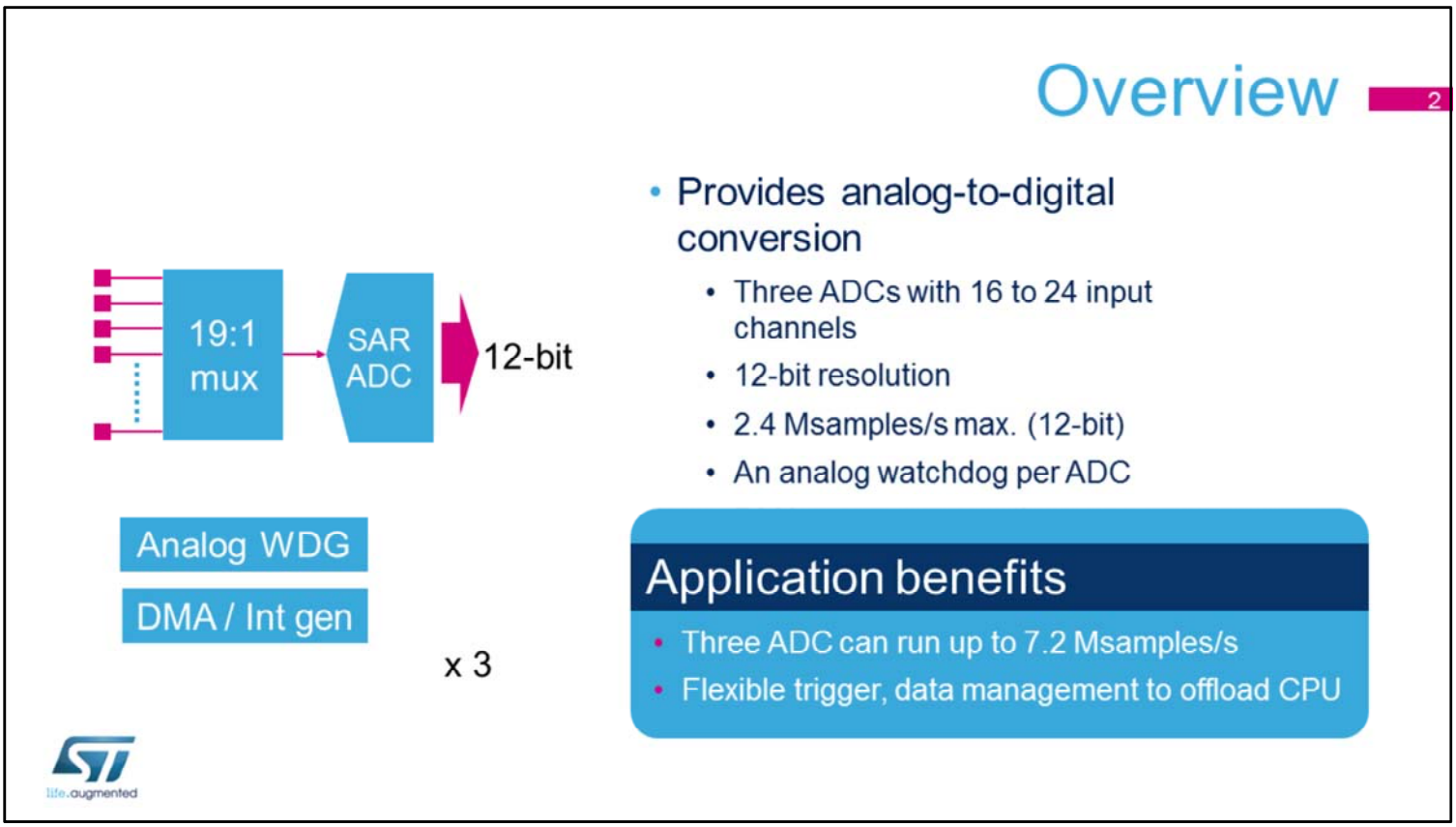
Hello and welcome to this presentation of the STM32F7 Analog-to-Digital Converter block. It will cover the main features of this block, which is used to convert the external analog voltage-like sensor outputs to digital values for further processing in the digital domain.



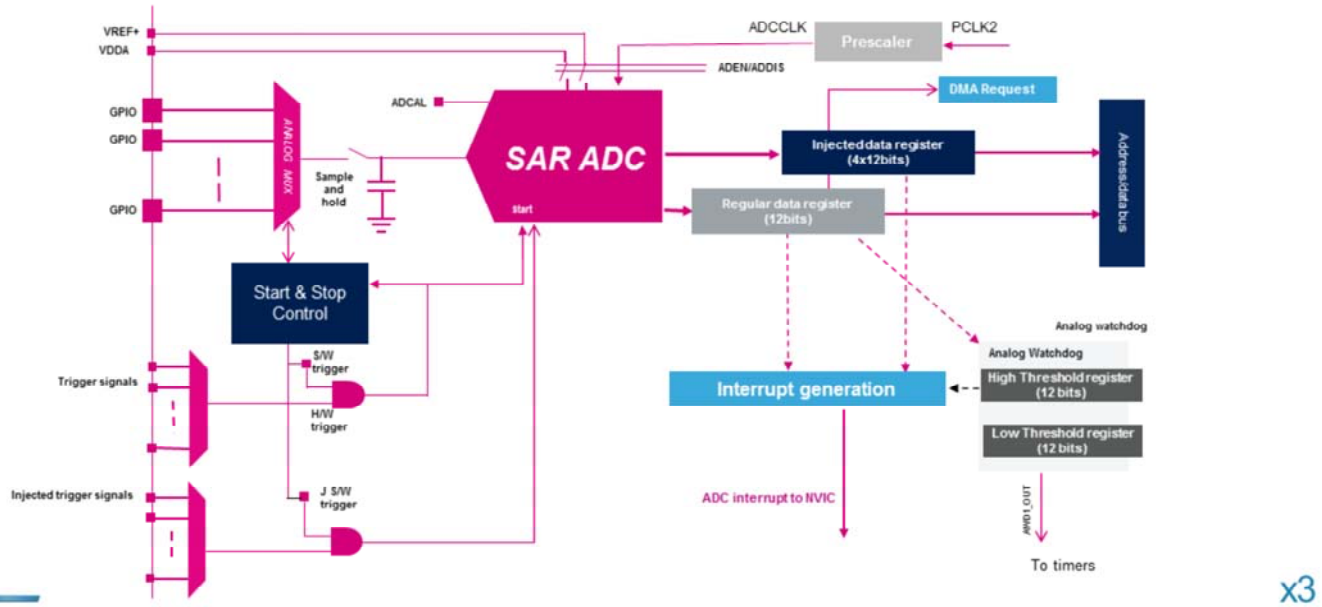
The analog-to-digital converters inside STM32 products allow the microcontroller to accept an analog signal, like a sensor output, and convert the signal into the digital domain. There are 16 to 24 analog inputs available across the three ADCs. The ADC module itself is a 12-bit successive approximation converter. The sampling speed is two mega samples per second. Each ADC module integrates an analog watchdog. The data can be made available either through DMA movement or interrupts. There are a number of triggering mechanisms and the data management can be configured to minimize the CPU workload.

Key features

Features	Description
ADC units	3 modules
Input channel	16..24 external channels (GPIOs), single-end
Technology	12-bit successive approximation
Conversion time	417 ns, 2.40 Msamples/s (when f_{ADC_CLK} = 36 MHz)
Functional mode	Single, Continuous, Scan, Discontinuous, or Injected
Triggers	Software or external trigger (for Timers & IOs)
Special functions	Analog watchdogs, Dual/Triple mode, Self-calibration
Data processing	Interrupt generation, DMA requests

3 analog-to-digital converters are integrated inside STM32F7 products. The input channel is connected to up to 24 GPIO channels capable of converting signals in either Single-end or Differential mode. The ADCs can convert signals in excess of 5 mega samples per second. 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.

Block diagram



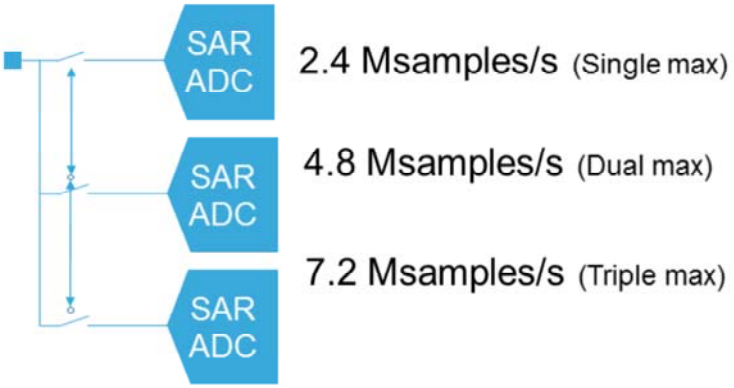
This slide shows the general block diagram for the 3 analog-to-digital converters embedded in the STM32F7.

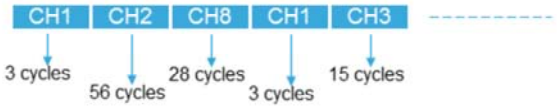
High performance features

5

- Several high performance features are implemented

- 2.4 Msamples/s for 36 MHz ADC clock
- Interleave mode can support up to 7.2 Msamples/s
- Flexible sequencer
- Self-calibration to reduce offset





life.augmented

The ADC supports up to 2.4 mega samples per second of conversion. By using triple interleaved mode, it can be extended to 7.2 mega samples per second. The sequencer allows the user to convert up to 16 channels in any desired order. Also each channel can have different sampling period. The ADC offers an auto calibration mechanism. 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 reestablished.

ADC conversion speeds

- Conversion speed is resolution dependent

- ADC needs minimum 3ADC_CLKs per sample period and 12ADC_CLKs for conversion (12-bit).
- 36 MHz maximum clock with a 15 cycle results in 2.4 Msamples/s
- Speed up by lower resolution
  - 10-bit : 10ADC_CLKs(+3) => 2.77 Msamples/s
  - 8-bit : 8ADC_CLKs (+3) => 3.27 Msamples/s
  - 6-bit : 6ADC_CLKs (+3) => 4.00 Msamples/s

Resolution	$t_{Conversion}$
12 bits	12 Cycles
10 bits	10 Cycles
8 bits	8 Cycles
6 bits	6 Cycles

The ADC needs a minimum of 3 clock cycles for the sampling and 12 clock cycles for the conversion. With an 36 MHz ADC clock, it can achieve 2.4 mega samples per second. For higher speed sampling, it is possible to reduce the resolution down to 6 bits then the sampling speed can go up to 4 mega samples per second.

Sampling times

Programmable sampling time

- The following sampling times can be selected:
  - 3 cycles
  - 15 cycles
  - 28 cycles
  - 56 cycles
  - 84 cycles
  - 112 cycles
  - 144 cycles
  - 480 cycles
- If Scan mode is selected, each input channel can have a different sampling time
  - One ADC can scan the different input source independent to the source impedance.

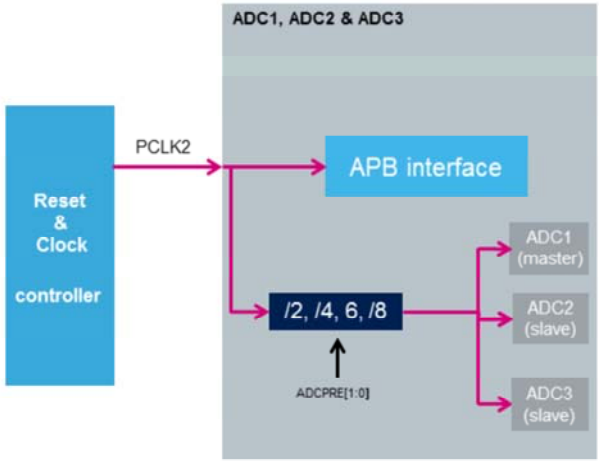
life.augmented

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.

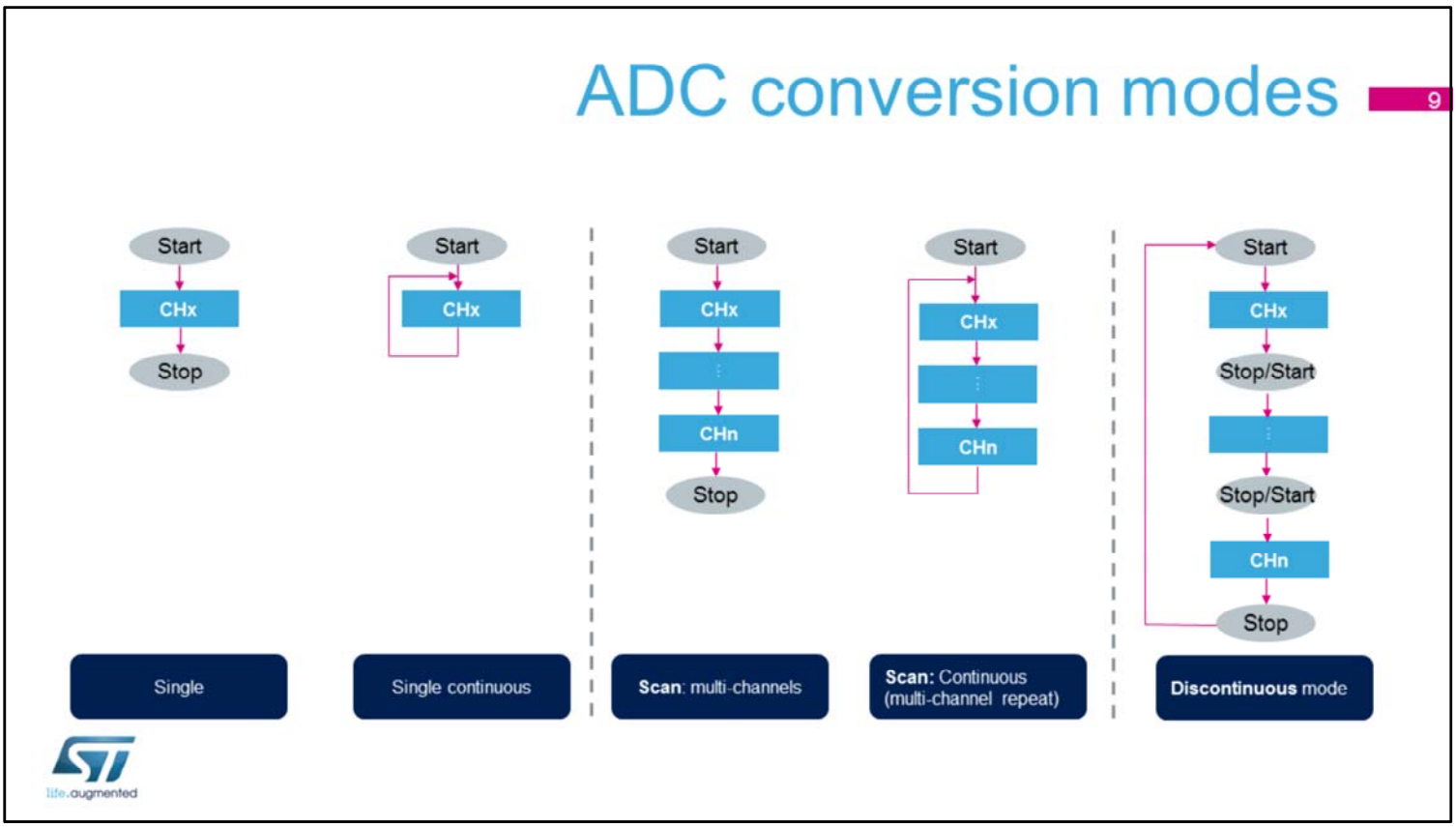
ADC clocks

Flexible clock selection

- ADC has two clock domains
  - APB interface is running by PCLK2
  - Analog portion of ADC is running from APB2 clock divided by 2, 4, 6 or 8.



The ADC has two clock domains, one for the interface and another for the analog core. The analog clock input has a prescaler to select the clock speed.

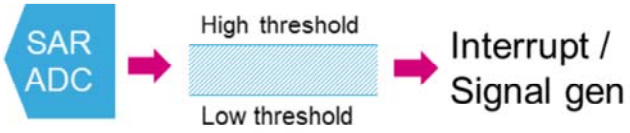


The ADC supports several conversion modes:
–Single mode, which converts only one channel, in single-shot or continuous mode.
–Scan mode, which converts a complete set of pre-defined programmed input channels, in single-shot or continuous mode.
–Discontinuous mode, converts only a single channel at each trigger signal from the list of pre-defined programmed input channels.

Analog watchdog

Reduced software overhead

- Each of the 3 ADCs has a window comparator
  - One 12-bit analog watchdog can monitor one selected channel or all enabled channels
- Each watchdog continuously monitors an over- and/or under-threshold condition, then generates either an interrupt or external signal or stops a timer.

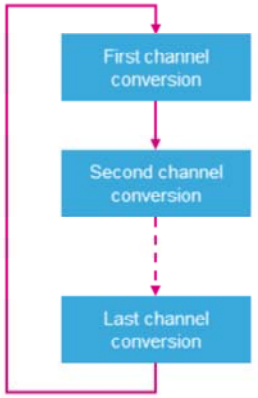


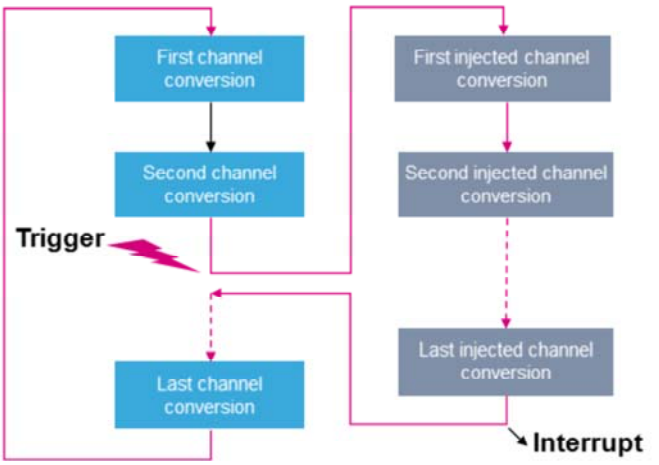
Each ADC has an integrated analog watchdog with high and low threshold settings. The ADC conversion value is compared to this threshold window. If the result exceeds the threshold, an interrupt or external signal can be generated or a timer can be immediately stopped without CPU intervention.

Conversion modes

Regular and injected conversions

- Regular Scan mode
- Scan mode with injected high priority trigger





life.augmented

In normal scan conversion mode, the conversion is repeated for all channels one by one.
If an injected conversion is started, the ADC stops converting the normal conversion channels and proceed to convert the injected configured channel. At the end of all injected channels conversion, the ADC resumes the conversion from the channel where the normal conversion was stopped.
When the trigger occurs, the regular conversion needs to be finished before the ADC proceeds to the injected conversion.

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.
- 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.

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 converted data. An overrun flag can be generated if data is not read before the next converted data is ready. For injected channel conversions, 4 dedicated data registers are available.

Injected conversions

Interruption during of the regular 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.

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
EOC	The end of regular conversion group
JEOC	The end of injected conversion group
AWD	An analog watchdog threshold breach detection occurs
OVR	A data overrun occurs

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

life.augmented

Each ADC can generate 4 different interrupts: end of conversion, end of injected conversion, analog watch dog, and data overrun.
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.

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	2.40	Msamples/s
	6-bit mode	4.00	Msamples/s
DNL	f_{ADC} = 36 MHz	+/-2	LSB
INL	f_{ADC} = 36 MHz	+/-3	LSB
ENOB	f_{ADC} = 36 MHz	10.8	bits
Consumption	VDDA + VREF+	1.9	mA

The following table shows performance parameters for the ADC.

Features for each individual ADC

ADC features	ADC1	ADC2	ADC3
Dual mode	Master	Slave	Slave
Internal channel connection	Bandgap Temp sensor VBAT	-	-

The STM32F7 embeds three ADCs. ADC 1, ADC 2 and ADC 3 can be configured to work together in Dual or Triple mode, so that each analog-to-digital conversion can be synchronized between the two or three modules.

Related peripherals

- Refer to these peripherals 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

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 documentation:
  - 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

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.