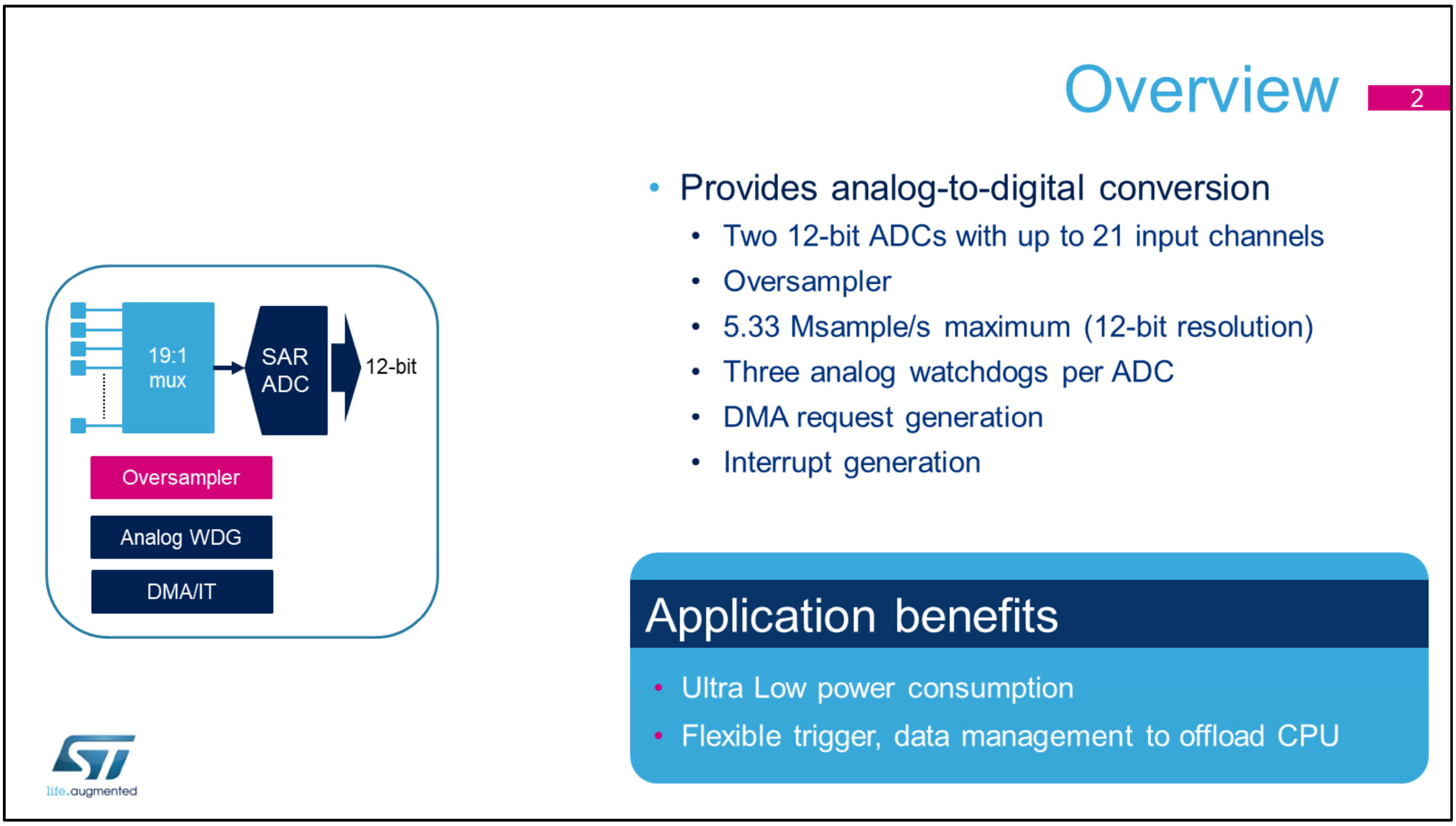
STM32L5 – ADC

Analog-to-Digital Converter

Revision 1.0

life.augmented

Hello and welcome to this presentation of the STM32L5 Analog-to-Digital Converter block. It will cover the main features of this block, which is used to convert analog voltages, like sensor outputs, to digital values for further processing in the digital domain.



Overview

19:1
mux
SAR
ADC
12-bit
Oversampler
Analog WDG
DMA/IT

- Provides analog-to-digital conversion
  - Two 12-bit ADCs with up to 21 input channels
  - Oversampler
  - 5.33 Msample/s maximum (12-bit resolution)
  - Three analog watchdogs per ADC
  - DMA request generation
  - Interrupt generation

Application benefits

- Ultra Low power consumption
- Flexible trigger, data management to offload CPU

life.augmented

The analog-to-digital converter inside the STM32L5 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 21 analog inputs available across the two ADCs

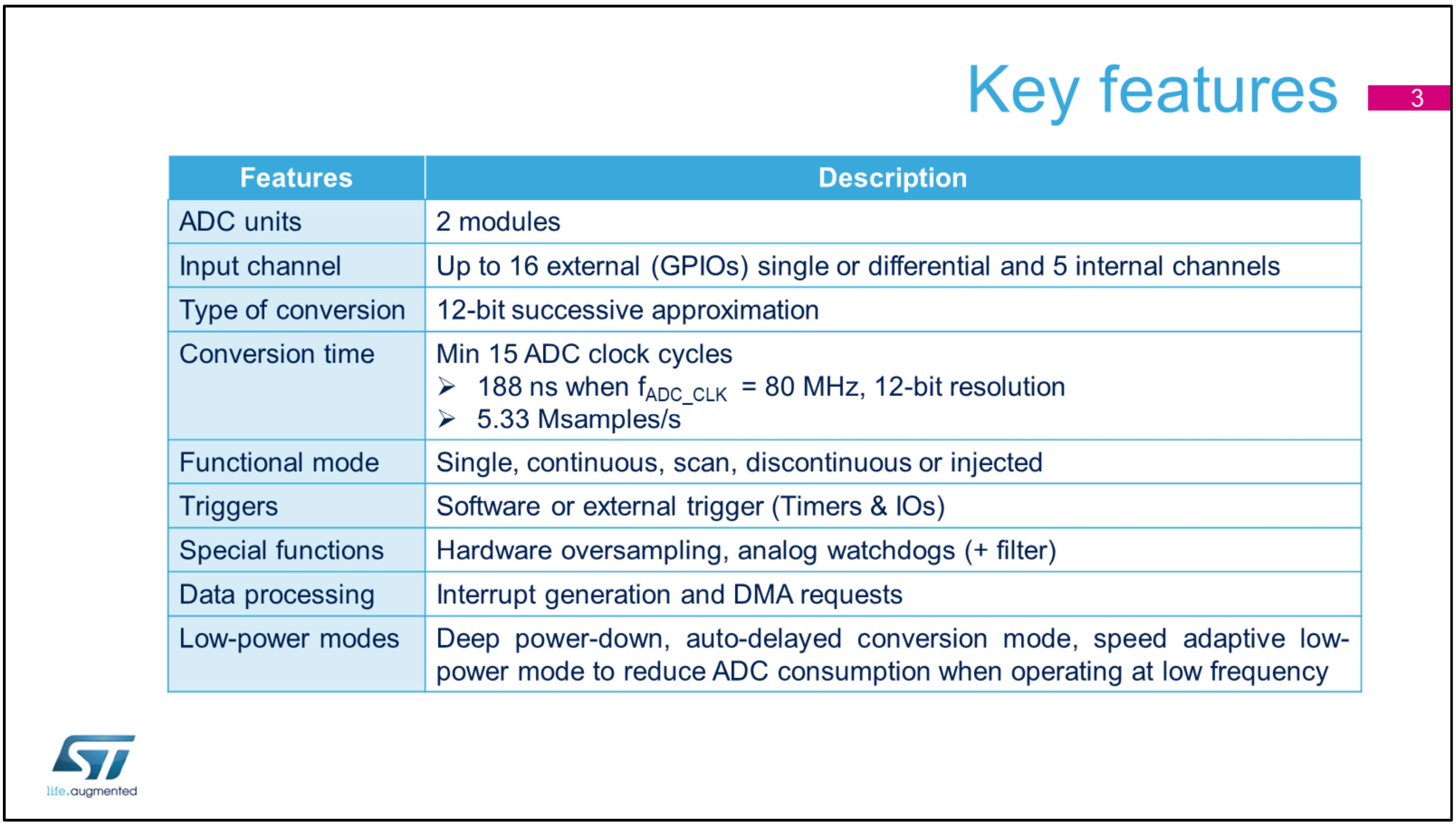
The oversampling unit preprocesses the data to offload the main processor. It can handle multiple conversions and average them into a single data with an increased data width, up to 16 bits.

The ADC module itself is a 12-bit successive approximation converter with additional oversampling hardware.

The sampling speed is 5.33 mega sample per second for 12-bit resolution.

The data can be made available to the system by either through DMA transfers, interrupts or polling.

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
ADC units	2 modules
Input channel	Up to 16 external (GPIOs) single or differential and 5 internal channels
Type of conversion	12-bit successive approximation
Conversion time	Min 15 ADC clock cycles ➢ 188 ns when f_{ADC_CLK} = 80 MHz, 12-bit resolution ➢ 5.33 Msamples/s
Functional mode	Single, continuous, scan, discontinuous or injected
Triggers	Software or external trigger (Timers & IOs)
Special functions	Hardware oversampling, analog watchdogs (+ filter)
Data processing	Interrupt generation and DMA requests
Low-power modes	Deep power-down, auto-delayed conversion mode, speed adaptive low-power mode to reduce ADC consumption when operating at low frequency

2 analog-to-digital converters are integrated inside STM32L5xx products.

The input channel is connected to up to 21 channels capable of converting signals in either Single-end or Differential mode.

The ADCs can convert signals at 5.33 mega sample per second in 12-bit mode when ADC clock frequency is 80 MHz.

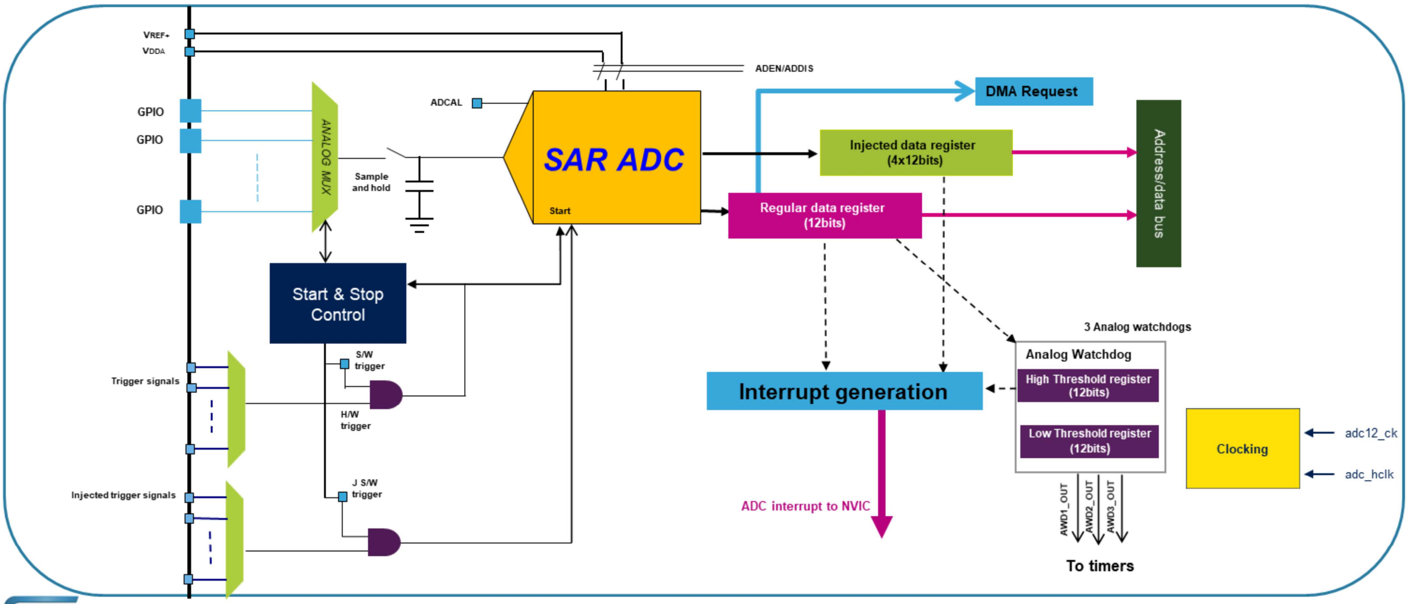
There are several functional modes which will be explained later.

There are also several different triggering methods. In order to offload the CPU, the ADC has an analog watchdog for monitoring thresholds. The analog watchdog has new filtering features.

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

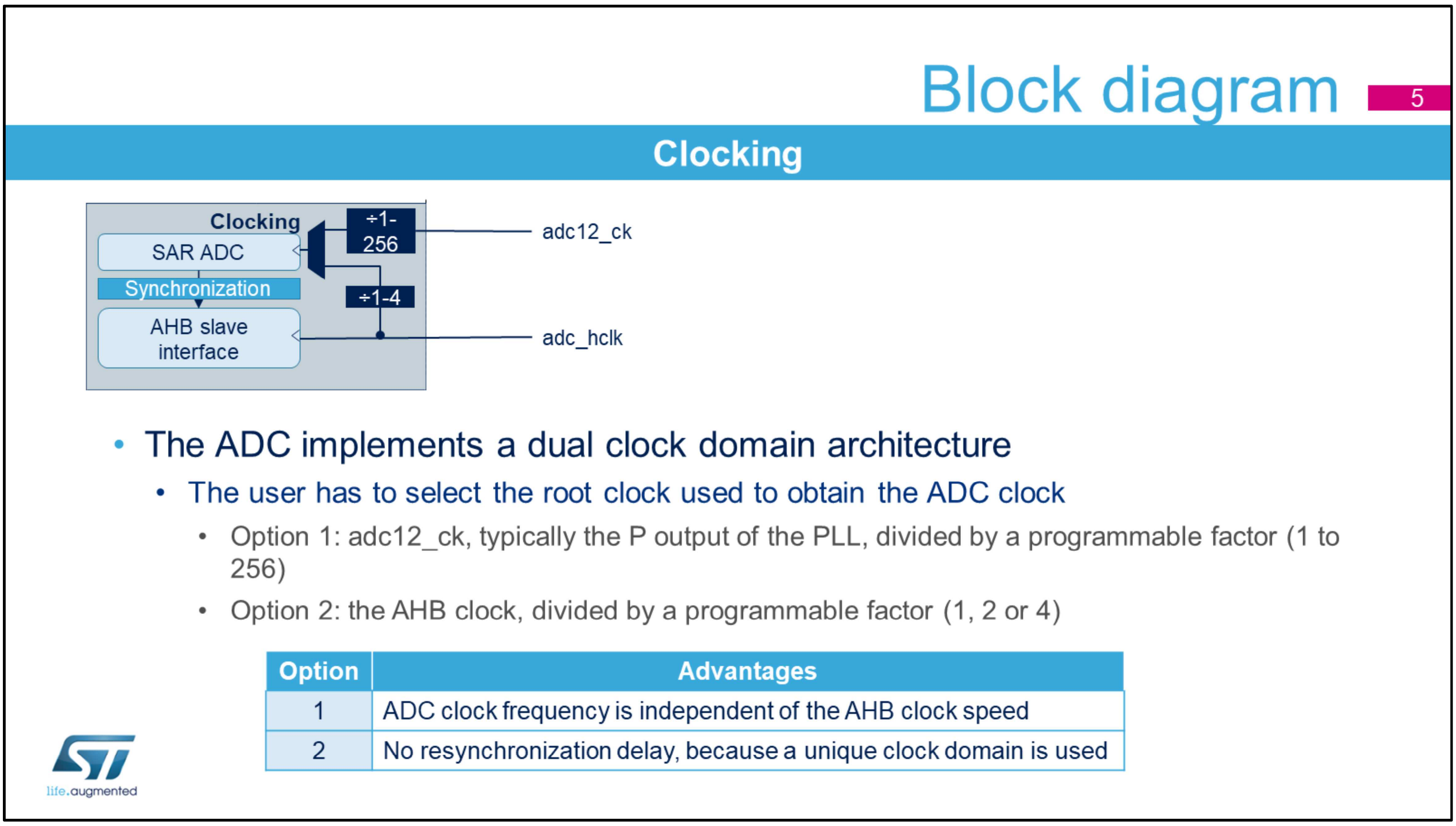


x2

This slide shows the general block diagram of the ADC. The main important sub-units of the ADC are:

- The power supplies and on/off control
- The analog front-end
- The trigger logic
- The digital back-end including the analog watchdogs and the AHB slave interface
- The clocking.

The next slides detail all these sub-units.



Block diagram

Clocking

Clocking

SAR ADC

Synchronization

AHB slave interface

÷1-256

÷1-4

adc12_ck

adc_hclk

- The ADC implements a dual clock domain architecture
  - The user has to select the root clock used to obtain the ADC clock
    - Option 1: adc12_ck, typically the P output of the PLL, divided by a programmable factor (1 to 256)
    - Option 2: the AHB clock, divided by a programmable factor (1, 2 or 4)

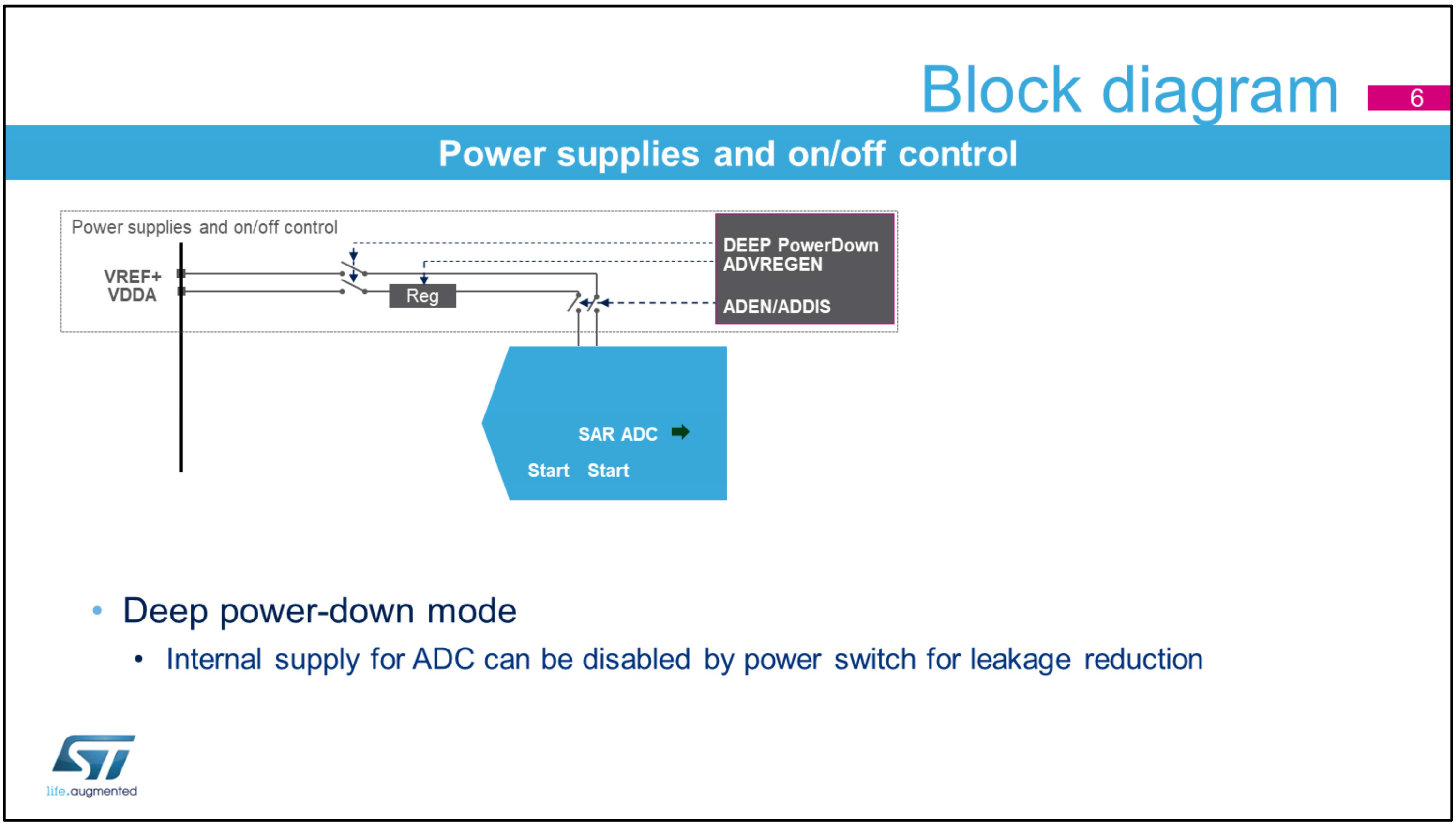
Option	Advantages
1	ADC clock frequency is independent of the AHB clock speed
2	No resynchronization delay, because a unique clock domain is used

life.augmented

The ADC has two clock inputs: adc12_ck and adc_hclk.
The AHB interface belongs to the adc_hclk clock domain.
Regarding the digital part of the SAR ADC, there are two options:

- Either using adc_hclk as the reference clock
- Or using adc12_ck, which is dedicated and independent of adc_hclk.

With this second option, dynamic frequency scaling can be implemented in the AHB and CPU clock domain, while the sampling clock is fixed.
However samples acquired in the adc12_ck clock domain have to pass to the AHB clock domain to be read by CPU or DMA, which requires a synchronization delay.
An uncertainty of the trigger instant is also added by the resynchronizations between the two clock domains.



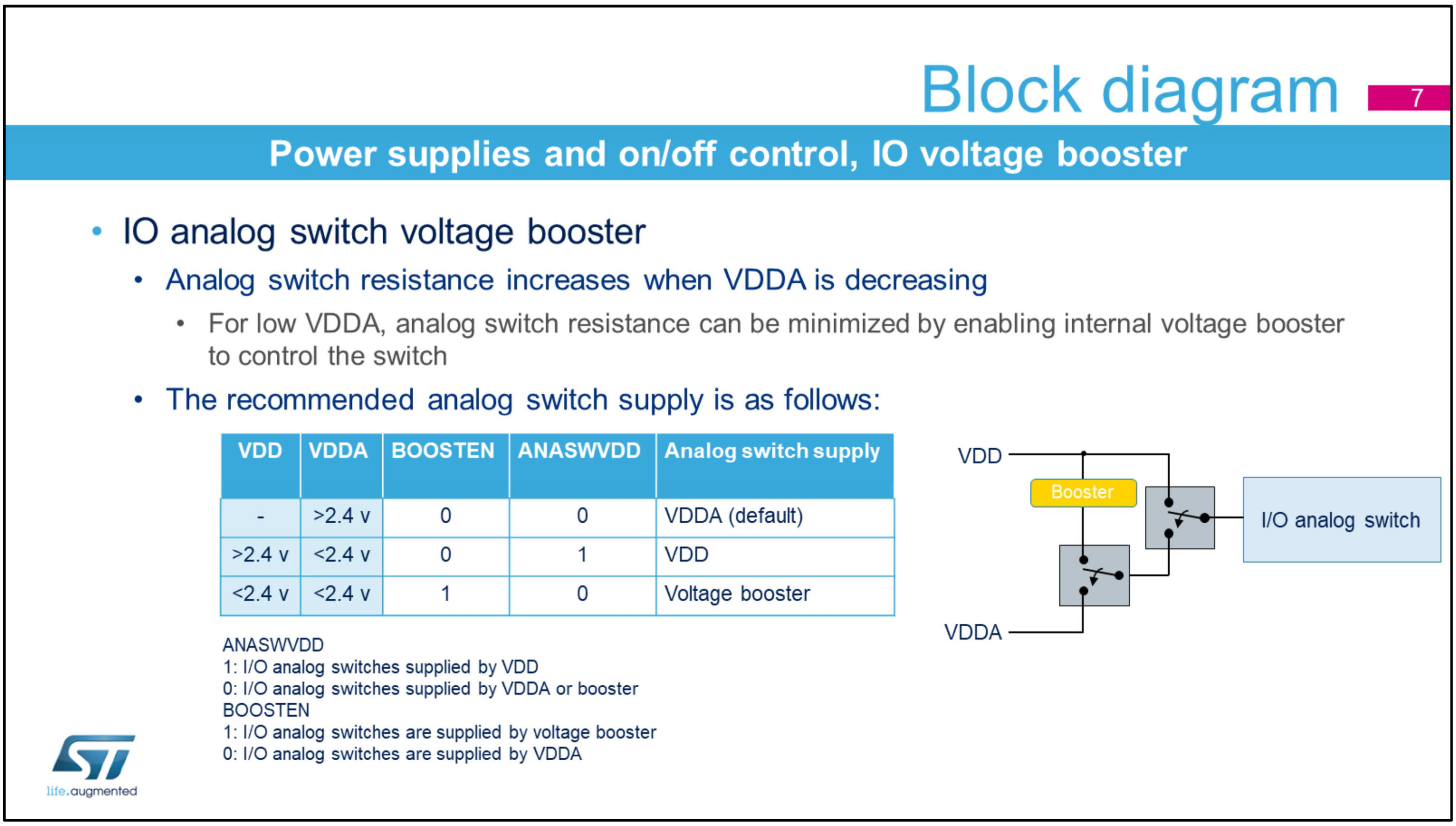
Block diagram

Power supplies and on/off control

- Deep power-down mode
  - Internal supply for ADC can be disabled by power switch for leakage reduction

The analog part of the ADC needs two power supplies:
- VREF+ which is the positive analog reference
- And VDDA which is the analog power supply.

By default, the ADC is in deep-power-down mode, where its supply is internally switched off to reduce the leakage currents.
To start ADC operations, it is first needed to exit Deep-power-down mode by setting the DEEP PowerDown bit to zero.
It is possible to save power by also disabling the ADC voltage regulator. This is done by writing bit ADVREGEN to zero.
Setting ADDIS to one disables the ADC. ADEN and ADDIS are then automatically cleared by hardware as soon as the analog ADC is effectively disabled.



Block diagram

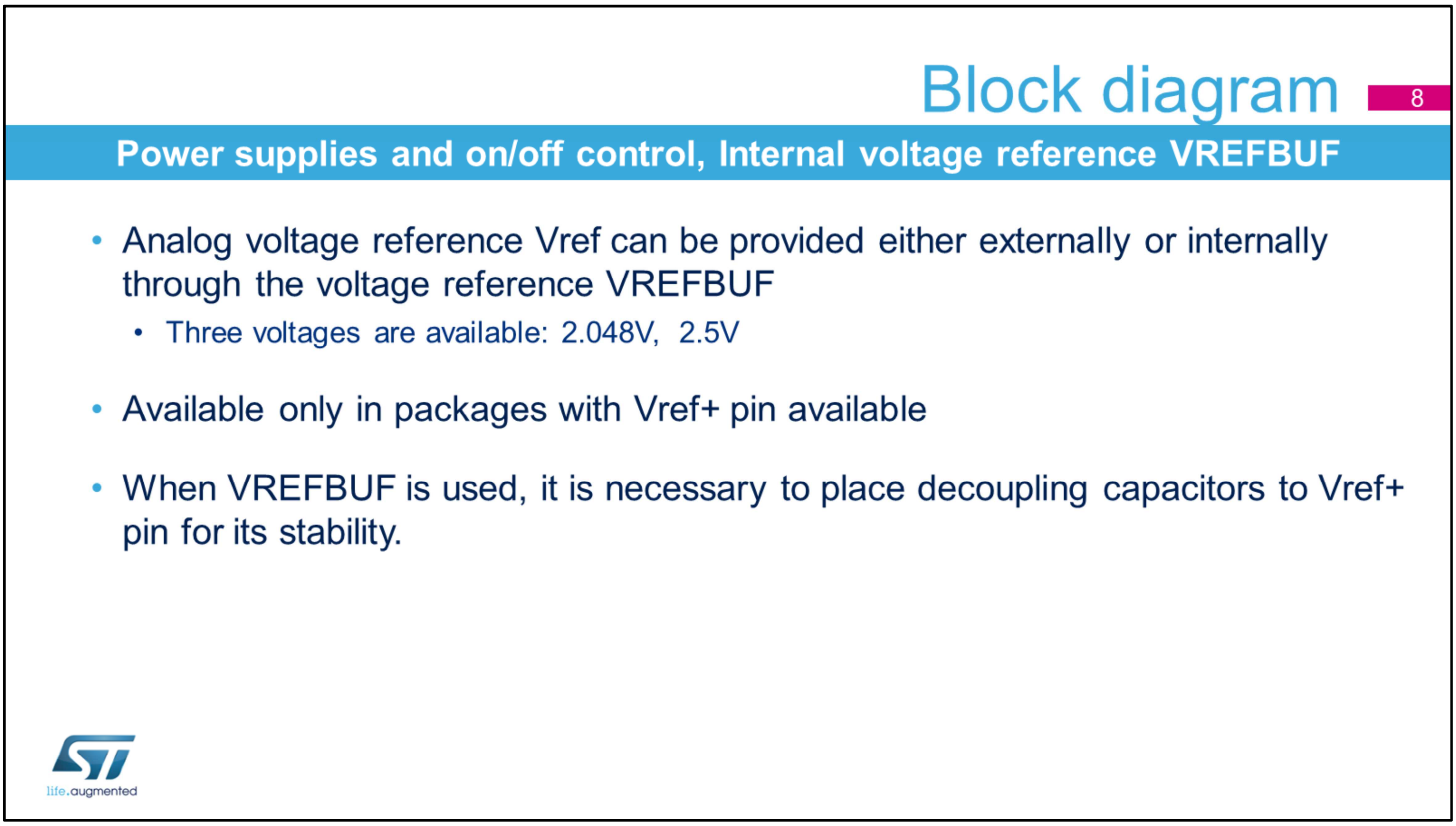
Power supplies and on/off control, IO voltage booster

- IO analog switch voltage booster
  - Analog switch resistance increases when VDDA is decreasing
    - For low VDDA, analog switch resistance can be minimized by enabling internal voltage booster to control the switch
  - The recommended analog switch supply is as follows:

VDD	VDDA	BOOSTEN	ANASWVDD	Analog switch supply
-	>2.4 v	0	0	VDDA (default)
>2.4 v	<2.4 v	0	1	VDD
<2.4 v	<2.4 v	1	0	Voltage booster

ANASWVDD
1: I/O analog switches supplied by VDD
0: I/O analog switches supplied by VDDA or booster
BOOSTEN
1: I/O analog switches are supplied by voltage booster
0: I/O analog switches are supplied by VDDA

The analog switch inside the IO has a resistance which increases when the analog switch supply decreases.
So for cases where VDDA and VDD are low, there is a possibility to enable a voltage booster which will supply the analog switch and guarantee low resistance.
The recommended supply for the analog switch is to use VDDA.
But when VDDA is lower than 2.4 Volt and VDD is larger than 2.4 Volt, the power supply can be switched to VDD.
If both VDDA and VDD are lower than 2.4 Volt, the voltage booster should be enabled.



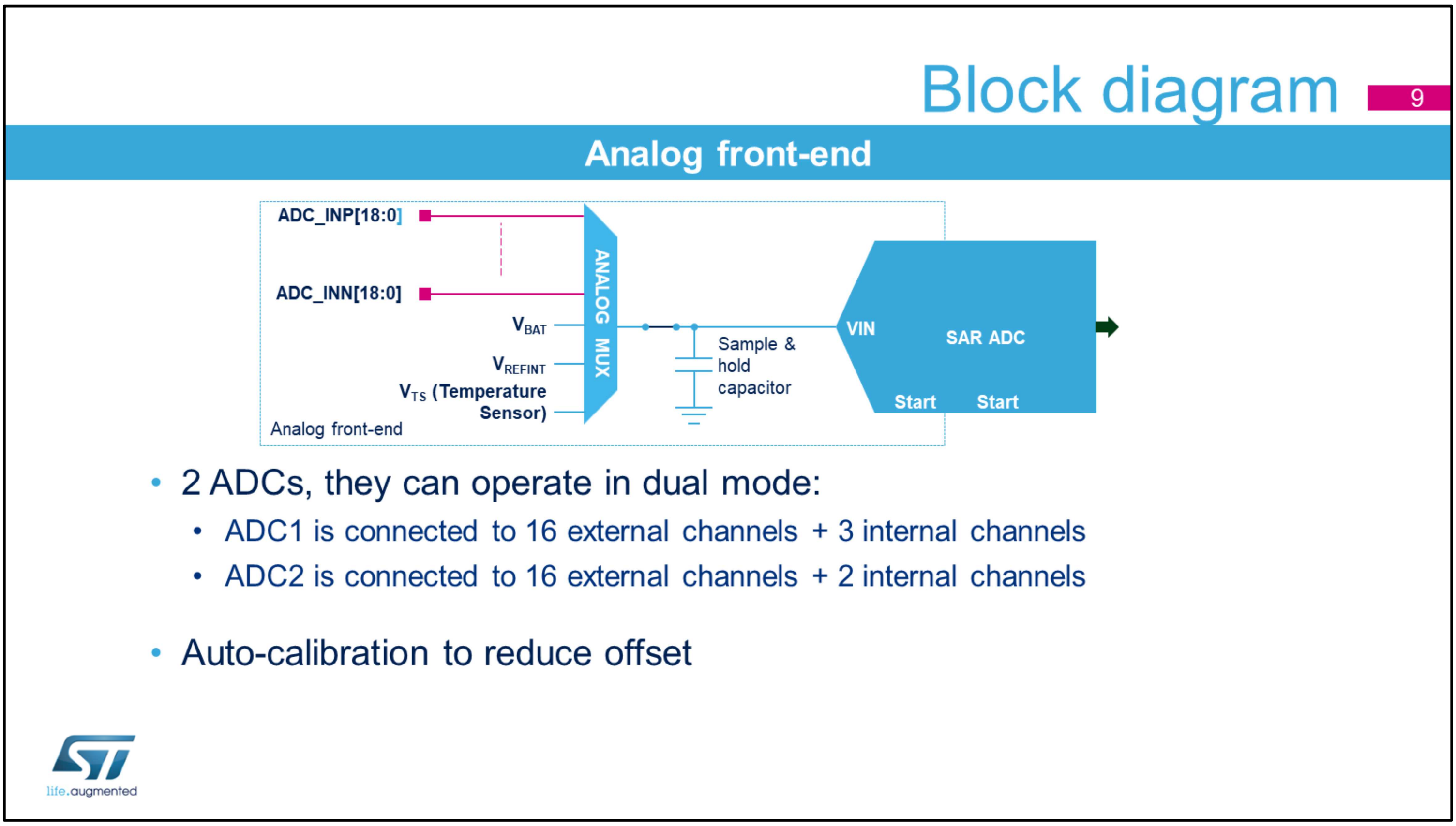
Block diagram

8

Power supplies and on/off control, Internal voltage reference VREFBUF

- Analog voltage reference Vref can be provided either externally or internally through the voltage reference VREFBUF
  - Three voltages are available: 2.048V, 2.5V
- Available only in packages with Vref+ pin available
- When VREFBUF is used, it is necessary to place decoupling capacitors to Vref+ pin for its stability.

It is possible to provide the internal reference voltage Vref internally using the VREFBUF.
Two voltage levels are available.
This can be used only when there is a Vref pin in the package (all packages except 32-pin STM32L531 UFQFPN32 and LQFP32 packages)
When Vref is provided by internal VREFBUF, decoupling capacitors must be placed to the Vref pin.



Block diagram

9

Analog front-end

- 2 ADCs, they can operate in dual mode:
  - ADC1 is connected to 16 external channels + 3 internal channels
  - ADC2 is connected to 16 external channels + 2 internal channels
- Auto-calibration to reduce offset

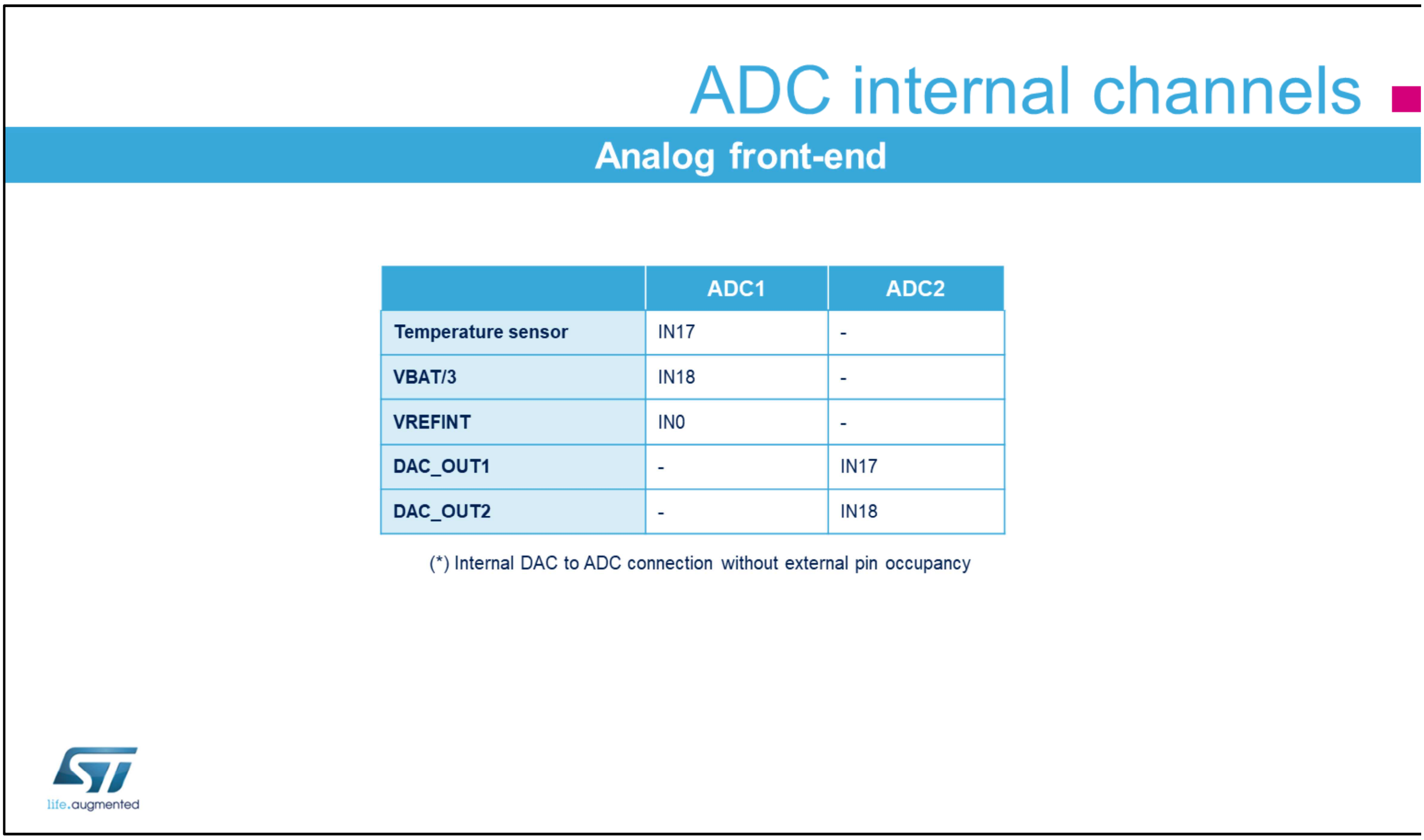
life.augmented

The STM32L5 supports up to 2 ADCs.
Each of them is connected to external analog channels and internal analog sources.
The internal channels are:

- The internal reference voltage (VREFINT)
- The internal temperature sensor (VSENSE)
- The VBAT monitoring channel (VBAT/3)
- DAC_OUT1,2 voltage

Note that analog inputs can be configured to be single-ended or differential.
The ADC offers an auto calibration mechanism.
Calibration is preliminary to any ADC operation. It removes the offset error which may vary from chip to chip due to process, supply voltage or temperature variation.
Single ended inputs and differential inputs are calibrated separately according to the state of the ADCCALDIF bit

value.
Software can request a calibration by setting the ADCAL bit to one.
The resulting calibration factor can be read from ADC_CALFACT register.
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.
High temperature excursion may also require to run the offset calibration.



ADC internal channels

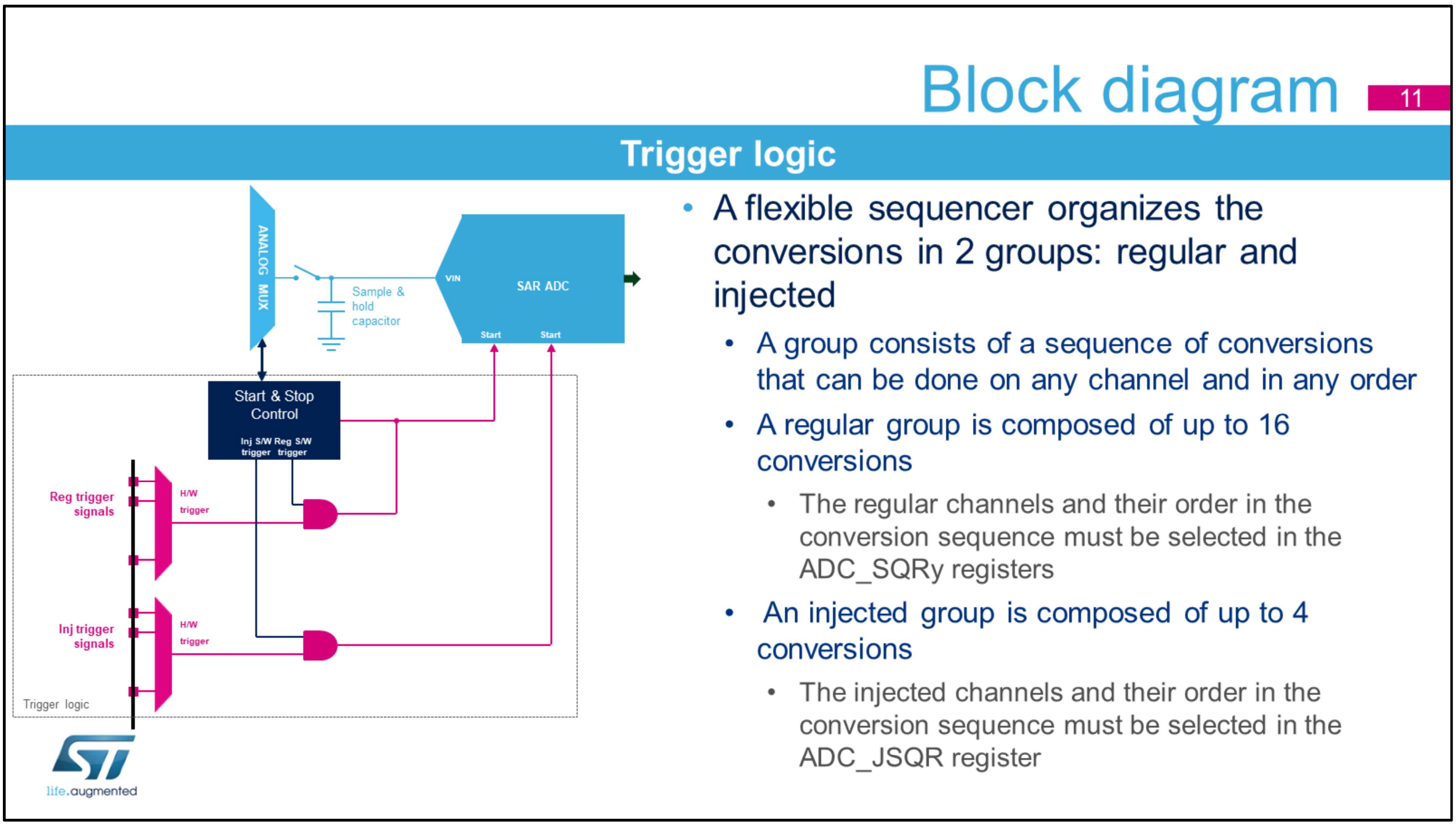
Analog front-end

	ADC1	ADC2
Temperature sensor	IN17	-
VBAT/3	IN18	-
VREFINT	IN0	-
DAC_OUT1	-	IN17
DAC_OUT2	-	IN18

(*) Internal DAC to ADC connection without external pin occupancy

life.augmented

This table indicates how the internal analog channels connected to two ADCs.

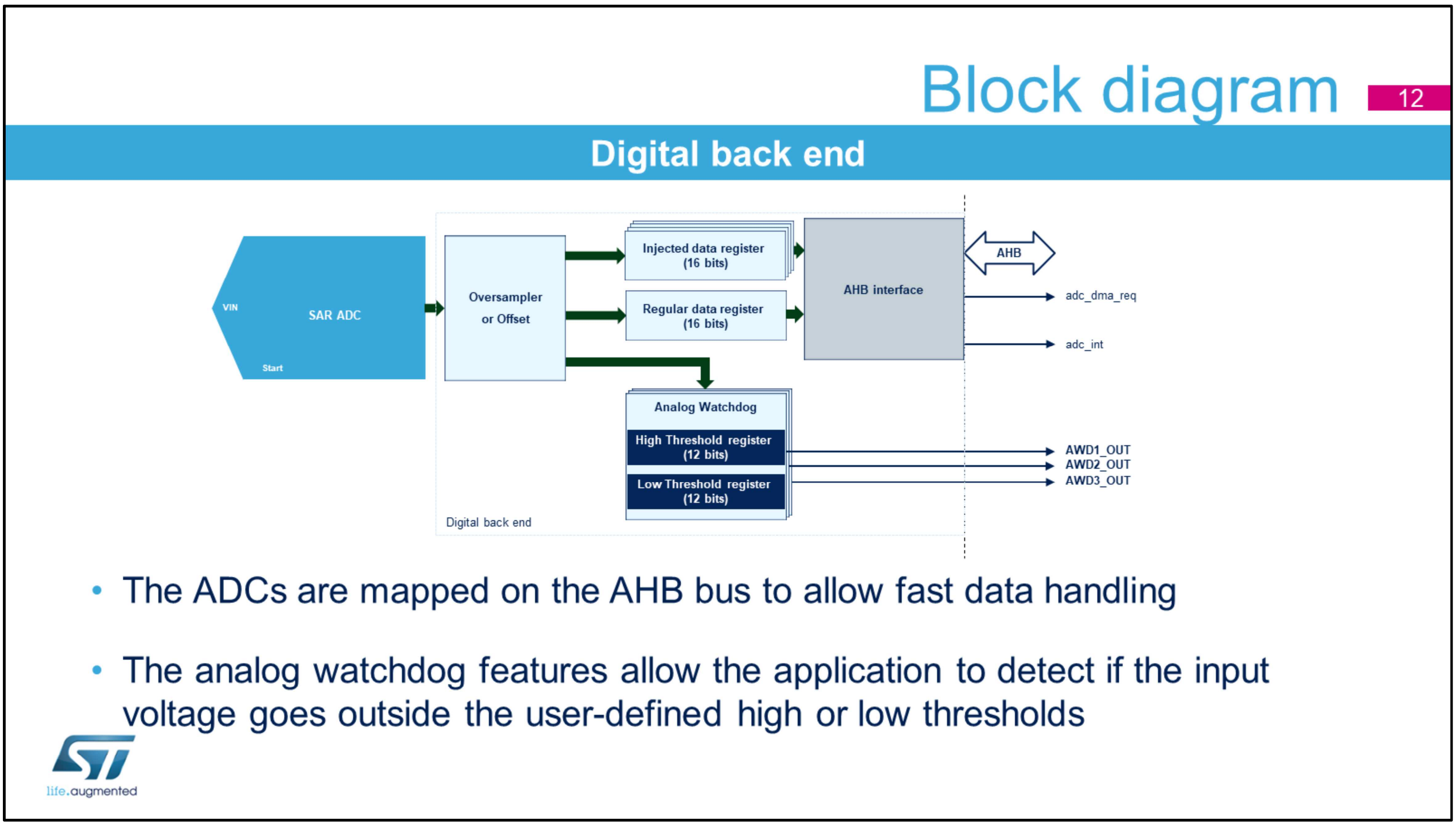


Block diagram

Trigger logic

- A flexible sequencer organizes the conversions in 2 groups: regular and injected
  - A group consists of a sequence of conversions that can be done on any channel and in any order
  - A regular group is composed of up to 16 conversions
    - The regular channels and their order in the conversion sequence must be selected in the ADC_SQRy registers
  - An injected group is composed of up to 4 conversions
    - The injected channels and their order in the conversion sequence must be selected in the ADC_JSQR register

Conversions are organized in two groups: the regular group and the injected group.
The injected group can pre-empt the execution of the regular group sampling sequence.
The user is in charge of selecting the size of each group, maximum 16 acquisitions in the regular group and 4 in the injected group.
They also have to assign the analog channels for each sampling within the sequence.
Each group has its own trigger logic.
The trigger can be an external signal, coming from GPIOs or timer outputs.
A sequence of acquisition can also be triggered by software.



Block diagram

12

Digital back end

VIN
SAR ADC
Start
Oversampler or Offset
Injected data register (16 bits)
Regular data register (16 bits)
AHB interface
AHB
adc_dma_req
adc_int
Analog Watchdog
High Threshold register (12 bits)
Low Threshold register (12 bits)
AWD1_OUT
AWD2_OUT
AWD3_OUT
Digital back end

- The ADCs are mapped on the AHB bus to allow fast data handling
- The analog watchdog features allow the application to detect if the input voltage goes outside the user-defined high or low thresholds

life.augmented

The digital back-end performs processing on the samples obtained in the SAR ADC in the over-sampler & offset unit.

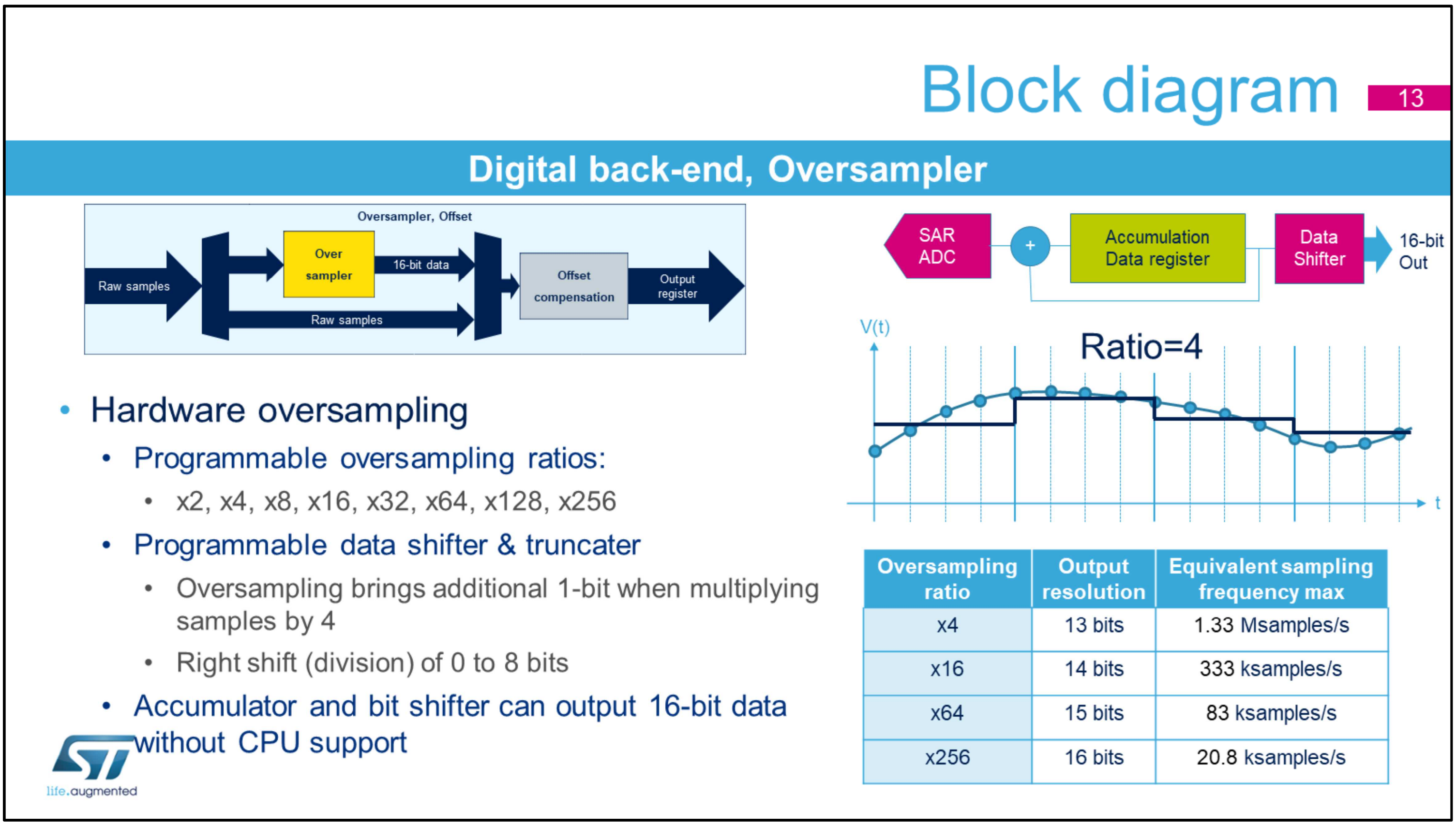
Raw samples maybe processed by the oversampler and offset compensation units before being provided to the software.

The results are then stored into registers, that are accessible from the AHB slave interface.

AHB supports a higher bandwidth and minimizes latency, because the CPU and DMA are also connected to the AHB interconnect.

The ADC is able to assert a request to the DMA, so that samples coming from regular channels are automatically moved to memory.

Injected channels do not support DMA requests.



Block diagram

13

Digital back-end, Oversampler

- Hardware oversampling
  - Programmable oversampling ratios:
    - x2, x4, x8, x16, x32, x64, x128, x256
  - Programmable data shifter & truncater
    - Oversampling brings additional 1-bit when multiplying samples by 4
    - Right shift (division) of 0 to 8 bits
  - Accumulator and bit shifter can output 16-bit data without CPU support

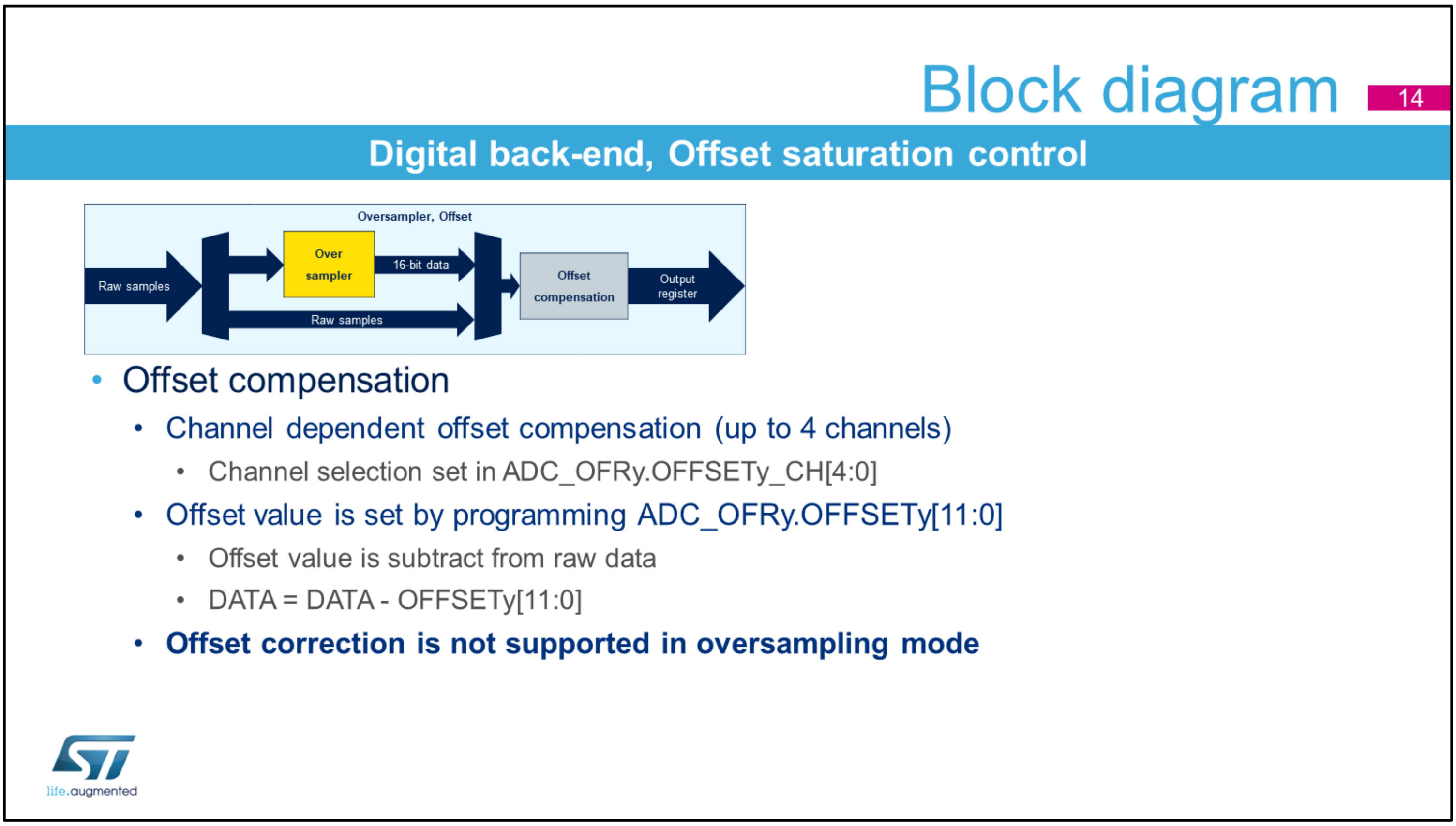
Oversampling ratio	Output resolution	Equivalent sampling frequency max
x4	13 bits	1.33 Msamples/s
x16	14 bits	333 ksamples/s
x64	15 bits	83 ksamples/s
x256	16 bits	20.8 ksamples/s

life.augmented

The ADC includes the oversampling hardware which accumulates data and then divides without CPU assistance.

The oversampler can accommodate from 2 to 256 time samples and right shift from one to eight binary digits. 12-bit data can be extended to be presented as 16-bit data.

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.



Block diagram

14

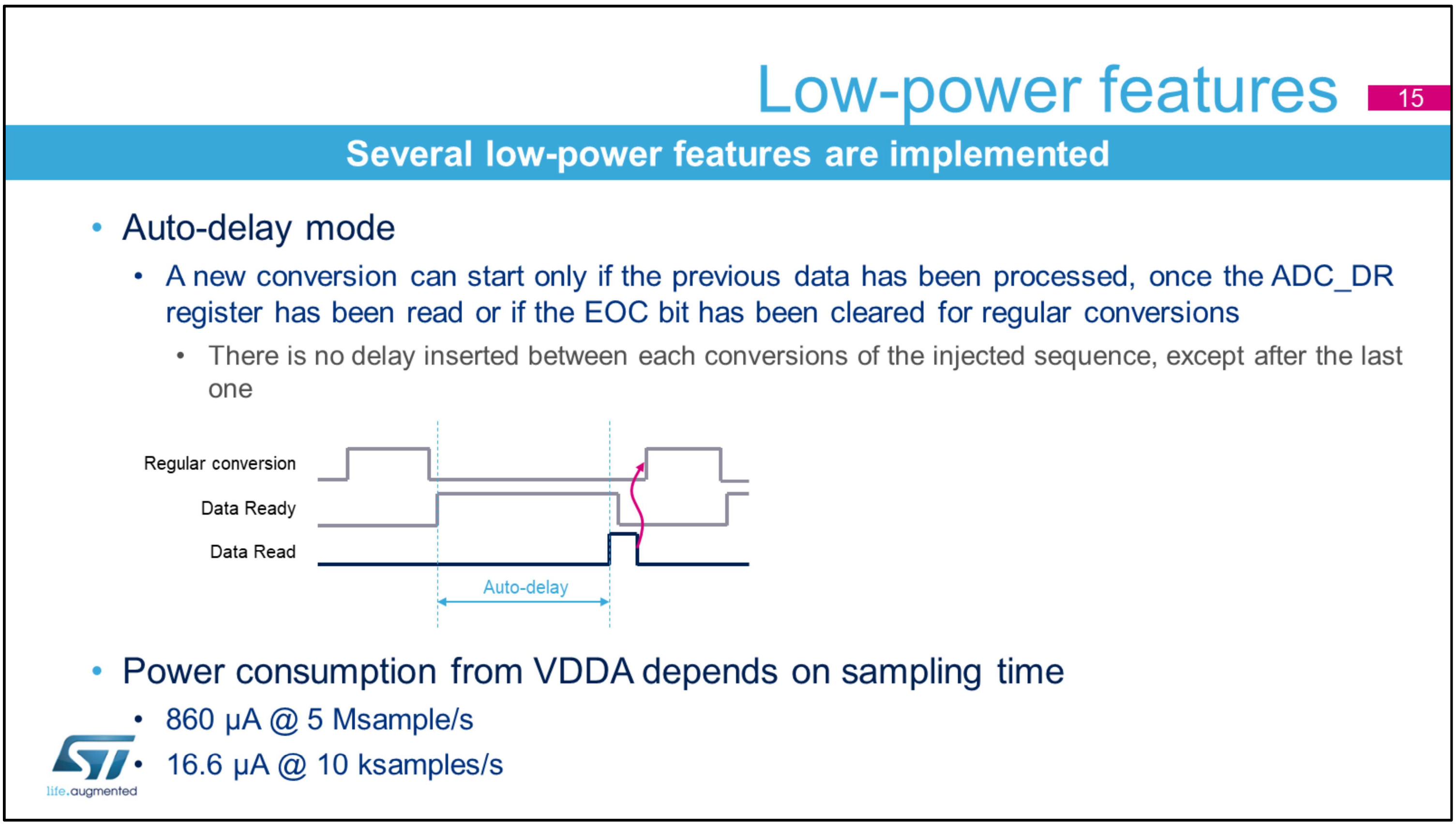
Digital back-end, Offset saturation control

Oversampler, Offset
Raw samples
Over sampler
16-bit data
Raw samples
Offset compensation
Output register

- Offset compensation
  - Channel dependent offset compensation (up to 4 channels)
    - Channel selection set in ADC_OFRy.OFFSETy_CH[4:0]
  - Offset value is set by programming ADC_OFRy.OFFSETy[11:0]
    - Offset value is subtract from raw data
    - DATA = DATA - OFFSETy[11:0]
  - **Offset correction is not supported in oversampling mode**

life.augmented

An offset y (y=1,2,3,4) can be applied to a channel by setting the bit OFFSETy_EN=1 into ADC_OFRy register. The channel to which the offset will be applied is programmed into the bits OFFSETy_CH[4:0] of ADC_OFRy register.
In this case, the converted value is decreased by the user-defined offset written in the bits OFFSETy[11:0]. The result may be a negative value so the read data is signed.



Low-power features

15

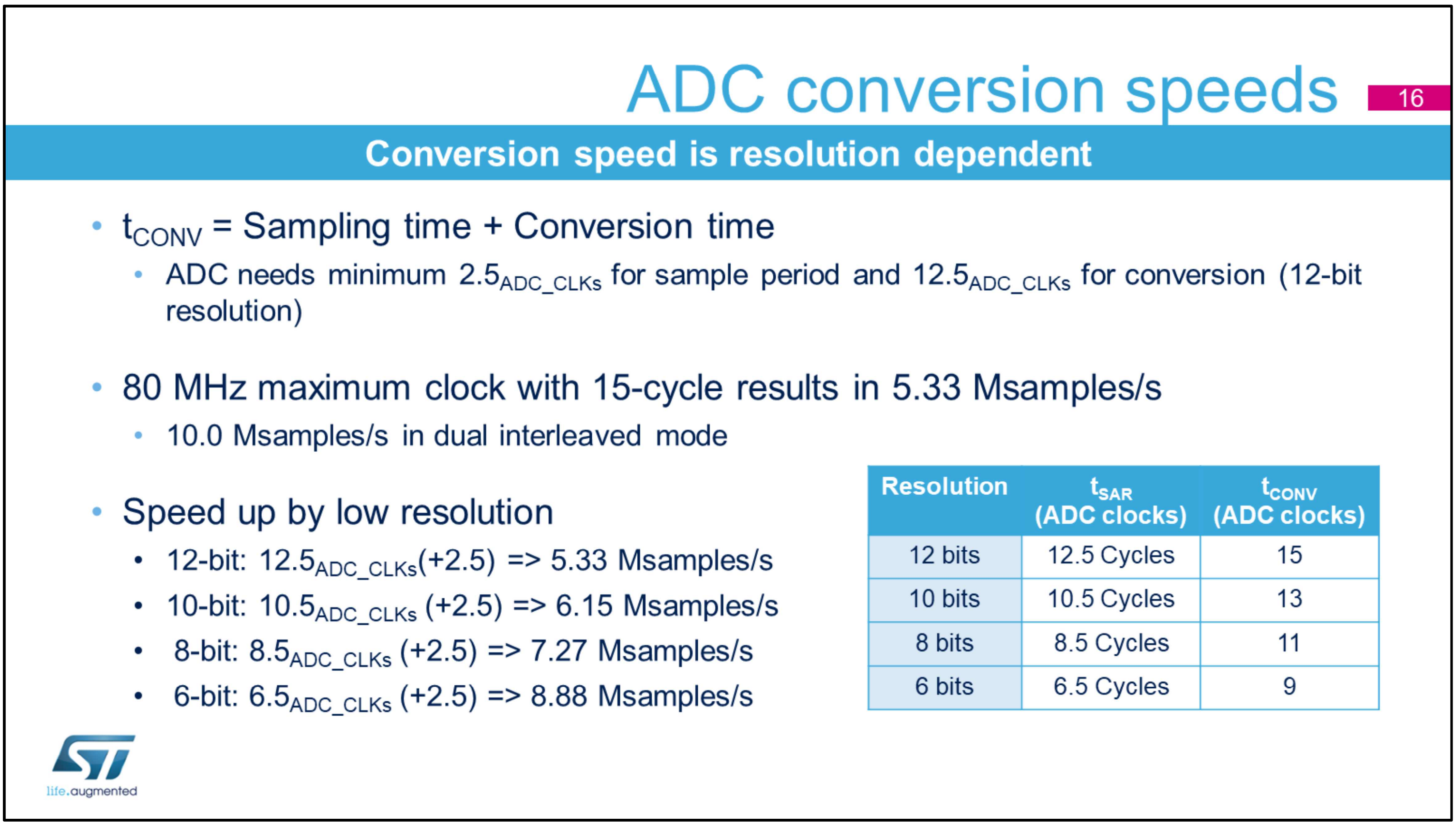
Several low-power features are implemented

- Auto-delay 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 for regular conversions
    - There is no delay inserted between each conversions of the injected sequence, except after the last one

Regular conversion
Data Ready
Data Read
Auto-delay
life.augmented

- Power consumption from VDDA depends on sampling time
  - 860 µA @ 5 Msample/s
  - 16.6 µA @ 10 ksamples/s

When auto-delay 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 is a way to automatically adapt the speed of the ADC to the speed of the system which will read the data.
This also avoids unnecessary conversions and thus reduces power consumption.
This Auto-delay mode does not apply to Injected conversions, except the last one when switching back to the regular conversions.
Power consumption from VDDA power supply depends on sampling time: from 16.6 µA at 10 Kilo samples per second to 860 µA at 5 mega samples per second.
For low sampling rates, the current consumption is reduced almost proportionally.



ADC conversion speeds

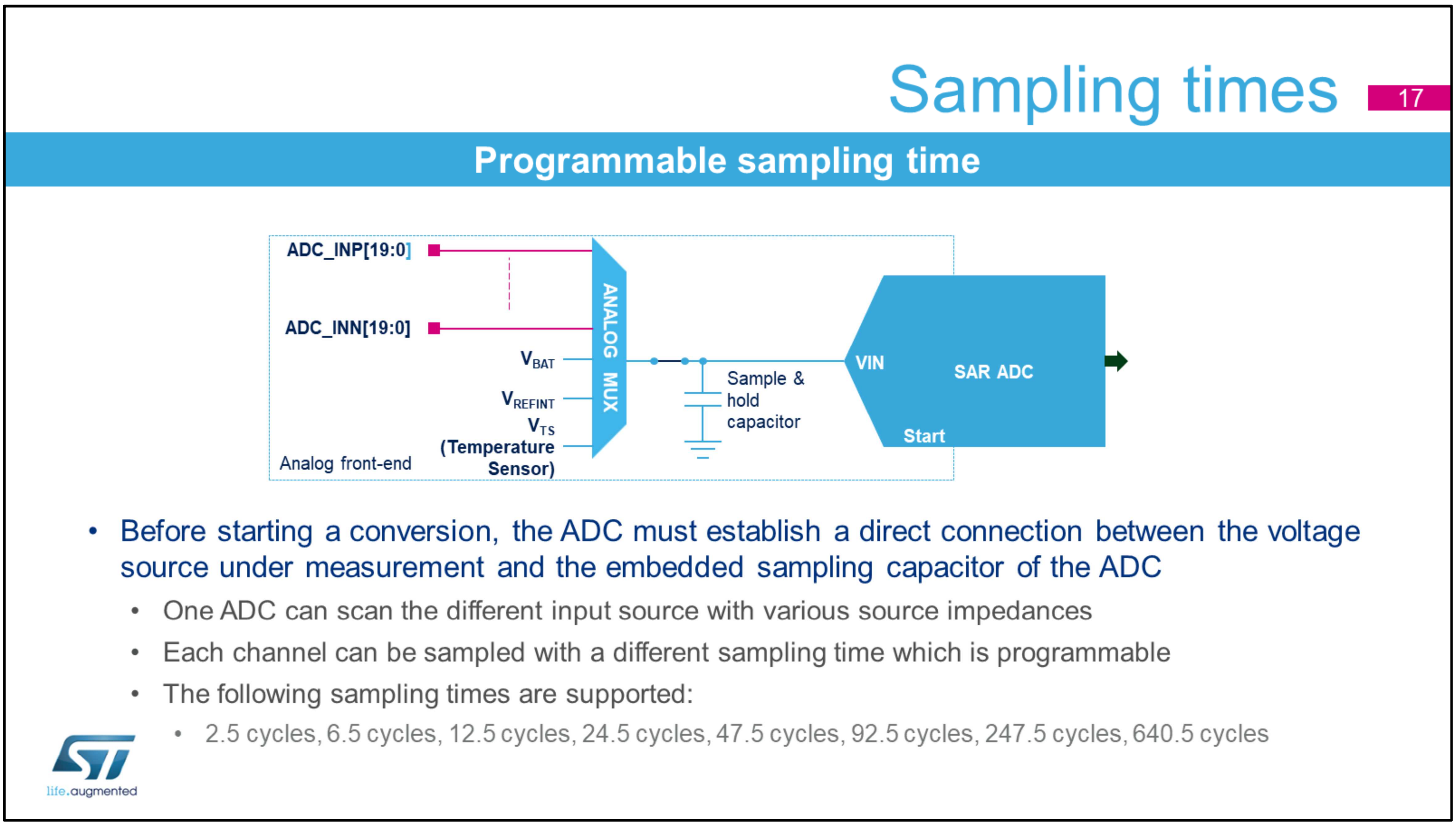
Conversion speed is resolution dependent

- t_{CONV} = Sampling time + Conversion time
  - ADC needs minimum 2.5_{ADC_CLKs} for sample period and 12.5_{ADC_CLKs} for conversion (12-bit resolution)
- 80 MHz maximum clock with 15-cycle results in 5.33 Msamples/s
  - 10.0 Msamples/s in dual interleaved mode
- Speed up by low resolution
  - 12-bit: 12.5_{ADC_CLKs}(+2.5) => 5.33 Msamples/s
  - 10-bit: 10.5_{ADC_CLKs} (+2.5) => 6.15 Msamples/s
  - 8-bit: 8.5_{ADC_CLKs} (+2.5) => 7.27 Msamples/s
  - 6-bit: 6.5_{ADC_CLKs} (+2.5) => 8.88 Msamples/s

Resolution	t_{SAR} (ADC clocks)	t_{CONV} (ADC clocks)
12 bits	12.5 Cycles	15
10 bits	10.5 Cycles	13
8 bits	8.5 Cycles	11
6 bits	6.5 Cycles	9

life.augmented

The global conversion time is equal to the sampling time plus the conversion time.
The ADC needs a minimum of 2.5 clock cycles for the sampling and 12.5 clock cycles for conversion for 12-bit mode.
With a 80 MHz ADC clock, it can achieve 5.33 mega sample per second.
For a higher sampling speed, it is possible to reduce the resolution down to 10, 8 or 6 bits.
When associating a master and a slave ADC, the interleaved mode maximum performance is one sample every 8 clocks, so 10 Mega sample per second at 80 MHz clock.



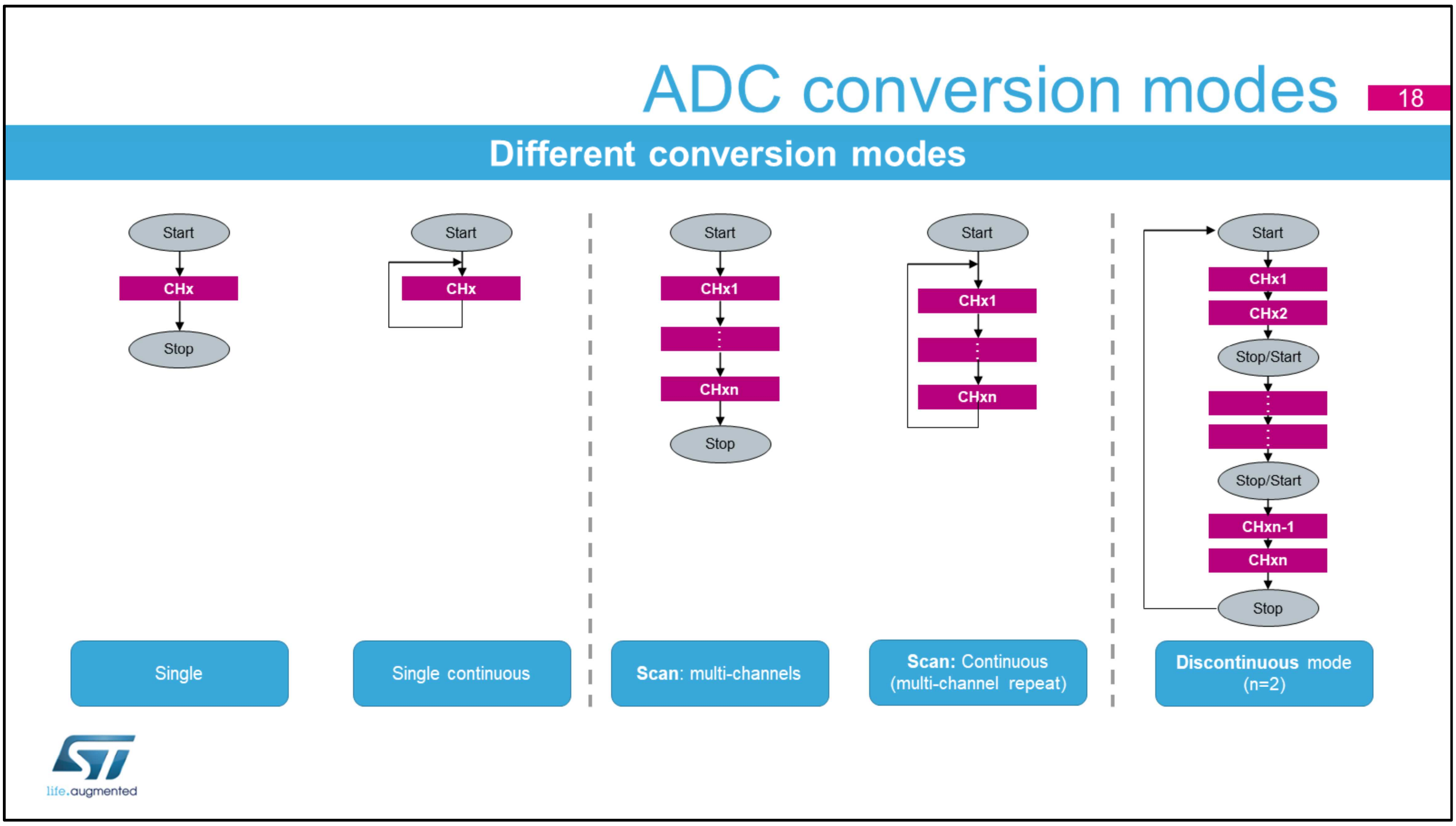
Sampling times

Programmable sampling time

- Before starting a conversion, the ADC must establish a direct connection between the voltage source under measurement and the embedded sampling capacitor of the ADC
  - One ADC can scan the different input source with various source impedances
  - Each channel can be sampled with a different sampling time which is programmable
  - The following sampling times are supported:
    - 2.5 cycles, 6.5 cycles, 12.5 cycles, 24.5 cycles, 47.5 cycles, 92.5 cycles, 247.5 cycles, 640.5 cycles

life.augmented

The first set of a conversion consists in loading the Sample & Hold capacitor with the voltage to be measured.
Longer sample times ensure that signals having a higher impedance are correctly converted.
The sampling times listed in this slide in ADC clock cycles are available.
The sampling time can be programmed individually for each input channel of the analog-to-digital converters.



ADC conversion modes

18

Different conversion modes

Single

Single continuous

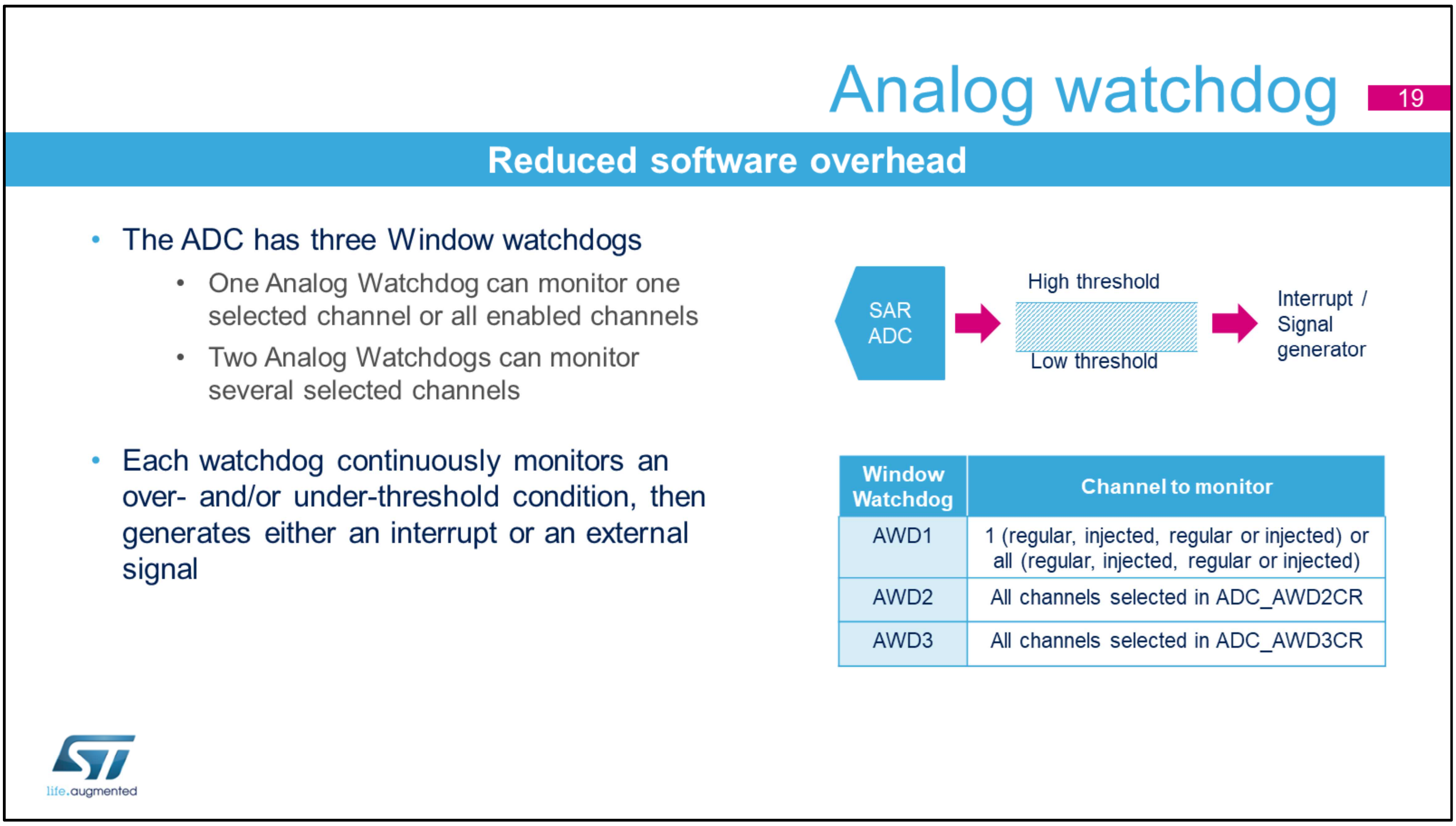
Scan: multi-channels

Scan: Continuous (multi-channel repeat)

Discontinuous mode (n=2)

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 a a short sequence (subgroup) of n conversions (n ≤ 8) that is part of the sequence of conversions. When an external trigger occurs, it starts the next n conversions selected in the ADC_SQR registers until all the conversions in the sequence are done



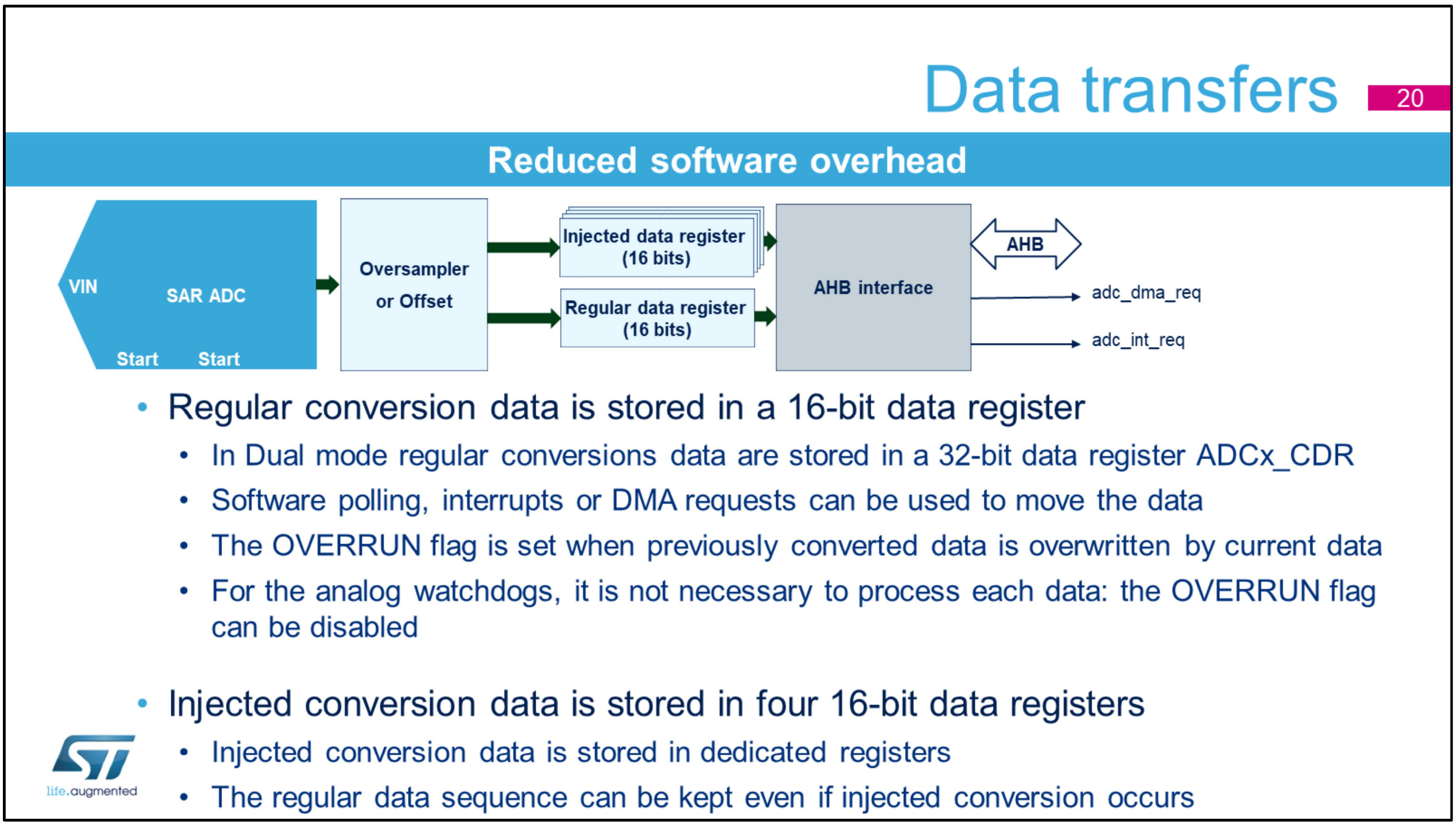
Analog watchdog

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 (regular, injected, regular or injected) or all (regular, injected, regular or injected)
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.



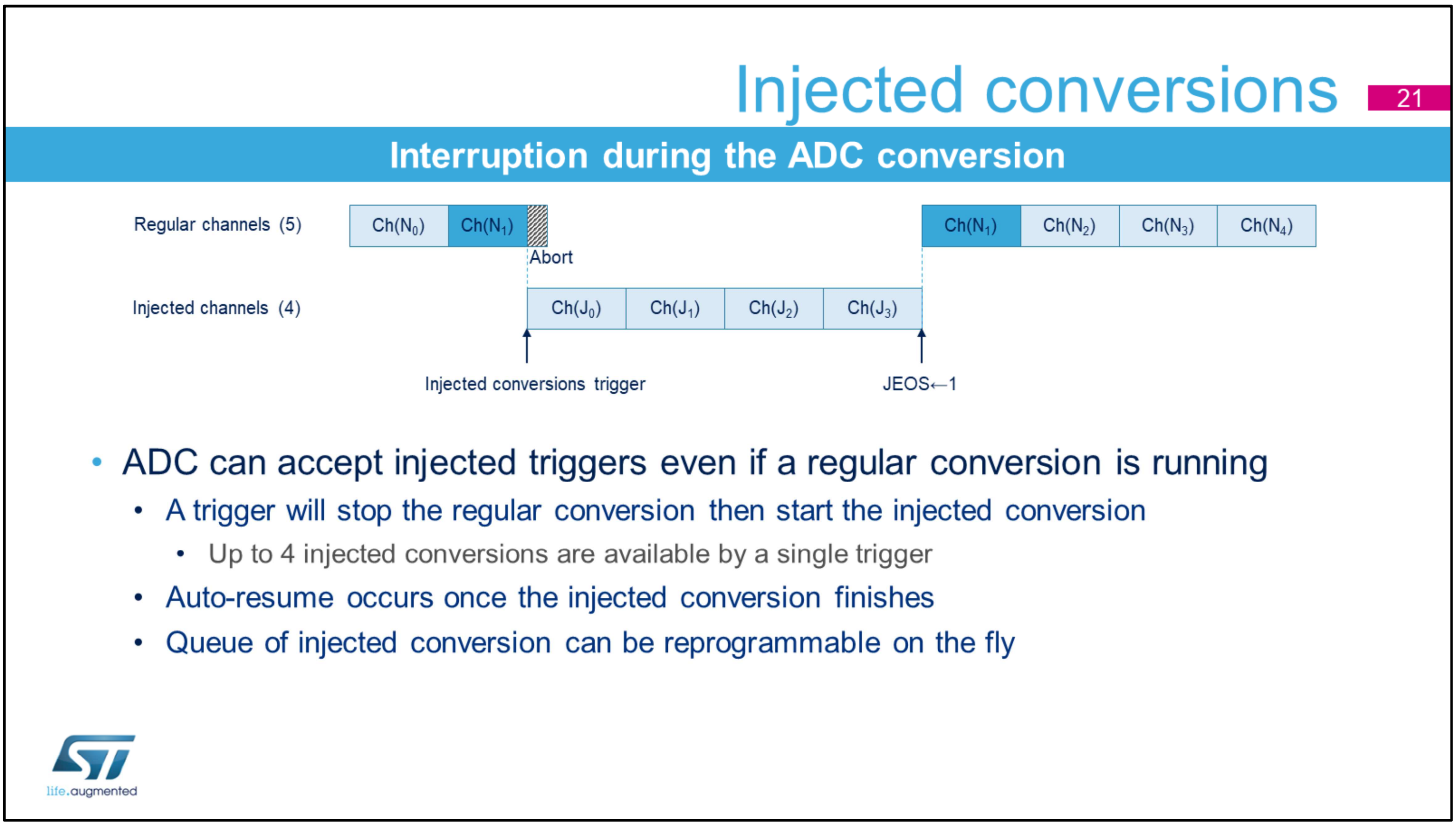
Data transfers

Reduced software overhead

- Regular conversion data is stored in a 16-bit data register
  - In Dual mode regular conversions data are stored in a 32-bit data register ADCx_CDR
  - 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.
In Dual mode, two samples are combined into a 32-bit register called ADCx_CDR, thus minimizing the number of transactions on AHB.
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.
For injected channel conversions, 4 dedicated data registers are available.



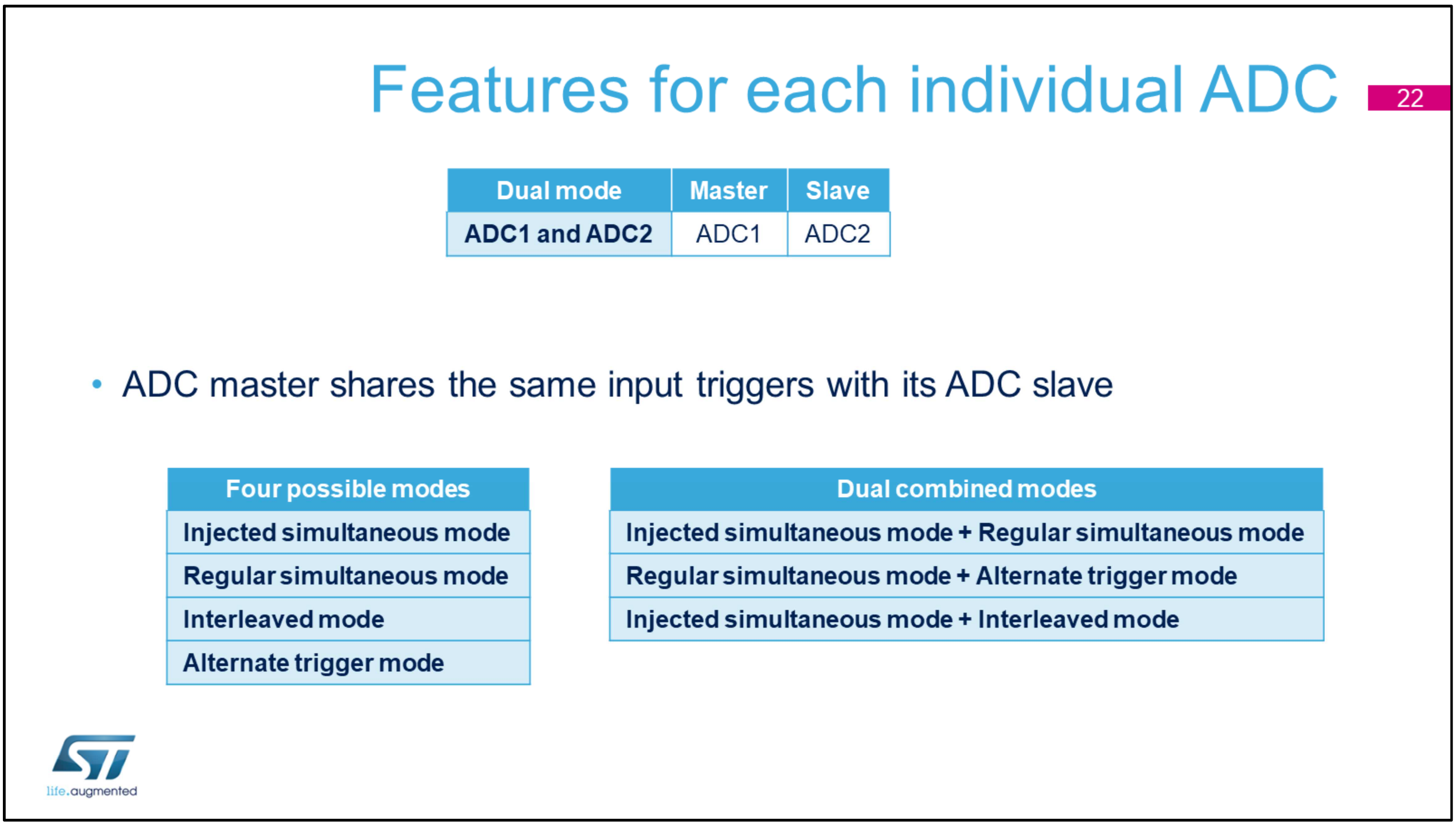
Injected conversions

Interruption during 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
  - 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 and 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.



Features for each individual ADC

Dual mode	Master	Slave
ADC1 and ADC2	ADC1	ADC2

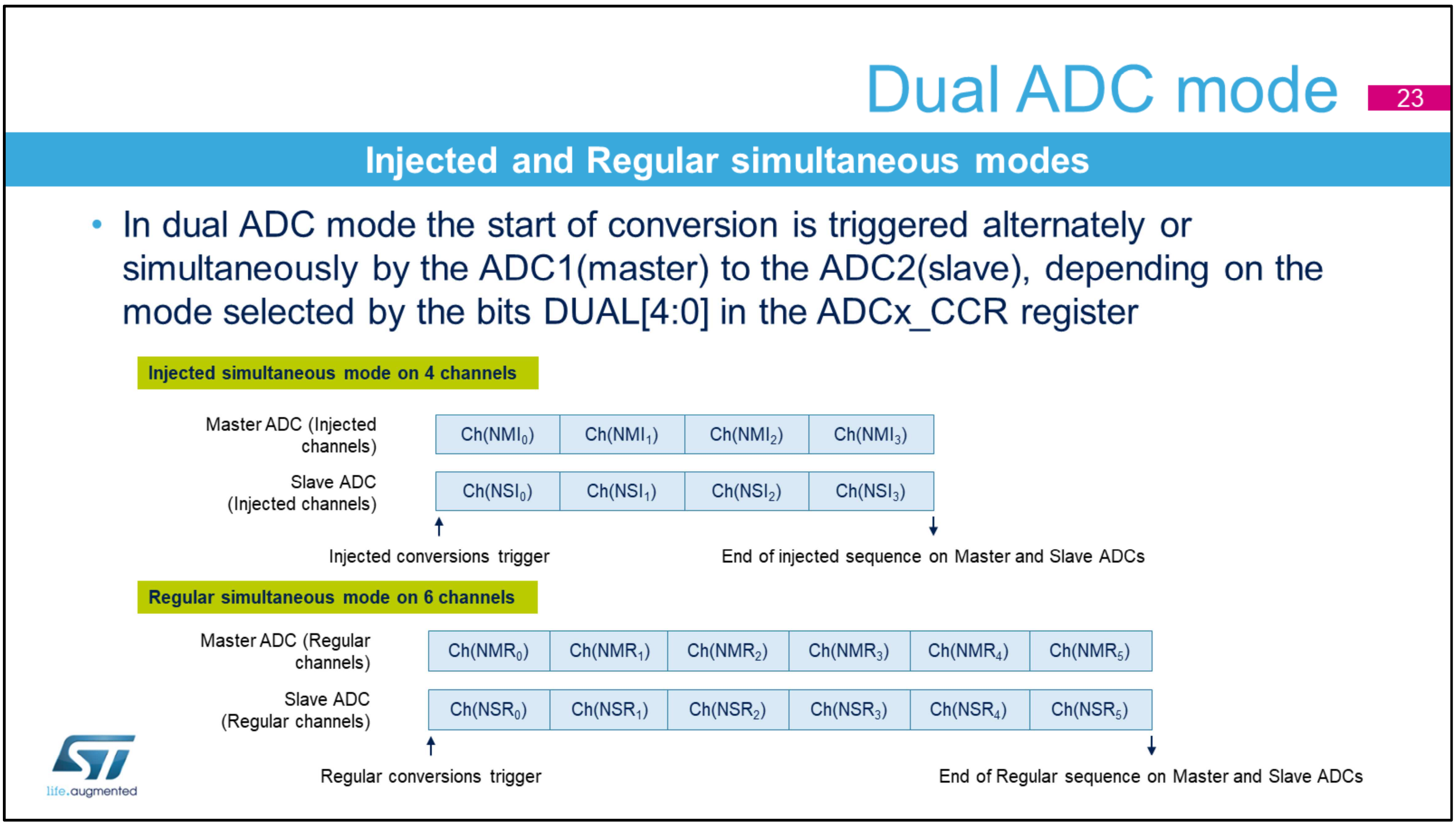
- ADC master shares the same input triggers with its ADC slave

Four possible modes
Injected simultaneous mode
Regular simultaneous mode
Interleaved mode
Alternate trigger mode

Dual combined modes
Injected simultaneous mode + Regular simultaneous mode
Regular simultaneous mode + Alternate trigger mode
Injected simultaneous mode + Interleaved mode

The STM32L5 embeds two ADCs.
ADC1 and ADC2 can be configured to work together in Dual mode, so that each analog-to-digital conversion can be synchronized between the two modules.
Four possible Dual ADC modes are implemented:
- Injected simultaneous mode
- Regular simultaneous mode
- Interleaved mode
- Alternate trigger mode

It is also possible to use these modes combined in the following ways:
- Injected simultaneous mode + Regular simultaneous mode
- Regular simultaneous mode + Alternate trigger mode
- Injected simultaneous mode + Interleaved mode.



Dual ADC mode

Injected and Regular simultaneous modes

- In dual ADC mode the start of conversion is triggered alternately or simultaneously by the ADC1(master) to the ADC2(slave), depending on the mode selected by the bits DUAL[4:0] in the ADCx_CCR register

Injected simultaneous mode on 4 channels

Master ADC (Injected channels)	Ch(NMI$_0$)	Ch(NMI$_1$)	Ch(NMI$_2$)	Ch(NMI$_3$)
Slave ADC (Injected channels)	Ch(NSI$_0$)	Ch(NSI$_1$)	Ch(NSI$_2$)	Ch(NSI$_3$)

Injected conversions trigger

End of injected sequence on Master and Slave ADCs

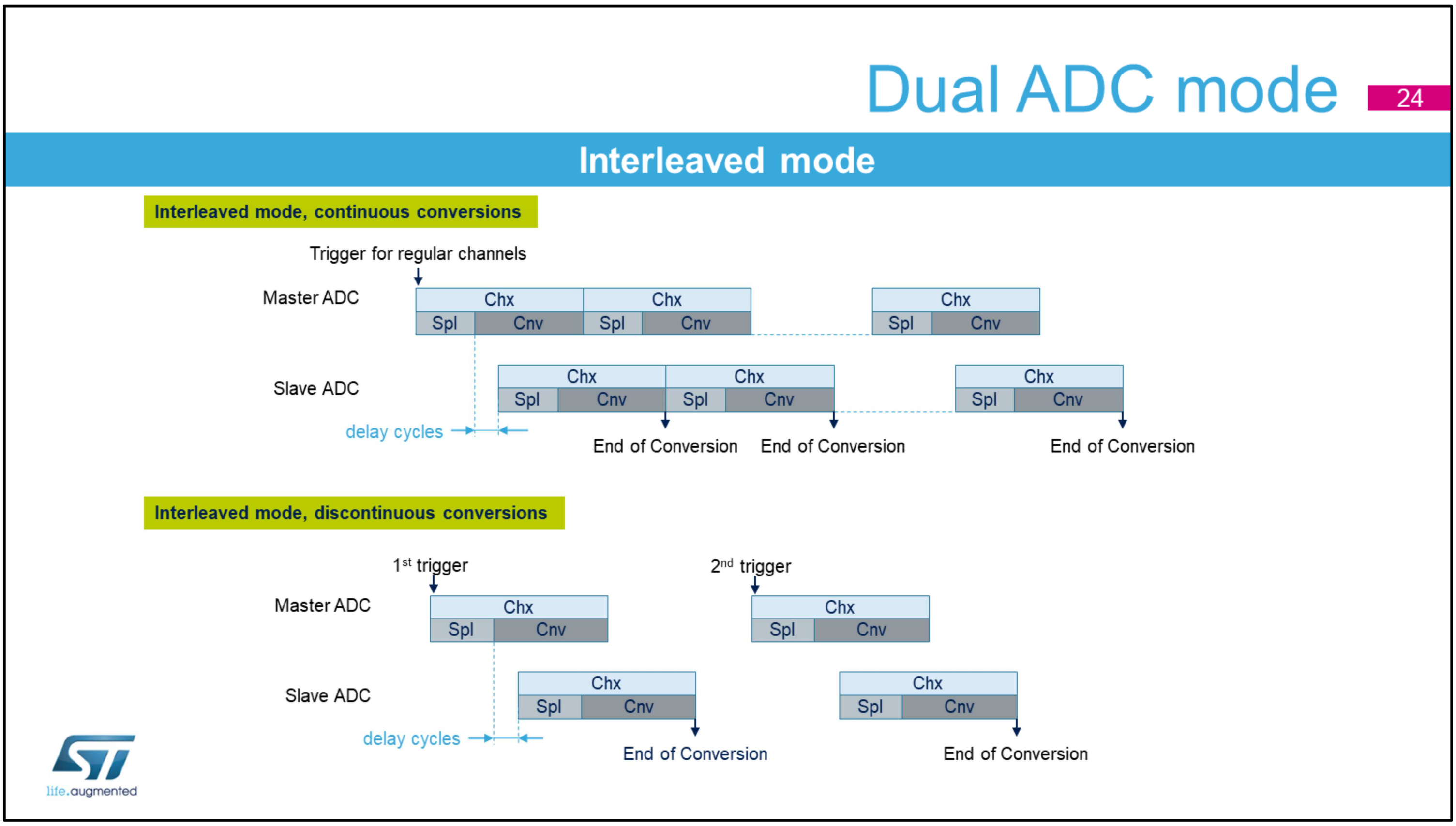
Regular simultaneous mode on 6 channels

Master ADC (Regular channels)	Ch(NMR$_0$)	Ch(NMR$_1$)	Ch(NMR$_2$)	Ch(NMR$_3$)	Ch(NMR$_4$)	Ch(NMR$_5$)
Slave ADC (Regular channels)	Ch(NSR$_0$)	Ch(NSR$_1$)	Ch(NSR$_2$)	Ch(NSR$_3$)	Ch(NSR$_4$)	Ch(NSR$_5$)

Regular conversions trigger

End of Regular sequence on Master and Slave ADCs

In dual ADC mode, conversions can be started either simultaneously or alternately on ADC master and ADC slave.
The converted data of the master and slave ADC can be read in parallel, by reading the ADC common data register (ADCx_CDR).
Do not convert the same channel on the two ADCs.
This slide describes the injected simultaneous mode and the regular simultaneous mode.
The trigger is used to simultaneously start the sequence of conversions on both Master and Slave ADC.
Conversion sequences must be equal on master and slave or must ensure that the interval between triggers is longer than both sequences.
In discontinuous mode, every simultaneous conversion requires an injected trigger.



Dual ADC mode

Interleaved mode

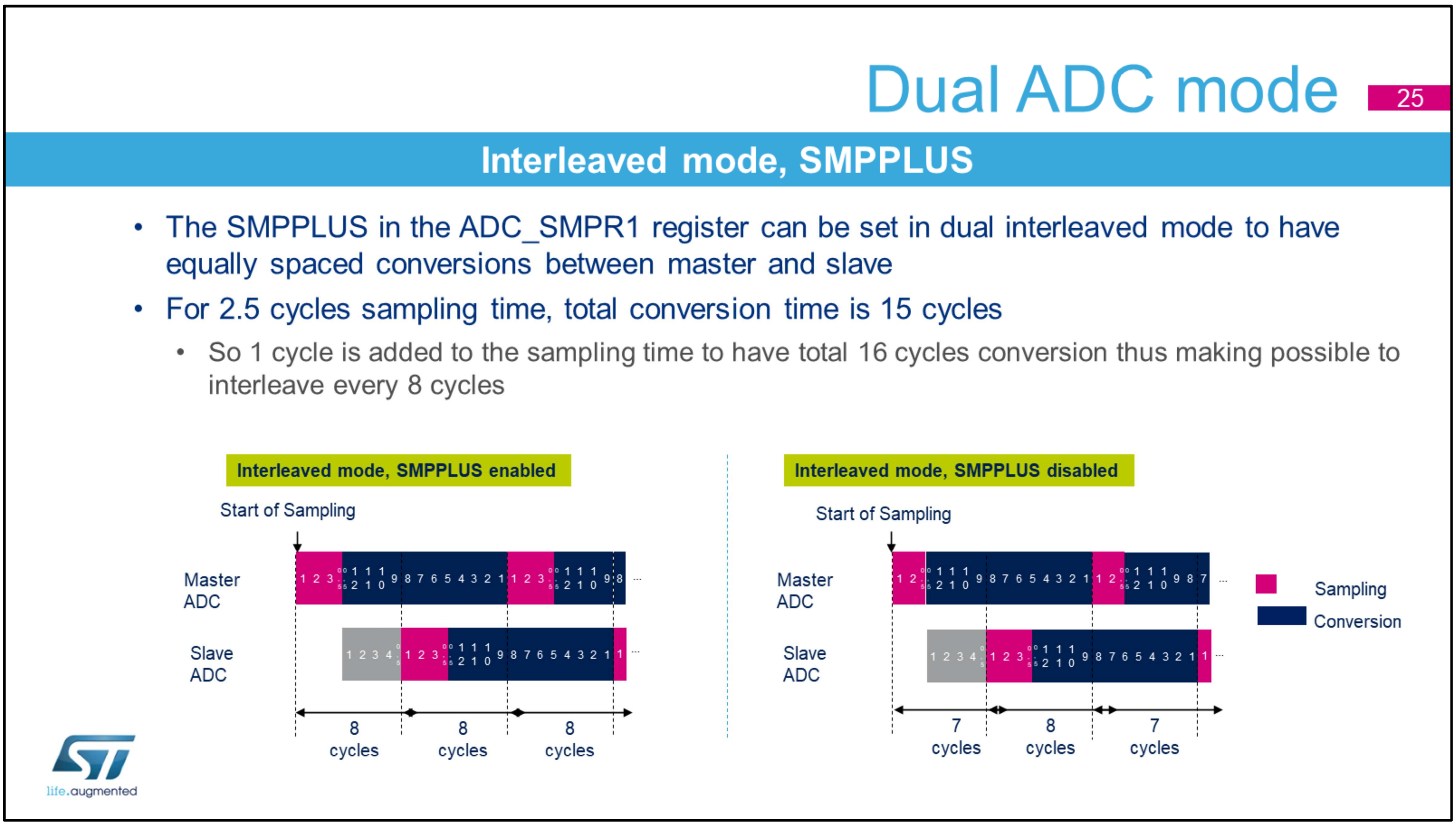
Interleaved mode, continuous conversions

Interleaved mode, discontinuous conversions

Interleaved mode converts a regular channel group (usually one channel).
The external trigger source which starts the conversion comes from ADC Master:

- ADC Master starts immediately
- ADC Slave starts after a configurable delay after the end of sampling of the master.

It prevents an ADC from starting a conversion while the complementary ADC is still sampling the input.
An EOC is generated at the end of each channel conversion.
In discontinuous mode, every simultaneous conversion requires a regular trigger.



Dual ADC mode

Interleaved mode, SMPPLUS

- The SMPPLUS in the ADC_SMPR1 register can be set in dual interleaved mode to have equally spaced conversions between master and slave
- For 2.5 cycles sampling time, total conversion time is 15 cycles
  - So 1 cycle is added to the sampling time to have total 16 cycles conversion thus making possible to interleave every 8 cycles

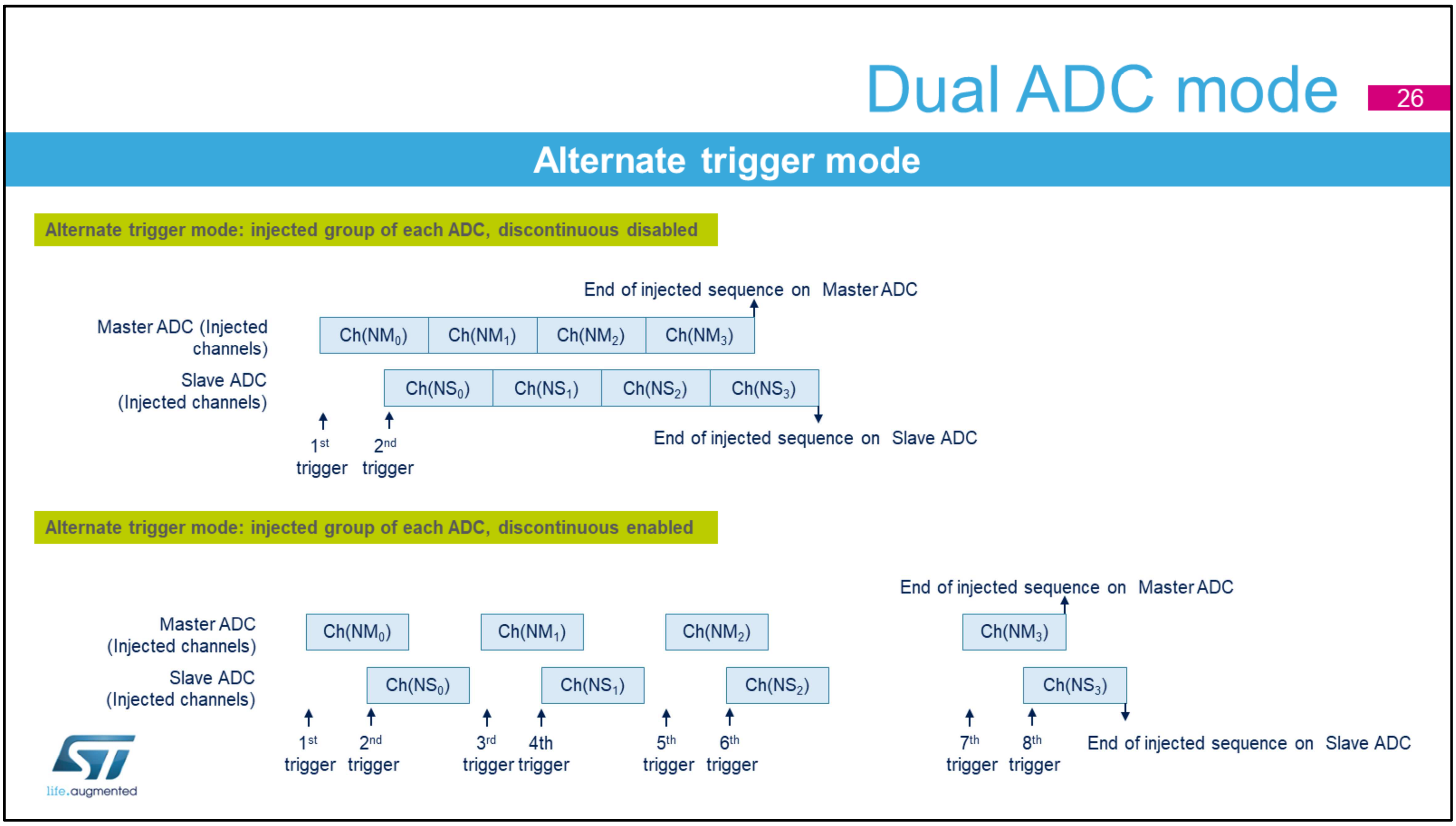
When a sampling time of 2.5 ADC clock cycles is selected, the total conversion time becomes 15 cycles in 12-bit mode.

If the dual interleaved mode is used, the sampling interval cannot be equal to 2.5 ADC clock cycles since an even number of cycles is required for the sampling time plus conversion time.

In the timing diagram on the right, the sampling time on the slave ADC has to be increased to 3.5 clock cycles, while the sampling time for the master ADC is 2.5 clock cycles.

The SMPPLUS bit can be used to change the sampling time 2.5 ADC clock cycles into 3.5 ADC clock cycles. In this way, the total conversion time becomes 16 clock cycles, thus making possible to interleave every 8 cycles.

The maximum number of samples per second is equal to 80 MHz divided by 8 = 10 Mega samples per second.



Dual ADC mode

26

Alternate trigger mode

Alternate trigger mode: injected group of each ADC, discontinuous disabled

End of injected sequence on Master ADC

Master ADC (Injected channels)

Ch(NM_0) Ch(NM_1) Ch(NM_2) Ch(NM_3)

Slave ADC (Injected channels)

Ch(NS_0) Ch(NS_1) Ch(NS_2) Ch(NS_3)

1st trigger 2nd trigger

End of injected sequence on Slave ADC

Alternate trigger mode: injected group of each ADC, discontinuous enabled

End of injected sequence on Master ADC

Master ADC (Injected channels)

Ch(NM_0) Ch(NM_1) Ch(NM_2) Ch(NM_3)

Slave ADC (Injected channels)

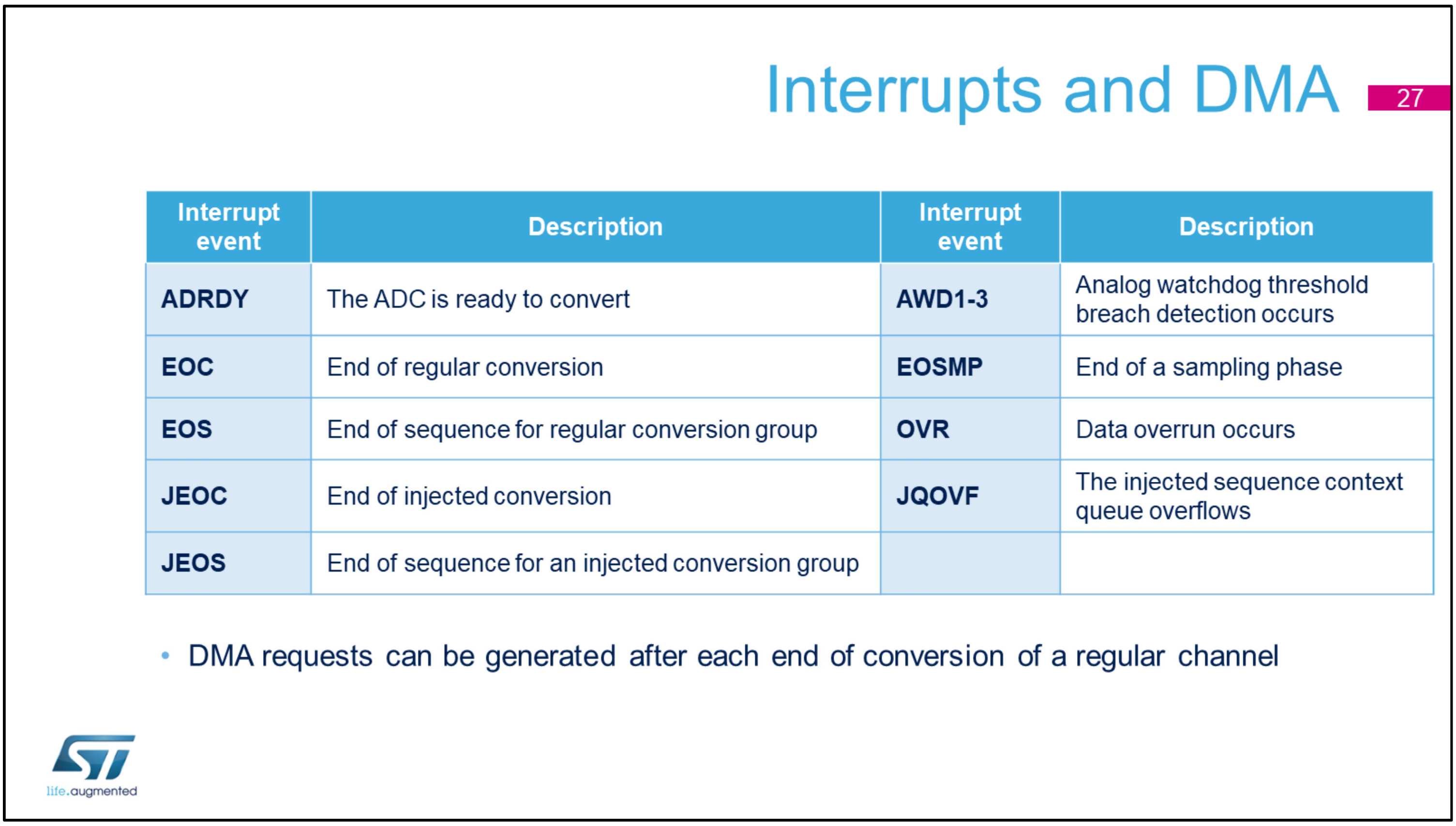
Ch(NS_0) Ch(NS_1) Ch(NS_2) Ch(NS_3)

1st trigger 2nd trigger 3rd trigger 4th trigger 5th trigger 6th trigger 7th trigger 8th trigger

End of injected sequence on Slave ADC

life.augmented

The Alternate Trigger mode converts an injected group of channels.
Conversions are started only by using hardware triggers.
The external trigger source comes from the injected group multiplexer of the master ADC.
When discontinuous mode is disabled and the 1st trigger occurs, all injected master ADC channels in the group are converted.
When the 2nd trigger occurs, all injected slave ADC channels in the group are converted.
When discontinuous mode is enabled and the 1st trigger occurs, the first injected channel of the master ADC is converted.
When the 2nd trigger occurs, the first injected channel of the slave ADC is converted.

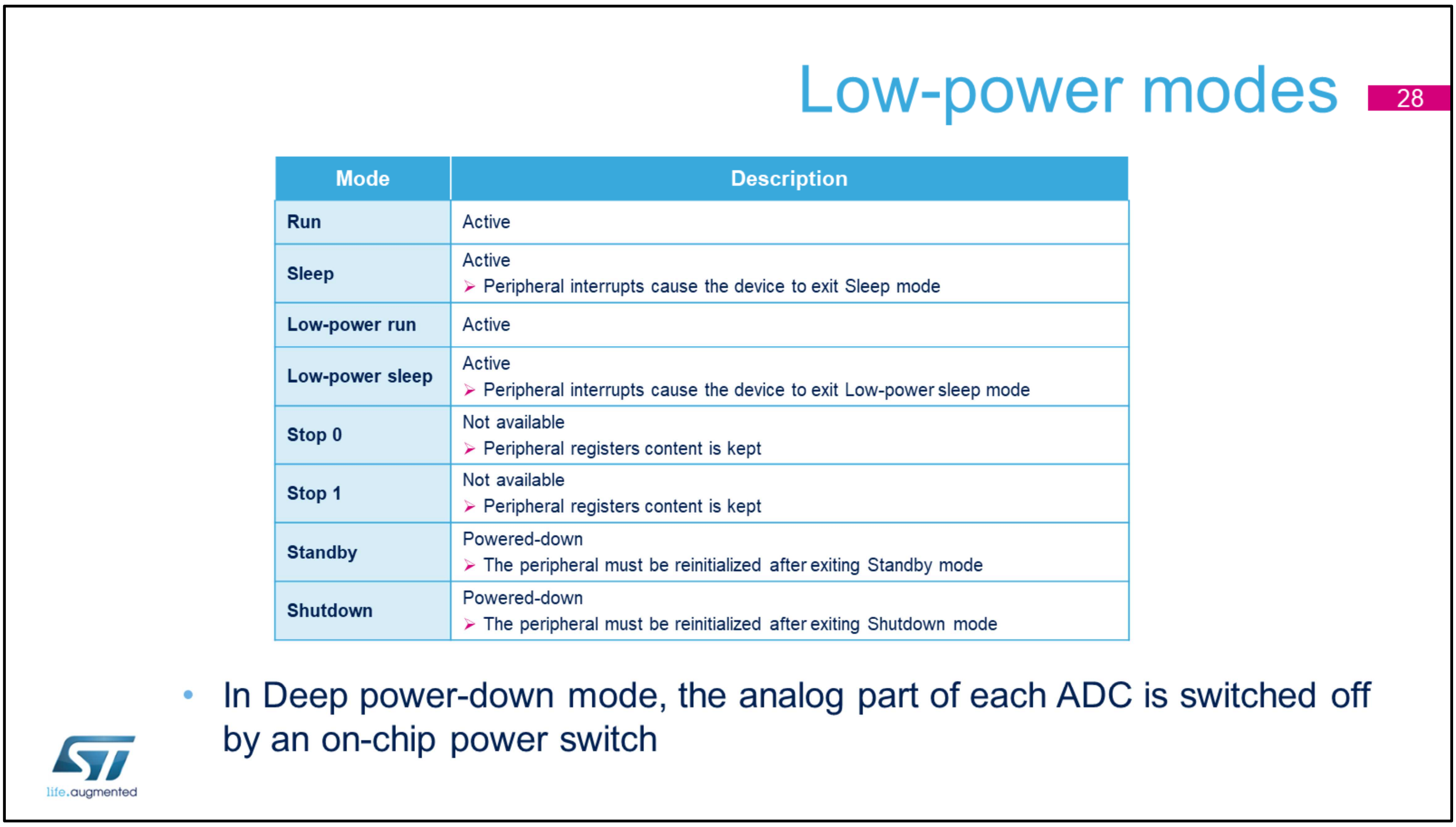


Interrupts and DMA

Interrupt event	Description	Interrupt event	Description
ADRDY	The ADC is ready to convert	**AWD1-3**	Analog watchdog threshold breach detection occurs
EOC	End of regular conversion	**EOSMP**	End of a sampling phase
EOS	End of sequence for regular conversion group	**OVR**	Data overrun occurs
JEOC	End of injected conversion	**JQOVF**	The injected sequence context queue overflows
JEOS	End of sequence for an injected conversion group		

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

Each ADC can generate 9 different interrupts: ADC Ready, end of conversion, end of sequence, end of injected conversion, end of injected sequence, analog watchdog, 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.

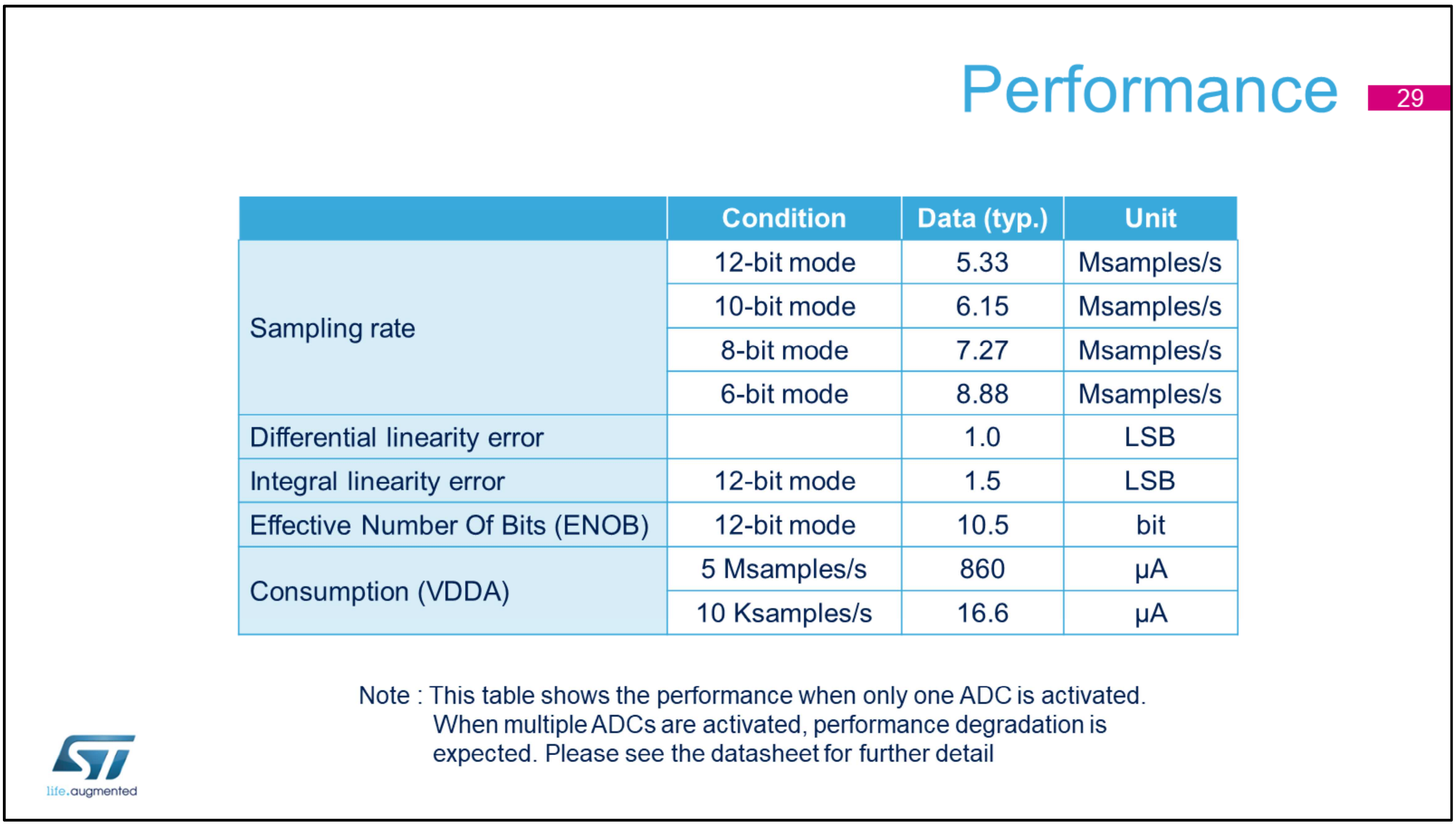


Low-power modes

Mode	Description
Run	Active
Sleep	Active ➢ Peripheral interrupts cause the device to exit Sleep mode
Low-power run	Active
Low-power sleep	Active ➢ Peripheral interrupts cause the device to exit Low-power sleep mode
Stop 0	Not available ➢ Peripheral registers content is kept
Stop 1	Not available ➢ Peripheral registers content is kept
Standby	Powered-down ➢ The peripheral must be reinitialized after exiting Standby mode
Shutdown	Powered-down ➢ The peripheral must be reinitialized after exiting Shutdown 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, Sleep, Low-power run and Low-power sleep modes.
In Stop 0 or Stop 1 mode, the ADCs are not available but the contents of their registers are kept.
In Standby or Shutdown 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.

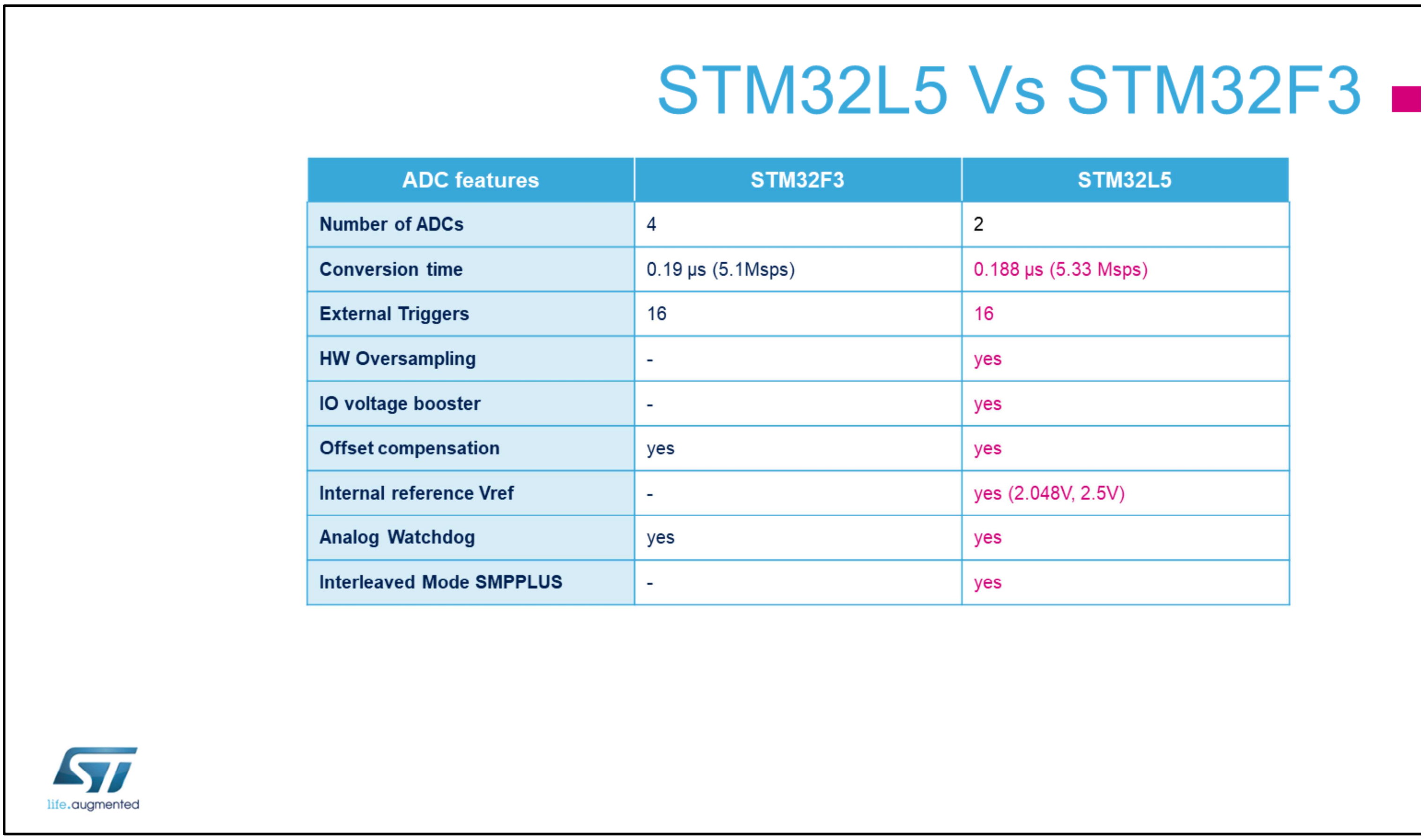


Performance

	Condition	Data (typ.)	Unit
Sampling rate	12-bit mode	5.33	Msamples/s
	10-bit mode	6.15	Msamples/s
	8-bit mode	7.27	Msamples/s
	6-bit mode	8.88	Msamples/s
Differential linearity error		1.0	LSB
Integral linearity error	12-bit mode	1.5	LSB
Effective Number Of Bits (ENOB)	12-bit mode	10.5	bit
Consumption (VDDA)	5 Msamples/s	860	µA
	10 Ksamples/s	16.6	µA

Note : This table shows the performance when only one ADC is activated. When multiple ADCs are activated, performance degradation is expected. Please see the datasheet for further detail

The following table shows performance parameters for the ADC.

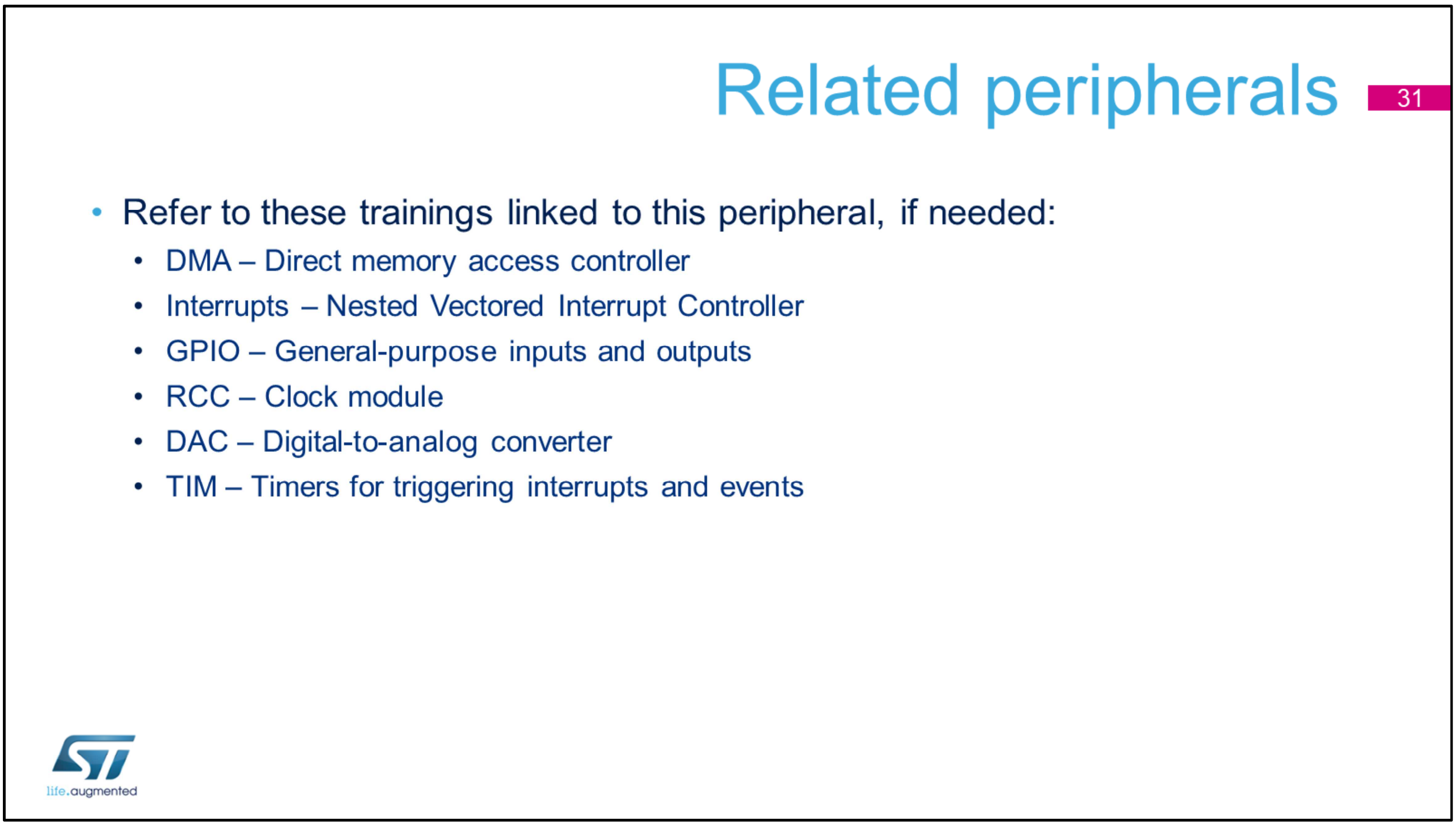


STM32L5 Vs STM32F3

ADC features	STM32F3	STM32L5
Number of ADCs	4	2
Conversion time	0.19 µs (5.1Msps)	0.188 µs (5.33 Msps)
External Triggers	16	16
HW Oversampling	-	yes
IO voltage booster	-	yes
Offset compensation	yes	yes
Internal reference Vref	-	yes (2.048V, 2.5V)
Analog Watchdog	yes	yes
Interleaved Mode SMPPLUS	-	yes

life.augmented

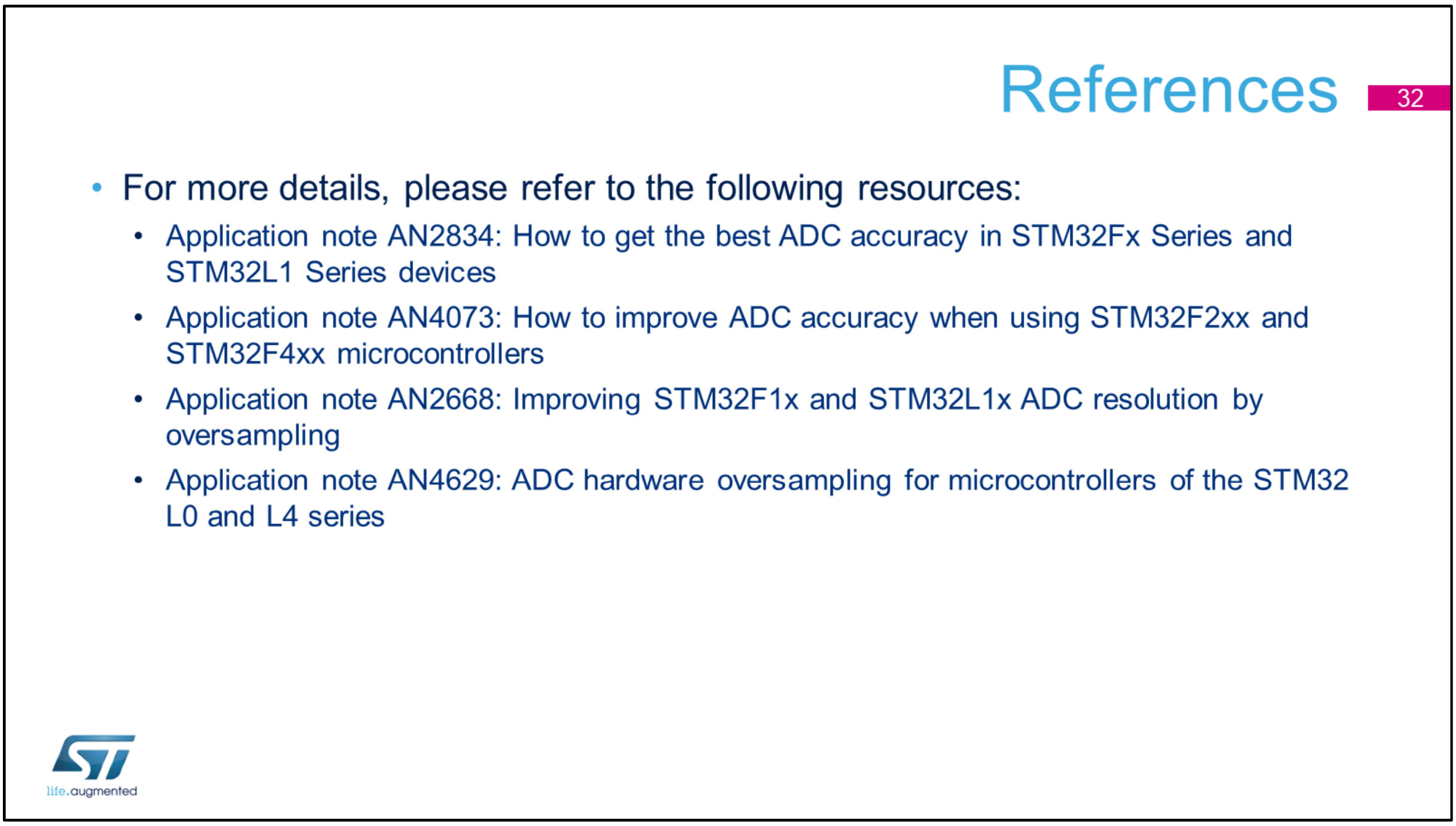
This table highlights the new features implemented in the STM32L5’s ADCs with regard to the STM32F3’s ADCs.



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

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

32

- 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.