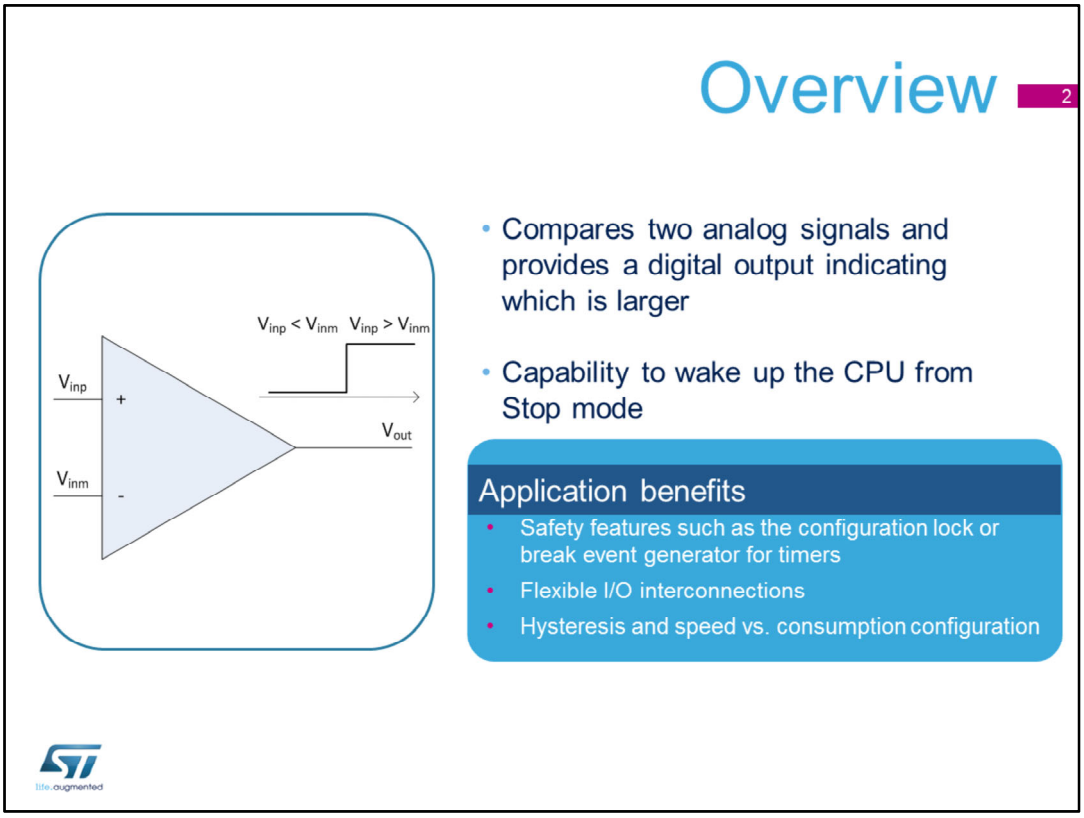
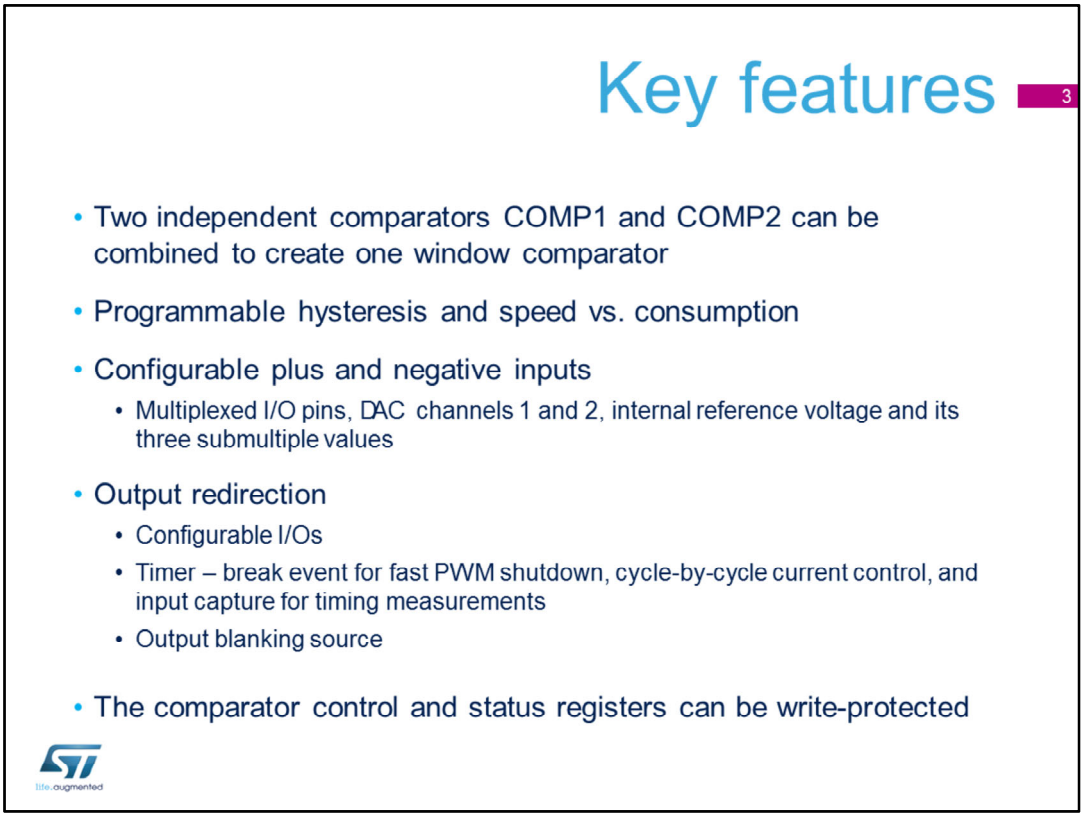
Hello, and welcome to this presentation of the STM32G0 analog comparators. It covers the main features of the ultra-low-power comparators and gives some application examples.



The two analog comparators inside STM32 microcontrollers provide a binary output which indicates if the analog voltage on the plus input is larger than the voltage on the negative input. It allows the MCU to react when the analog signal crosses a predefined threshold. The comparator continuously monitors voltage in contrast to an analog-to-digital converter which operates in sampled mode.
The comparator can be used to wake up devices from Sleep and Stop modes.
Applications can benefit from the flexible configuration of comparator properties which can be locked for safety reasons. Another safety feature of the comparator is its ability to generate a break signal for timers allowing to safely stop the generation of PWM driving signals.



The two integrated analog comparators can be combined into a single window comparator.
The analog properties of the comparator, including hysteresis or a trade-off between speed and power consumption, are configurable.
It offers flexible inter-connections of inputs and outputs allowing the selection of thresholds for several external and internal inputs such as DAC outputs or internal reference voltage outputs.
The comparator output can be connected to I/Os using the alternate function channels or internally redirected to a variety of timer inputs, enabling the break event for fast PWM shutdown. It is also possible to create cycle-by-cycle current control or input captures for timing measurements.
The COMPx control registers can be locked until the next microcontroller reset.

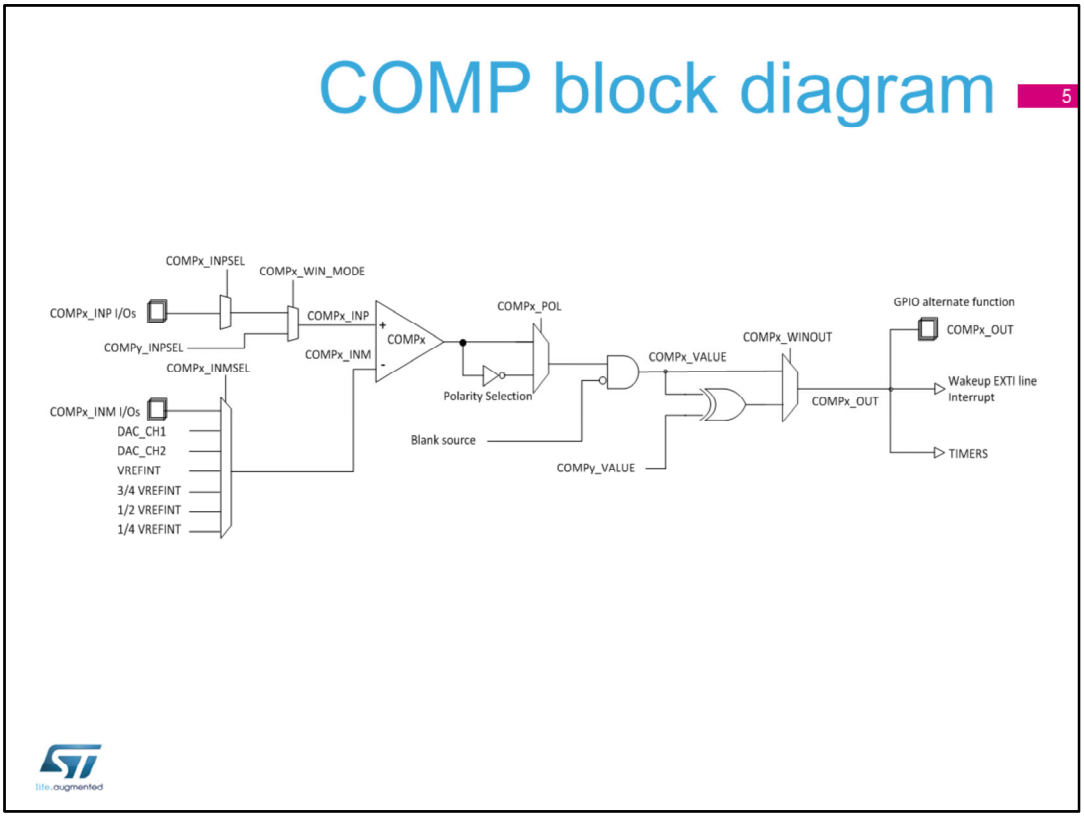
Main difference vs STM32F0

4

- The STM32G0 implementation is similar to the STM32F0, except for the following items:

Item	STM32F0x	STM32G0x
Window mode	Through CPU intervention	Full HW solution
Timer connection	To OCREF_CLR input	To ETR input
Comparators interchangeability	COMP1 and COMP2 not identical	Both comparator are identical
Blanking input	NA	Selectable blanking source

This table highlights the differences between the STM32F0 and STM32G0 comparators.



This slide shows the general block diagram of the comparator integrated in STM32G0 microcontrollers.
The multiplexors on the left select the voltage sources to be compared: GPIOs, DAC outputs, VREFINT with four divide ratios.
The output of the comparator can be inverted.
When the Blank source is active, the result of the voltage comparison is ignored and the COMPx_VALUE is negated.
The XOR gate and the last multiplexor on the right enable or disable the window mode.
Then the state of the comparator can be connected to:

- GPIOs
- EXTI module to generate a wakeup request or an event to the CPU
- Timer inputs.

COMP inputs (1/2)

COMPARATOR 1	COMPARATOR 2
COMP1_CSR[INMSEL]	COMP2_CSR[INMSEL]
0000 = 1/4 VREFINT	0000 = 1/4 VREFINT
0001 = 1/2 VREFINT	0001 = 1/2 VREFINT
0010 = 3/4 VREFINT	0010 = 3/4 VREFINT
0011 = VREFINT	0011 = VREFINT
0100 = DAC Channel1	0100 = DAC Channel1
0101 = DAC Channel2	0101 = DAC Channel2
0110 = PB1	0110 = PB3
0111 = PC4	0111 = PB7
1000 = PA0	1000 = PA2
COMP1_CSR[INPSEL]	COMP2_CSR[INPSEL]
00 = PC5	00 = PB4
01 = PB2	01 = PB6
10 = PA1	10 = PA3
11= Open	11= Open

Each comparator has a non-inverting input and an inverting input.
The INMSEL field in the COMP1_CSR and COMP2_CSR registers is used to select the inverting input.
The INPSEL field in the COMP1_CSR and COMP2_CSR registers is used to select the non-inverting input.

COMP inputs (2/2)

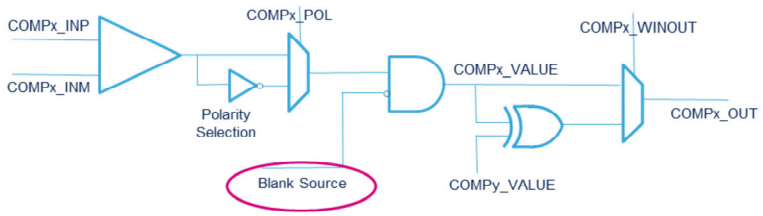
COMPARATOR 1	COMPARATOR 2
COMP1_CSR[BLANKSEL]	**COMP2_CSR[BLANKSEL]**
00000: No blanking	00000: No blanking
xxxx1: TIM1 OC4 enabled as blanking source	xxxx1: TIM1 OC4 enabled as blanking source
xxx1x: TIM1 OC5 enabled as blanking source	xxx1x: TIM1 OC5 enabled as blanking source
xx1xx: TIM2 OC3 enabled as blanking source	xx1xx: TIM2 OC3 enabled as blanking source
x1xxx: TIM3 OC3 enabled as blanking source	x1xxx: TIM3 OC3 enabled as blanking source
1xxxx: TIM15 OC2 enabled as blanking source	1xxxx: TIM15 OC2 enabled as blanking source
COMP1_CSR[WINOUT]	**COMP2_CSR[WINOUT]**
0: COMP1_OUT = COMP1_VALUE	0: COMP2_OUT = COMP1_VALUE
1: COMP1_OUT = COMP1_VALUE XOR COMP2_VALUE	1: COMP2_OUT = COMP1_VALUE XOR COMP2_VALUE

Each comparator's output can be masked during a blanking time defined by the timer output compare value selected in the BLANKSEL field.
The two comparators can be associated to form a window comparator through the WINOUT field.

Comparator output blanking function

8

- For comparators, possibility to have blanking to mask spurious over-current during turn-on, or zero-crossing events during turn-off (for Predictive Functional Control)
  - 5 sources from TIM1, TIM2, TIM3, TIM15

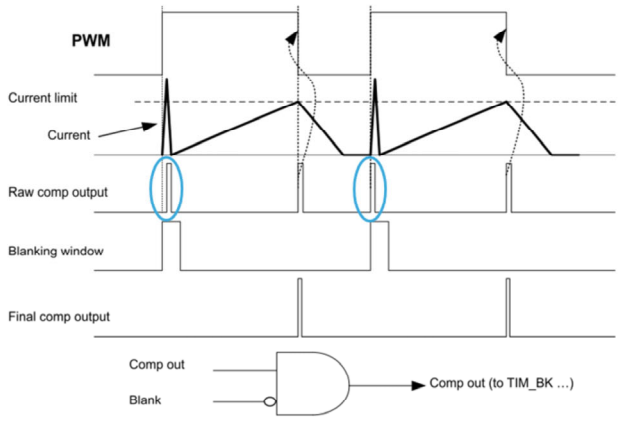


The blanking function aims to mask the output of the comparator during period of times indicated by a timer. This is typically used in the PFC technique (Predictive Functional Control).

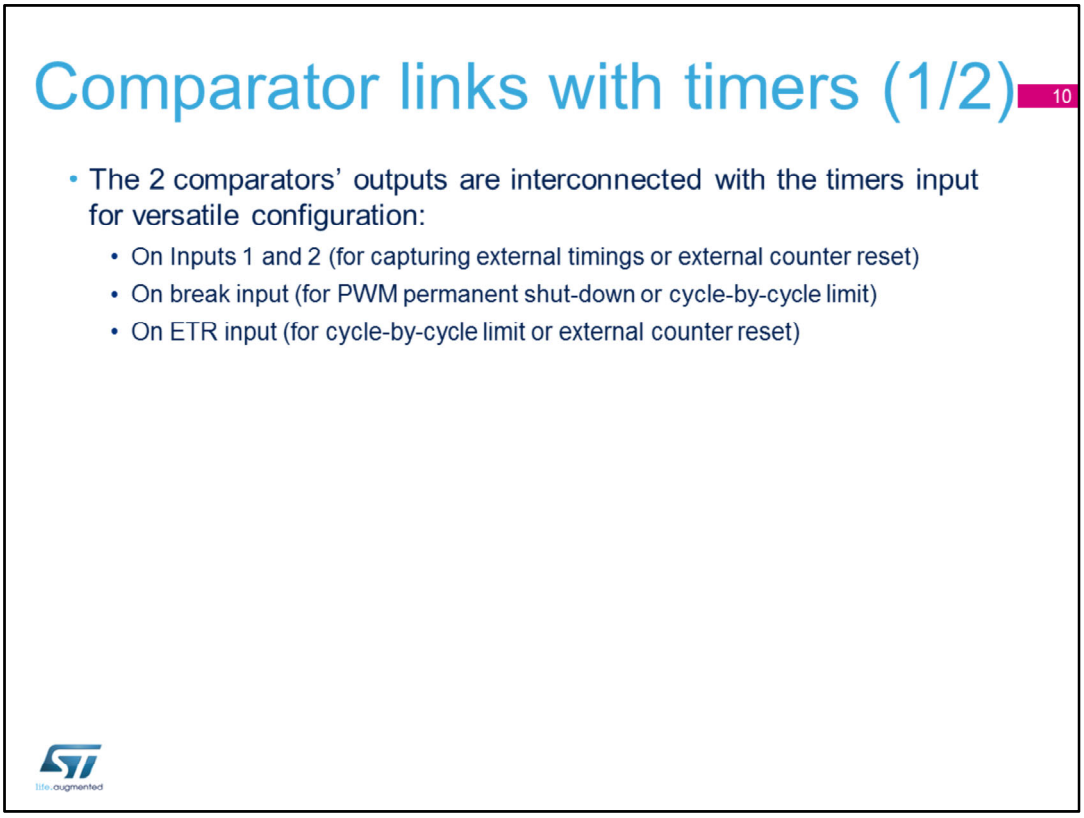
Comparator output blanking function

9

- Prevents current regulation tripping due to short-duration current spikes at the beginning of the PWM period
- Masks the COMP output redirected to timer break input



The comparator can be used in the cycle-by-cycle regulation loop for monitoring the peak value of the current flowing into the load. The purpose of the blanking function is to prevent incorrect current regulation tripping due to short duration current spikes at the beginning of the PWM period. Short current spikes caused by activating the power switches can produce false pulses on the comparator output – marked by the blue color on the diagram. These pulses need to be masked by a blanking window to avoid false fault detection. The blanking window waveform can be generated by one of the timer output channels.



The comparators have internal connections with the timer units.
The output can be internally redirected to a wide range of timer inputs for the following purposes:

- Emergency shut-down of PWM signals, using BKIN and BKIN2 inputs
- Cycle-by-cycle current control, using Electronic Timing Relay (ETR) inputs
- Input capture for timing measures.

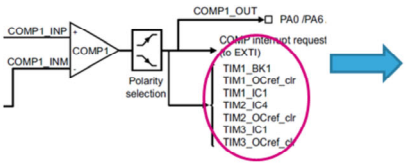
Comparator links with timers (2/2)

- Interconnection options are now directly selectable in the timers

	IC1	IC2	ETR	BKIN	BKIN2
TIM1	COMP1	COMP2	COMP1 / COMP2	COMP1 / COMP2	COMP1 / COMP2
TIM2	COMP1	COMP2	COMP1 / COMP2		
TIM3	COMP1	COMP2	COMP1 / COMP2		
TIM15				COMP1 / COMP2	
TIM16				COMP1 / COMP2	
TIM17				COMP1 / COMP2	

- Main change vs. STM32F0: the OCREF_CLR input is now replaced with a connection to the ETR input

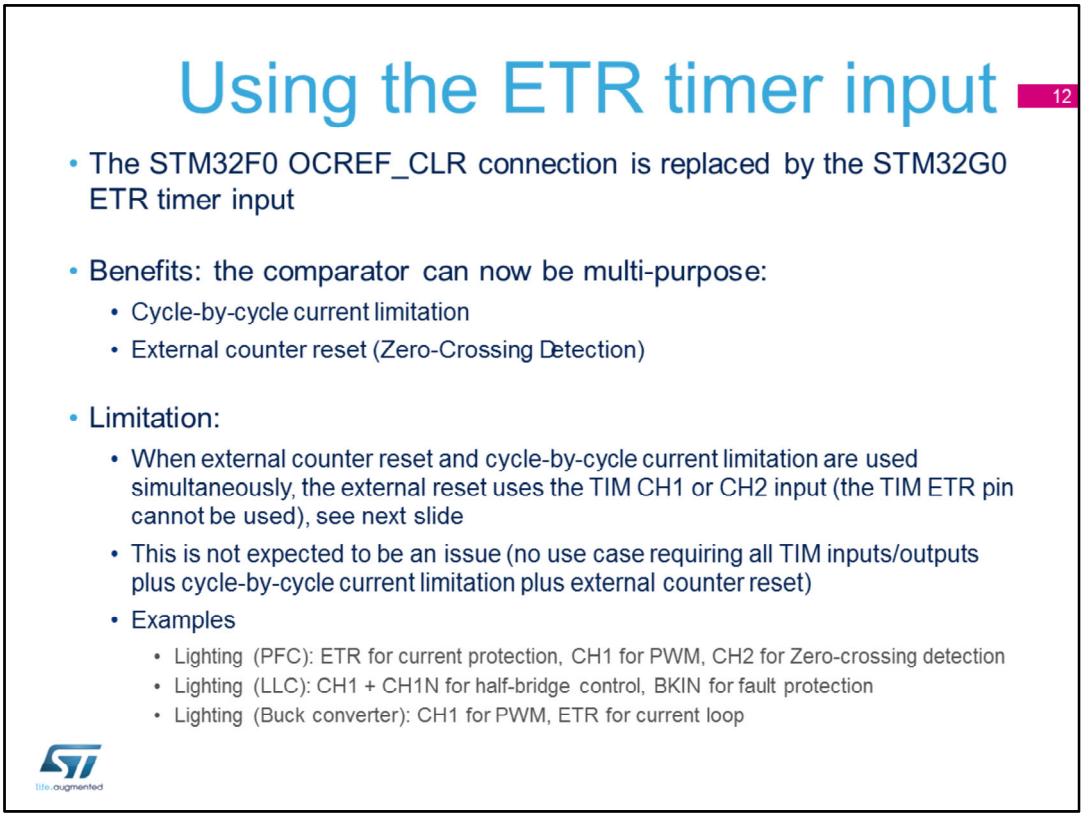
STM32F0



STM32G0

Replaced by a direct connection to the timer (input capture, ETR, BKIN, BKIN2

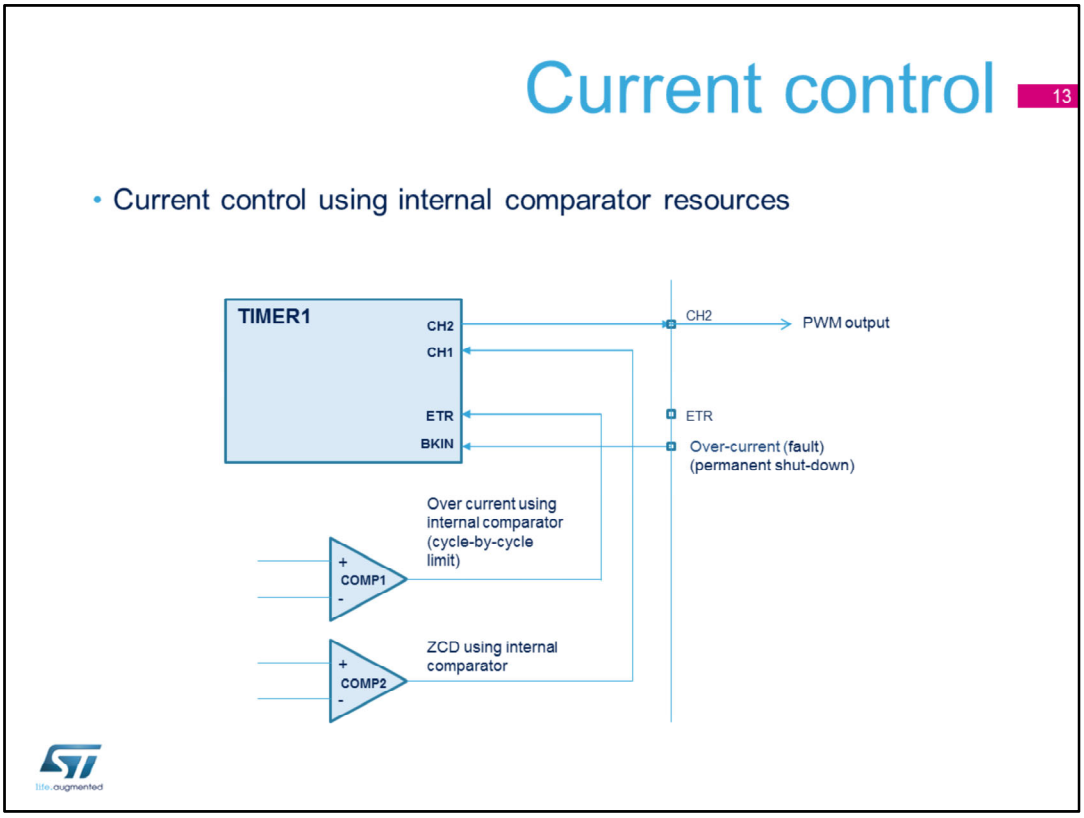
The upper table is the connection matrix between the comparator and the timer units.
The OCREF_CLR timer input used in STM32F0 is replaced with the ETR input in the STM32G0.



The connection between the comparators and the timer is generally used for two purposes:

- Cycle-by-cycle current limitation based on the blanking mechanism
- External counter reset when the voltage drops below a threshold: zero-crossing detection.

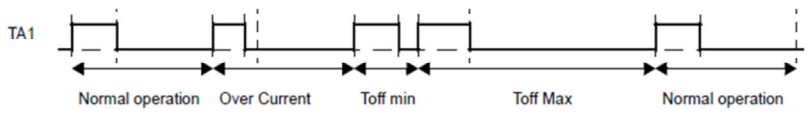
When both are needed simultaneously, the current limitation is based on the ETR timer input and a counter reset is signaled through a timer channel input.



The figure represents an example of direct connection between Timer and COMP units.
Over-current limitation uses the ETR input and external reset uses the CH1 input.

Timer capability summary

Device	Toffmax	Toffmin	Tonmax	Tonmin*	OverCurrent	Zero-cross
STM32F05x	Yes	Yes, with external loopback	Yes	Yes, with external loopback	Comparator or BKIN dig. Input	OCREF_CLR or TI1/2 inputs
STM32G0	Yes	**Yes, with internal link**	Yes	**Yes (comparator blanking)**	Comparator, BKIN and BKIN2 dig. Inputs	ETR or TI1/2 inputs

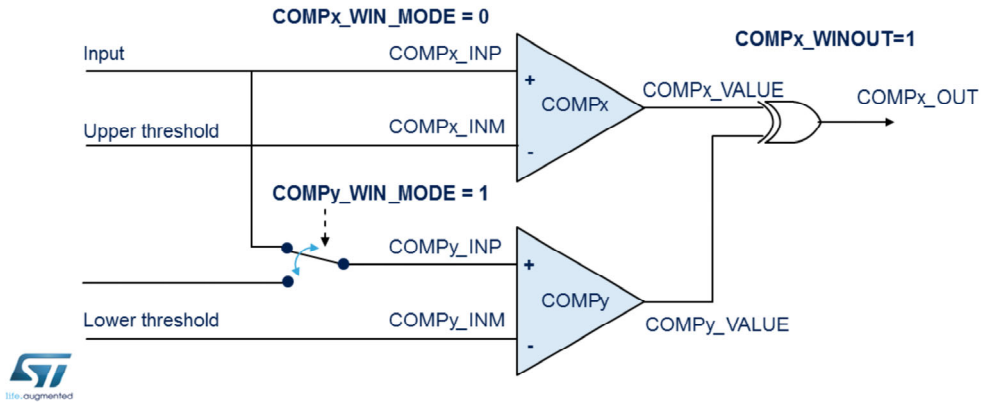


When combined with a PWM output from a timer, the cycle-by-cycle current control loop is simplified in the STM32G0 with regard to STM32F0 microcontrollers.
No external loopback is required.
Tonmin and Toffmin are controlled by direct signals connecting the comparators to the timers.

True window comparator

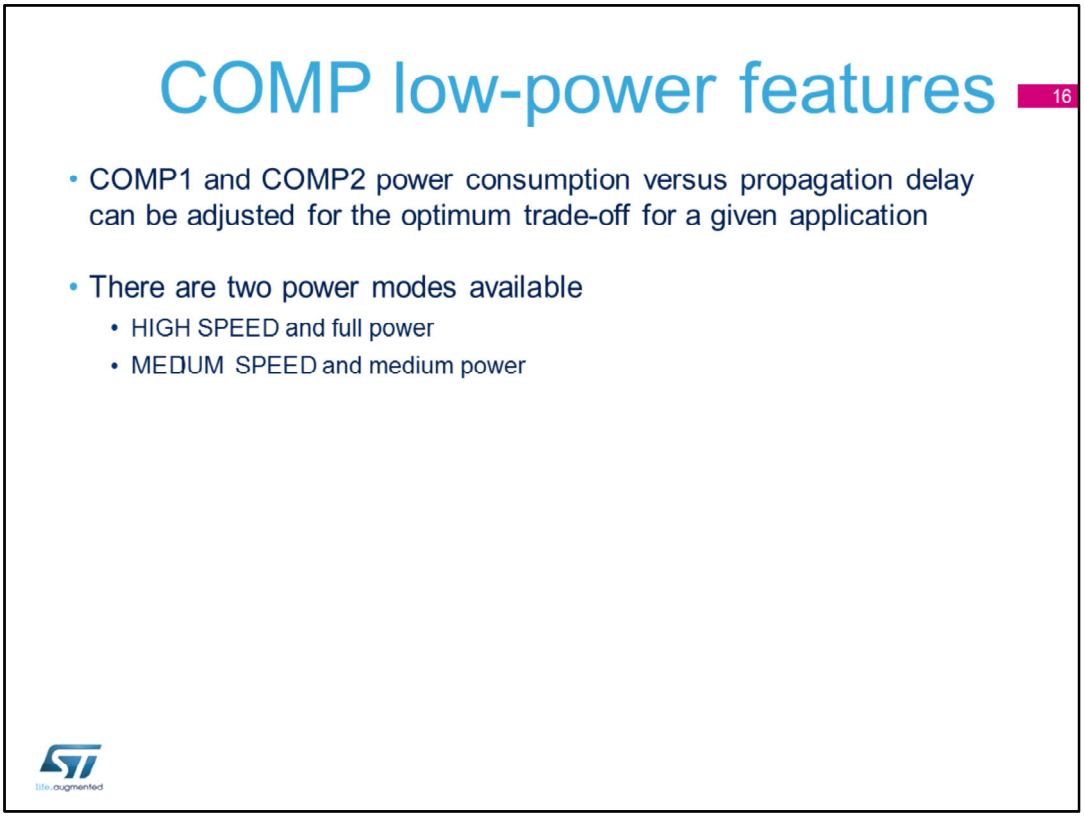
15

Area	Lower threshold	Upper threshold	In window?	Output
1	Below	Below	Outside	0
2	Above	Below	Inside	1
3	Above	Above	Outside	0
	Below	Above	NA	1

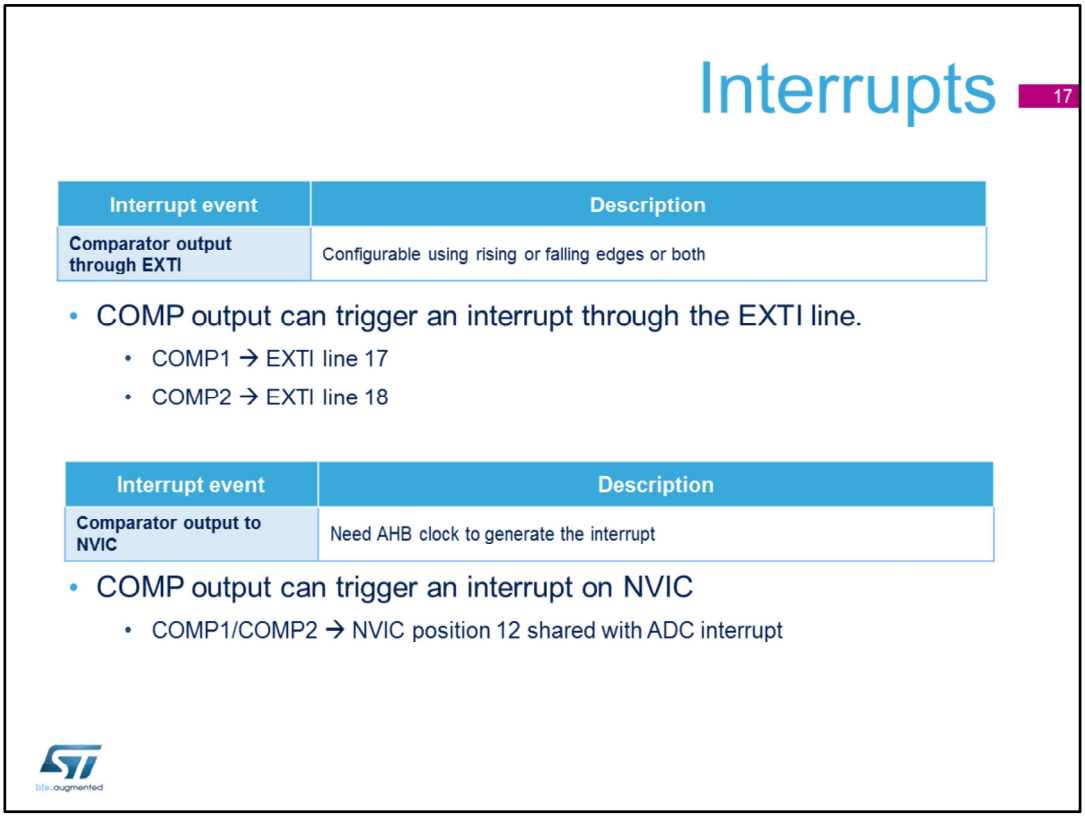


The purpose of the window comparator is to trigger an interrupt if the analog voltage exceeds the predefined lower and upper voltage thresholds applied to the inverting inputs of each comparator. This event can generate an interrupt through the EXTI line. Two non-inverting inputs can be connected internally by enabling the WINMODE bit and therefore save one IO for another purpose.

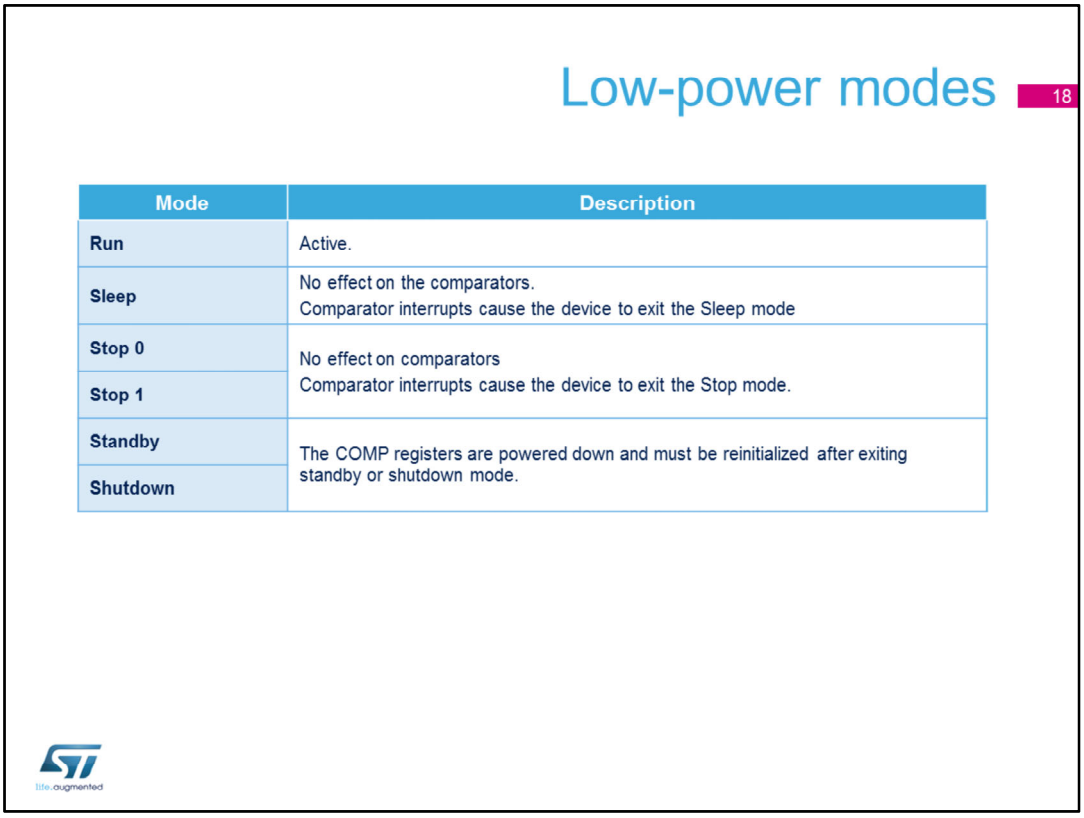
The comparator performs continuous voltage monitoring while the ADC watchdogs periodically sample the input voltage.



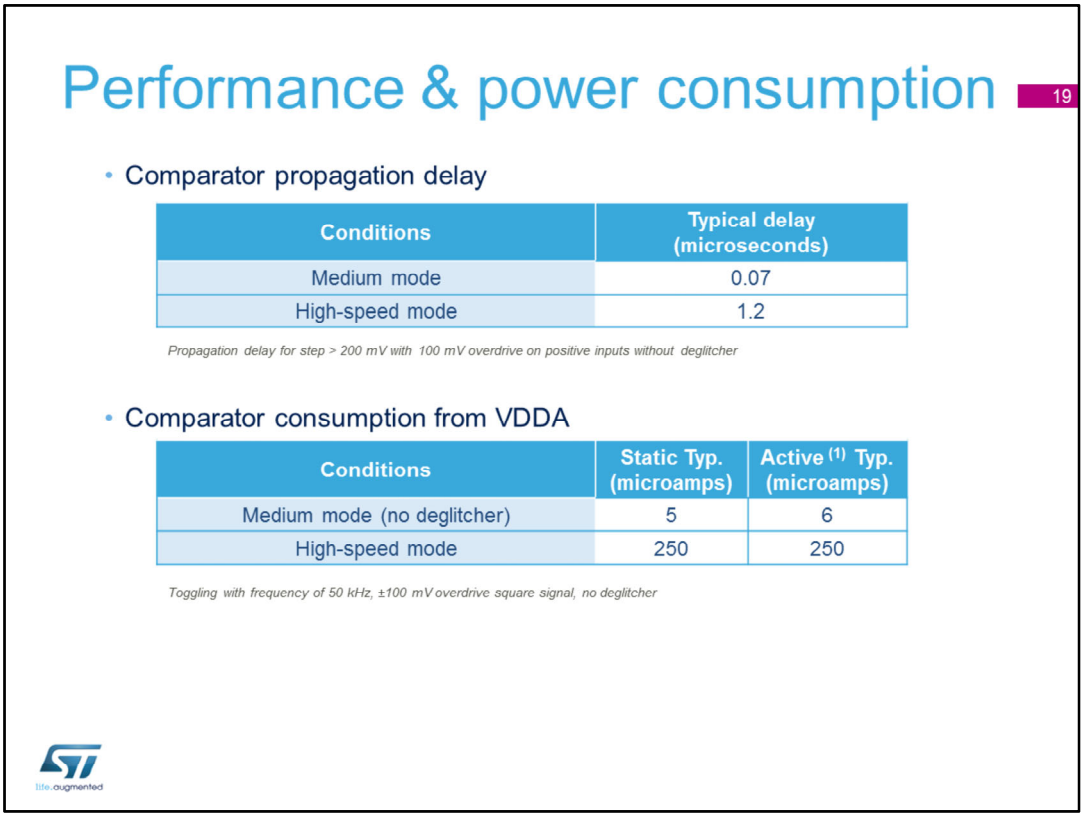
The comparators' power consumption can be adjusted to have the optimum trade-off between the speed and energy efficiency for a given application.
There are three modes available: high-speed, medium-speed and ultra-low-power. High-speed mode is preferred for power conversion applications - for example, a motor control design.
The comparator can stay active even if the rest of the system is suspended and the clock is switched off.



The comparator can trigger an interrupt on the rising, falling or both edges of the comparator output through the EXTI line. This is required to exit the Stop modes. The output can also be connected to the CPU's nested vectored interrupt controller (NVIC).



The on-chip comparator remains active in the following modes: Run, Sleep, and Stop modes.
In Standby and Shutdown modes, it is powered-down and must be reinitialized for use if returning to one of the higher powered modes.



The on-chip comparator configuration capability allows the user to select the best performance point for the targeted application. It replaces the external stand-alone comparator, thereby reducing the bill of materials.