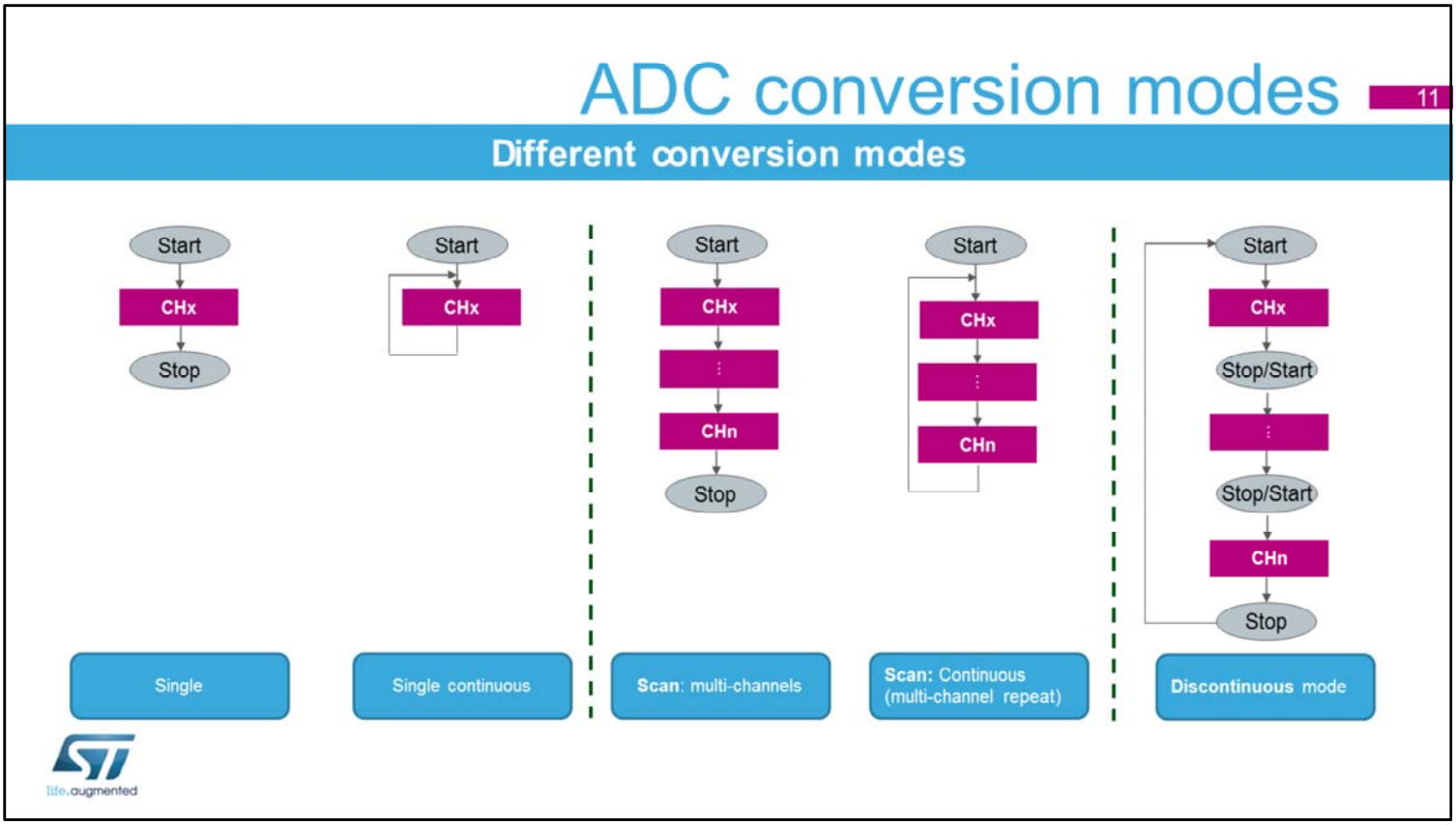
Flexible dock selection

- ADC clock can be selected from
  - APB clock divided by 1, 2 or 4.
    If a trigger event depends on the APB clock, the latency between the event and start of conversion is deterministic.
  - Dedicated ADC clock
    - The CPU can run slowly even if the ADC is running full-speed
    - When the ADC_CK source is PLLPCLK or HSI16, it is asynchronous to APB clocks. An uncertainty of the trigger instant is added by the resynchronizations between the two clock domains



life.augmented

The ADC has a selectable clock source. When the system needs to run synchronously, the APB 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. In this case, the ADC implements two clock domains: PCLK and ADC_CLK and delays are needed to perform the resynchronizations between them.



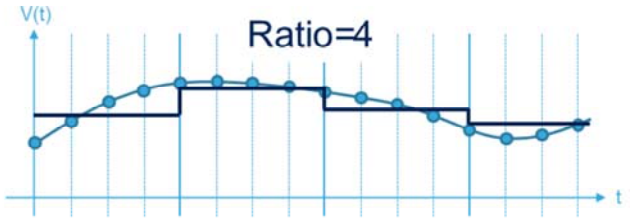
The ADC supports several conversion modes:

- 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 x256
- Programmable data shifter & truncater
  - Right shift of 0 to 8 bits
  - 4-bit truncation so that the result fits into the 16-bit data registers ($2^8 * 2^{12} = 2^{20}$)
- Averaging, data rate reduction, SNR improvement, and basic filtering



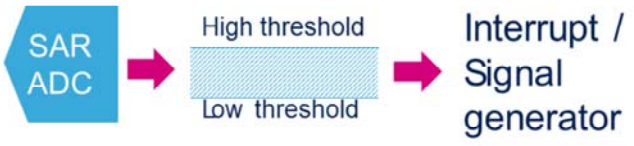
Oversampling ratio	Output resolution	Equivalent sampling frequency (max.)
x1(none)	12 bits	2.5 Msamples/s
x16	16 bits	156.25 ksamples/s
x256	20 bits before shift & truncate	9.77 ksamples/s

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 truncater.
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.

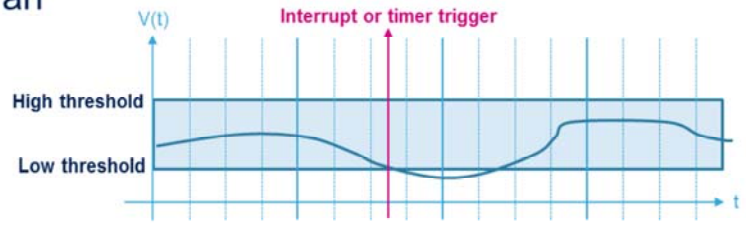
Analog watchdog

Reduced software overhead

- The ADC has three Window watchdogs
  - One Analog Watchdog can monitor one selected channel or all enabled channels
  - Two 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



Window Watchdog	Channel to monitor
AWD1	1 or all
AWD2	All channels selected in ADC_AWD2CR
AWD3	All channels selected in ADC_AWD3CR



Each ADC has 3 integrated 12-bit analog watchdogs 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 timer trigger signal can be asserted 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 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
- It is possible to configure if the data is preserved or overwritten when an overrun event occurs

The ADC conversion result is stored in a 16-bit data register. The system can use CPU polling, interrupts or the DMA controller 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. In case of overrun, either the new sample is dropped or the previous sample is overwritten.

Interrupts and DMA

Interrupt event	Description	Interrupt event	Description
ADRDY	The ADC is ready to convert	**AWD1-3**	Analog watchdog threshold breach detection occurs
EOC	End of conversion	**EOSMP**	End of a sampling phase
EOS	End of sequence	**OVR**	Data overrun occurs
EOCAL	End of calibration	**CCREADY**	Channel Configuration Ready

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

Each ADC can generate 8 different interrupts: ADC Ready, end of conversion, end of sequence, end of calibration, analog watchdog, end of sampling, data overrun and channel configuration ready.
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 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.

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.

Performance

	Condition	Data (typ.)	Unit
Sampling rate	12-bit mode	2.5	Msamples/s
	10-bit mode	2.9	Msamples/s
	8-bit mode	3.5	Msamples/s
Differential linearity error		±1.2	LSB
Integral linearity error	12-bit mode	±2.5	LSB
Effective Number Of Bits (ENOB)	12-bit mode	10.2	bit
Consumption	2.5 Msamples/s	475	µA
	10 ksamples/s	17	µA

life.augmented

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

Feature comparison

ADC in STM32F0 MCUs	ADC in STM32G0 MCUs
TSMC180	TSMC090
Performance resolution	Same
Conversion time: 1 µs, 12-bit resolution	0.4 µs, 12-bit resolution (2.5 Msamples/s)
Calibration, Sampling time data alignment, and DMA support	Same
Low power: freq, wait mode & conversion phase, and Analog Input channel	Same
1 Analog watchdog	3 Analog watchdogs
Conversion trigger: Tim1/2/3/15	Tim1/2/3/6/15
Conversion mode, interruption	Same
Supply: 2.4 to 3.6 V	1.62 to 3.6 V
No HW oversampling	HW oversampling
Simple sequencer, one global sampling time	Advanced sequencer, two sampling time groups

life.augmented

This table summarizes the differences between the STM32F0 ADC and the STM32G0 ADC.

Related peripherals

- Refer to these trainings linked to this peripheral, if needed:
  - DMA – Direct memory access controller
  - Interrupts – Nested Vectored Interrupt Controller
  - GPIO – General-purpose inputs and outputs
  - RCC – Clock module
  - DAC – Digital-to-analog converter
  - TIM – Timers for triggering interrupts and events

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.

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.