

STM32G4- HRTIM

High-Resolution Timer
Revision 1.0



Hello, and welcome to this presentation on the STM32G4 High-resolution timer. It will cover the main features for managing the complex waveforms required for digital power conversion applications.

Why high-resolution?

- The high-resolution timer (HRTIM) addresses digital power applications with high PWM switching frequencies
 - Higher duty cycle resolution (Buck / Boost converters,...)
 - Better accuracy on output voltage
 - No limit cycling issues
 - Finer frequency and phase adjustments (LLC and phase-shift converters,...)
 - Better regulation and load transient management

Timer	Timer resolution	TIMx	HRTIMER
	Equivalent Clock frequency	5.9ns 170MHz	184ps 5.44GHz
Resolution ($F_{PWM} = 300kHz$)		566 levels (9.1 bit)	18133 levels (14.1 bit)
Frequency adjustment step ($F_{PWM} = 300kHz$)		532Hz	16.5 Hz



The high-resolution timer essentially addresses digital power applications, such as switch mode power supplies, that require high PWM switching frequencies. It allows the drastic improvement of the duty-cycle resolution, typically for Buck or Boost converters. This results in a better output voltage resolution and regulation, and avoids the limit cycling issues which can happen with a regular resolution timer, when the duty cycle accuracy is lower than the ADC accuracy. This also allows a finer frequency with phasing adjustments, which is a key parameter for resonant converters using LLC or full-bridge phase-shifted topologies. The table presents the differences between standard and high-resolution timer, for a product running at 170MHz, for both the duty-cycle and the frequency resolution.

High-resolution, practically

3

- The HRTIM generates high-resolution signals thanks to a Delay Locked Loop (DLL), dividing by 32 the input clock period.
- The DLL requires a calibration, prior to the HRTIMER start-up.
- The DLL operates in the 100 to 170 MHz frequency range only.
 - The resolution is scaled with the input frequency: at 100 MHz, it is 312 ps
- The high-resolution is available for any kind of waveform adjustment (period, duty-cycle or PWM-to-PWM phase adjustment), but not for input capture, performed with the regular input clock.

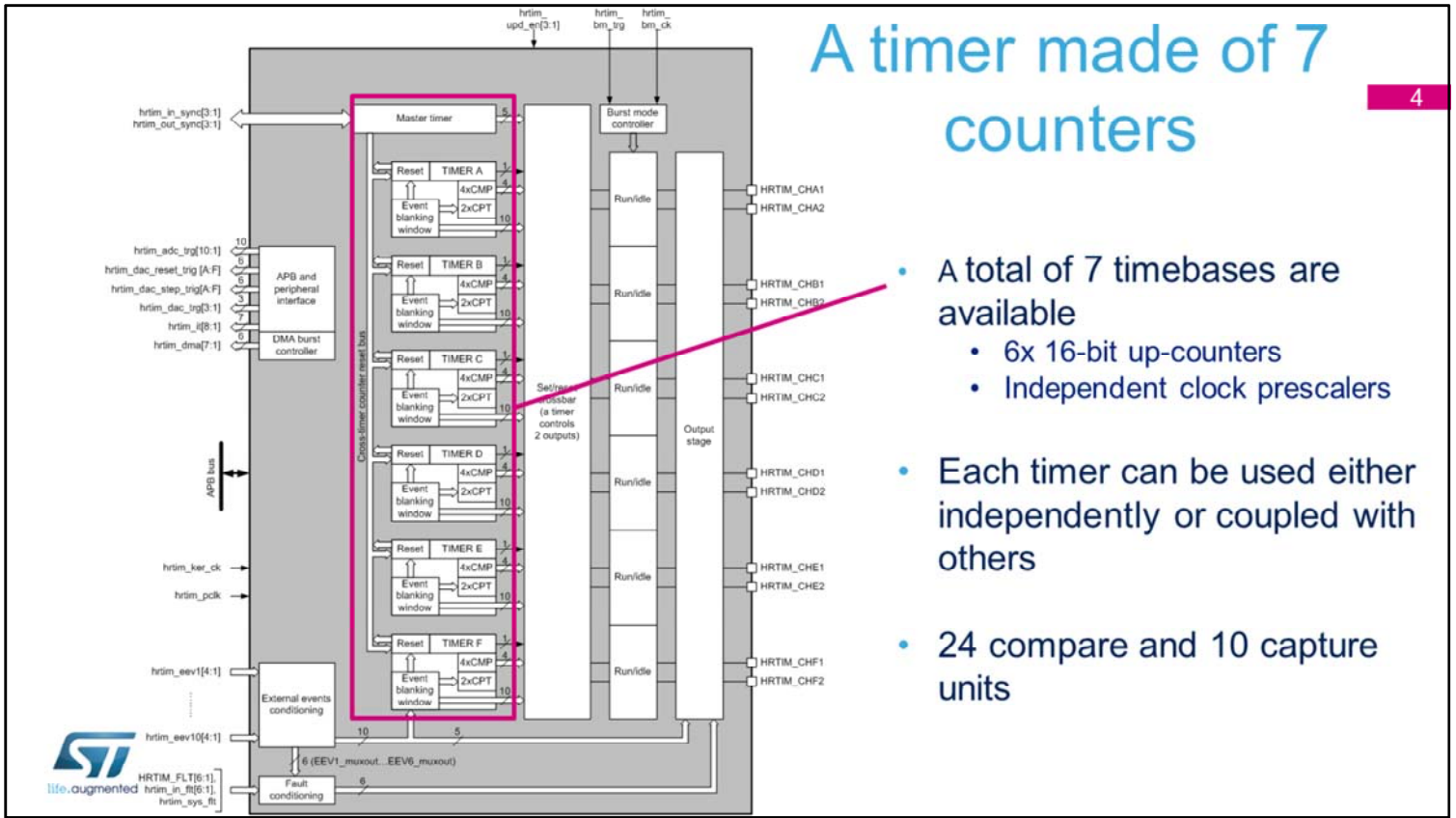


The high-resolution is implemented using a Delay-locked loop (DLL), which allows the division of the timer input clock period into 32 fractional steps.

This DLL requires a calibration prior to the HRTIMER start-up. It must be noticed that the DLL only operates between 100 and 170 MHz, yielding a resolution ranging from 312 ps down to 184 ps, as the resolution is scaled with the input frequency.

The high-resolution is available for any kind of waveform adjustment (period, duty-cycle, or PWM to PWM phase adjustment), but not for input capture, performed with the regular input clock (from 100 to 170 MHz).

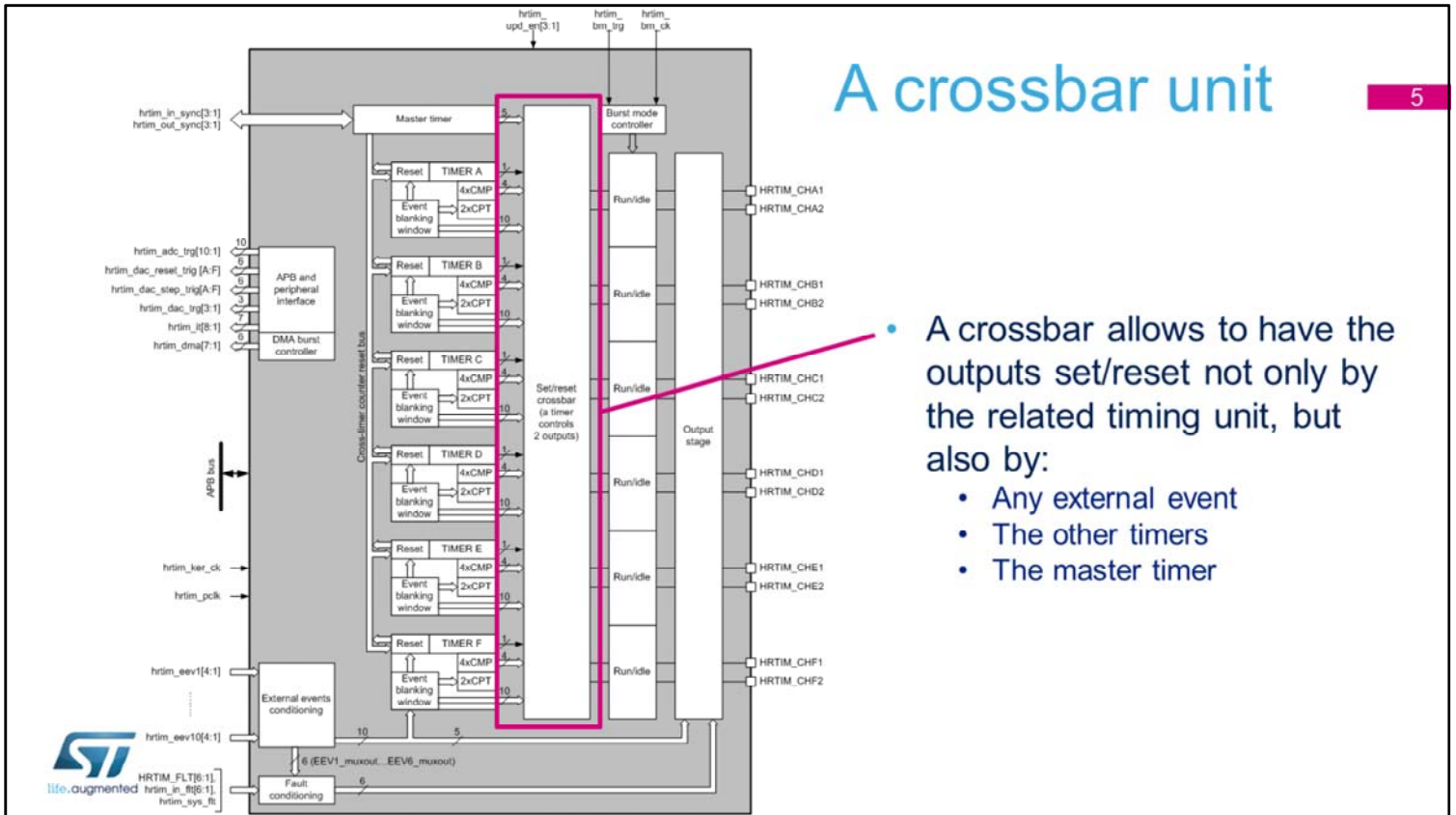
A timer made of 7 counters



- A total of 7 timebases are available
 - 6x 16-bit up-counters
 - Independent clock prescalers
- Each timer can be used either independently or coupled with others
- 24 compare and 10 capture units

Lets' now have an overview of the HRTIMER architecture. Although we talk about a single HR Timer, it is actually made up of seven 16-bit counters that can either be used independently or synchronized, each of them having an independent clock prescaler. It is equipped with a total of 28 compare and 12 capture units and can control up to 12 outputs.

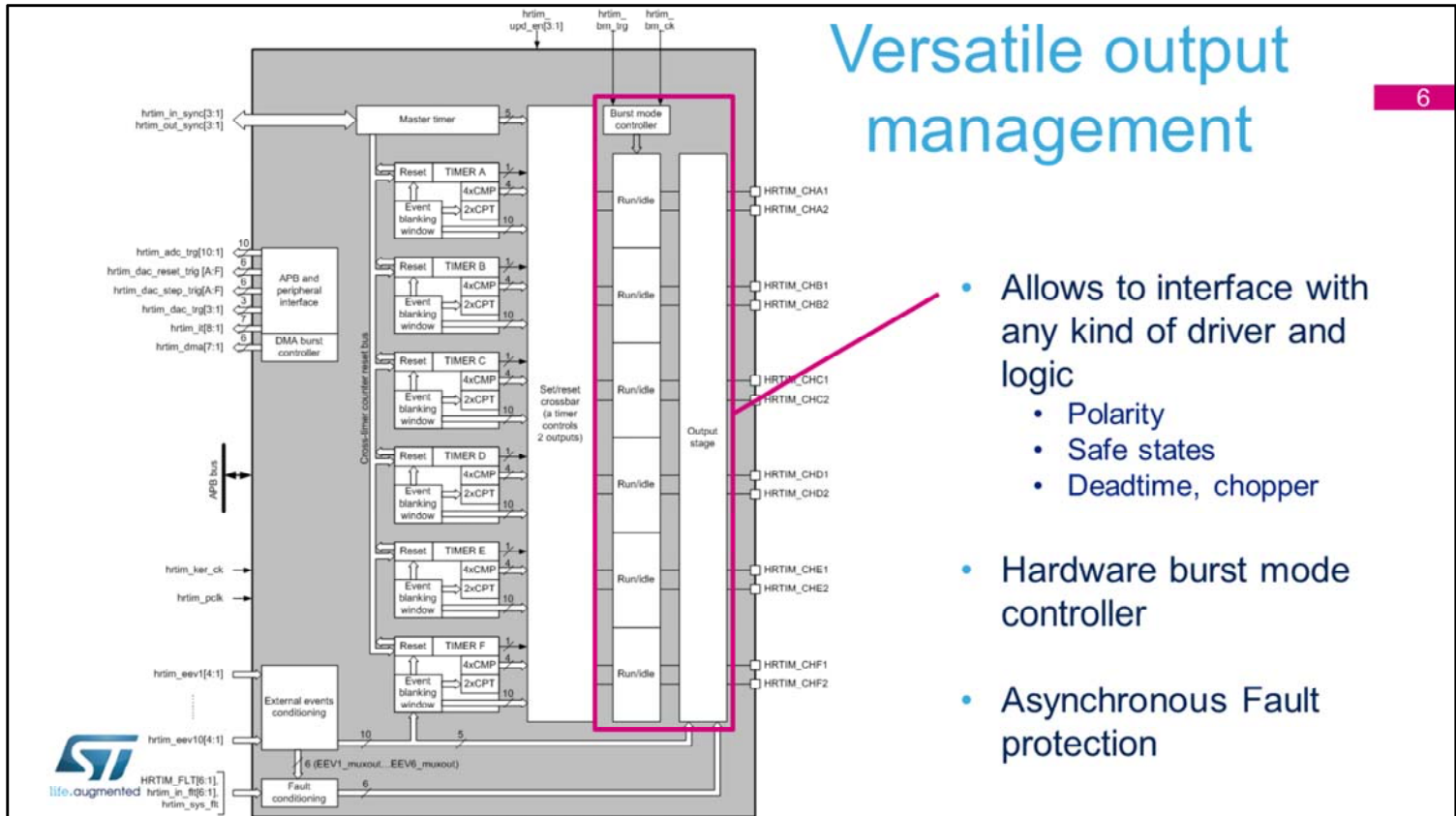
A crossbar unit



- A crossbar allows to have the outputs set/reset not only by the related timing unit, but also by:
 - Any external event
 - The other timers
 - The master timer

A large central crossbar unit allows to have the output pairs controlled not only by the related timing unit (for instance Timer A for output CHA1), but also by any external event, the other timing units and/or the master timer. A total of 32 set and reset events are available per output.

Versatile output management



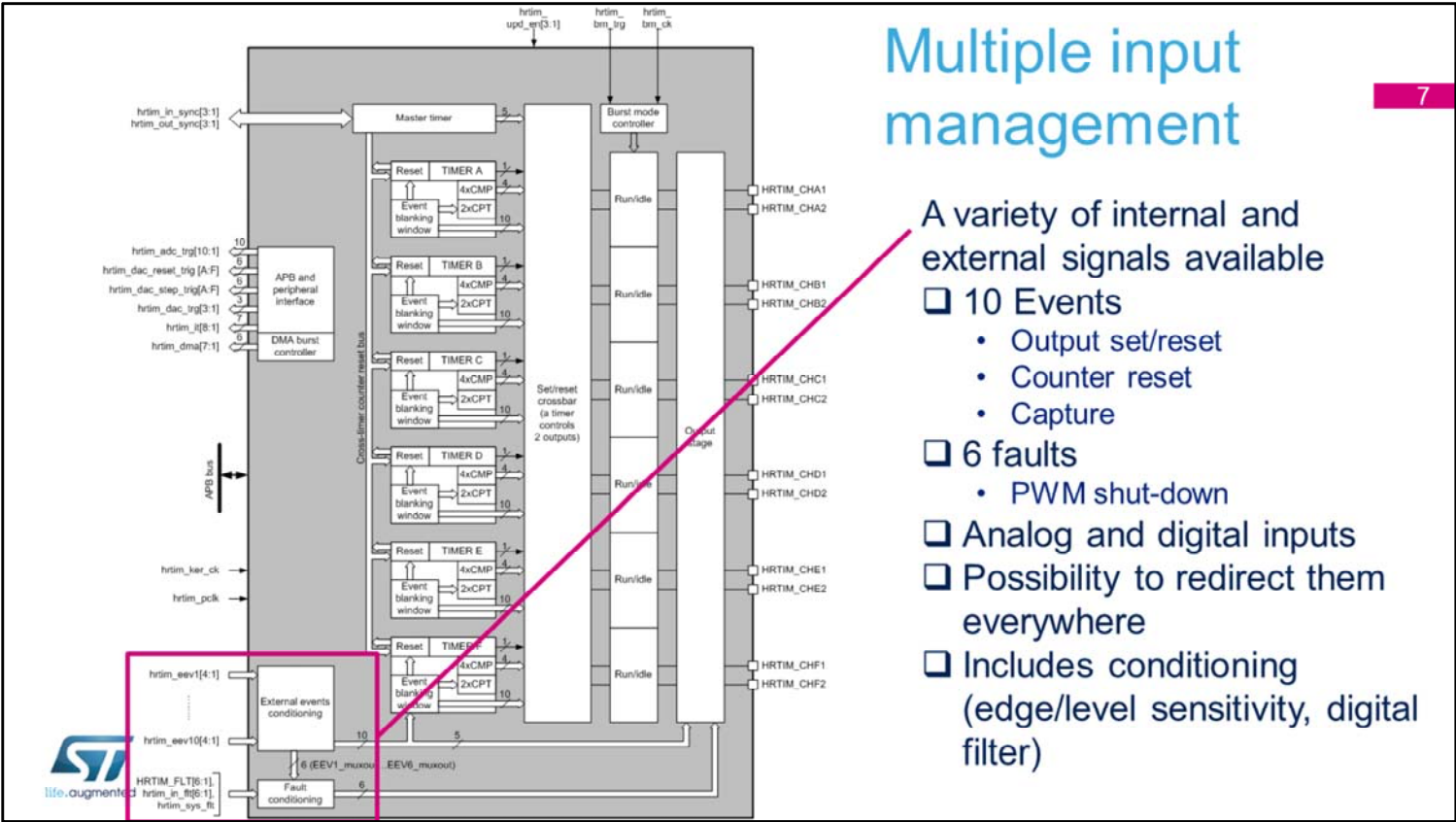
- Allows to interface with any kind of driver and logic
 - Polarity
 - Safe states
 - Deadtime, chopper
- Hardware burst mode controller
- Asynchronous Fault protection

The crossbar is followed by a versatile output management feature able to interface with any kind of gate driver; program the output polarity, the PWM idle and safe state levels; insert a deadtime; and/or add a chopper modulation.

A hardware burst mode controller facilitates the light load management with a built-in pulse skipping mechanism.

It also features an ultra-fast and asynchronous fault protection feature, independent from the system clock.

Multiple input management



A variety of internal and external signals available

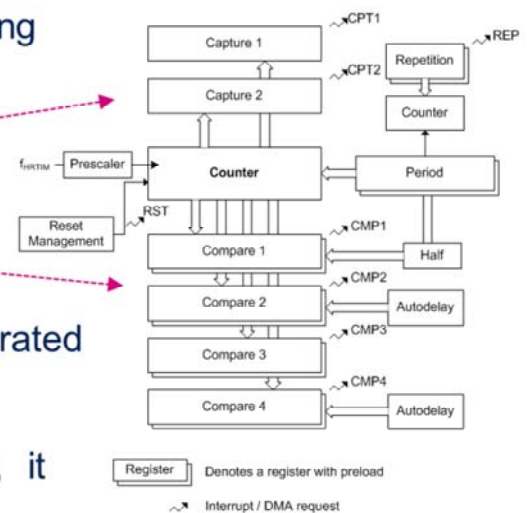
- 10 Events
 - Output set/reset
 - Counter reset
 - Capture
- 6 faults
 - PWM shut-down
- Analog and digital inputs
- Possibility to redirect them everywhere
- Includes conditioning (edge/level sensitivity, digital filter)

The HR Timer is able to handle 10 external events for dynamically modifying a waveform in addition to 6 fault signals for protecting power stages. The 10 events can be selected among external ones (on inputs) and internal ones (typically from comparators). The events allow performance of output set/reset, to reset the PWM counters or capture external timings. The fault signals are here to protect the power stages and shut down the PWM outputs. The HR Timer is also linked with other timers as well as the DMA, ADC and DAC peripherals via internal interconnections.

Timer A..F timing units

8

- The High Resolution Timer embeds 6 identical timing units made of:
 - 16-bit up-counter with programmable period
 - 2x 16-bit capture registers
 - 4x 16-bit compare registers
- A compare event is generated only if the counter matches the compare: several events can be generated in single period for multiple set/reset
- Each unit embeds all control features for 2 outputs, it can eventually operate as a standalone timer



Let's take a closer look at one of the High Resolution Timer's five identical timing units.

It is made of a 16-bit up-counter with a programmable overflow value to adjust the counting period.

Four 16-bit compare units and two 16-bit capture units are linked to this counter, each with interrupt and DMA capability.

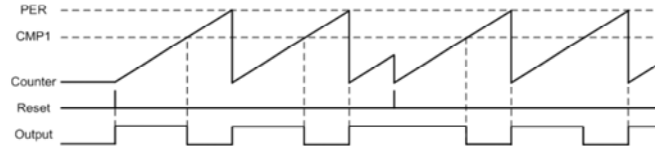
A compare event is generated when the counter exactly matches the compare value. Several compare events can thus be generated during a single period to have the outputs set and reset multiple times per PWM cycle.

The unit embeds all control features for a pair of outputs, it can eventually operate as an independent timer. Yet these outputs can also be controlled by other timers.

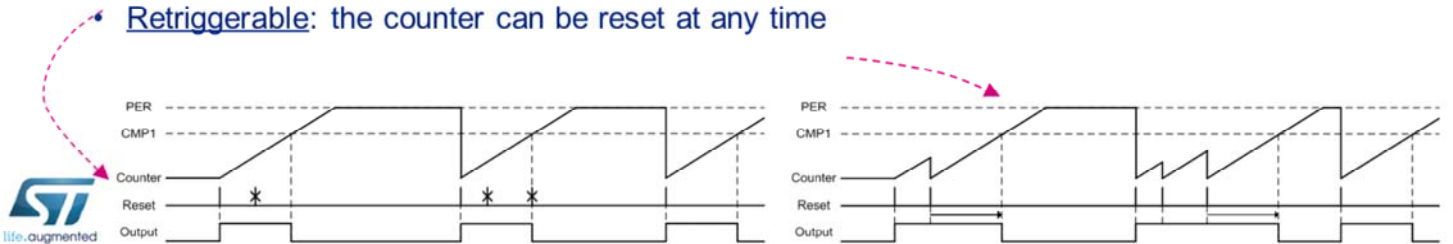
3 timer operating modes

9

- Continuous (free-running) with re-start on period or reset event
 - Standard PWM mode: fixed frequency, set with period register (roll-over)
 - Variable frequency PWM using reset events



- Single-shot mode
 - Non-retriggerable: reset events are discarded before PER event
 - Retriggerable: the counter can be reset at any time



The up-counters can operate in three modes.

In Continuous mode (also called Free-running mode), the counter rolls over to zero when it exceeds the value programmed in the period register, or re-starts counting if it is reset by an external event. This is the most common mode, used for fixed frequency converters or for externally synchronized variable-frequency converters.

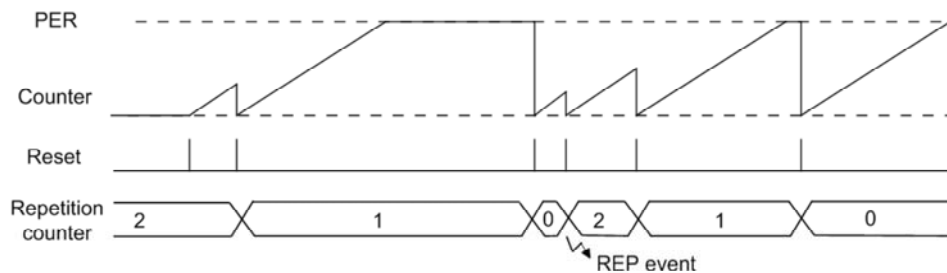
In Single-shot mode, the counter is started by a reset event and it stops when it reaches the period value set in the HRTIM_PERxR register. Two options are available here.

In Non-retriggerable mode, the counter reset events are discarded if they occur before the end of the counting phase. Typically, this is to make sure an externally triggered pulse will always have the same pulse width.

In Retriggerable mode, the counter can be reset at any time. In this case, an externally triggered pulse may have an extended pulse width if it receives multiple triggers closely enough.

Repetition counter 10

- Allows to off-load the CPU by decoupling the switching frequency and the interrupt frequency
 - Similar concept as with standard STM32 timers
- Can also be used when the timer is reset and in Single-shot mode, as shown below

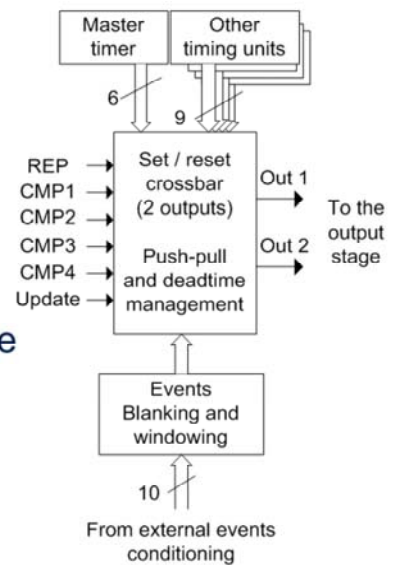


Each timing unit includes a repetition counter, similar to the one available in standard STM32 timers.

It allows off-loading of the CPU by decoupling the switching frequency and the interrupt frequency. The counter roll-over interrupt can be issued every single, 2nd, 3rd and up to 256th PWM period.

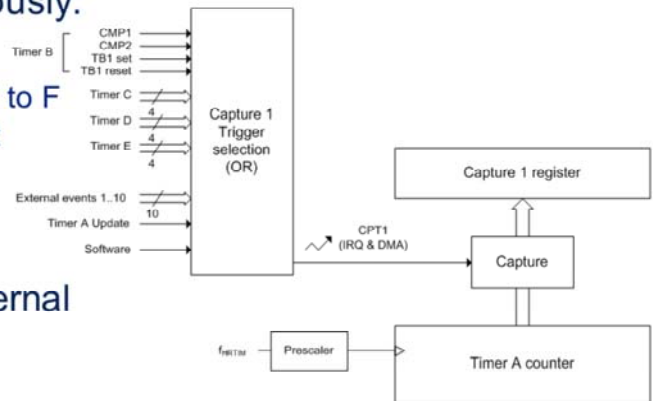
This feature is available both for Continuous and Single-shot modes. In this case, the repetition counter is decreased on each reset event.

- Set/reset events correspond to an output transition to active/inactive state (switch ON or OFF), regardless of polarity programming
- Each timing unit contains crossbar programming registers for the 2 outputs
 - 2x 32-bit registers per output for set and reset
- Up to 32 events can be selected simultaneously for multiple output set/reset events within a period
- Programming registers are preloaded and can be updated synchronously with other registers
 - Set/reset events can be changed on-the-fly during timer operation



The crossbar combines the set and reset events to build the output waveforms, regardless of polarity programming. The events may come from the timing unit itself, from other timing units, from the master timer or from external events. The timing unit contains 2 crossbar programming registers per output, one defining the set events, the other for reset events. Up to 32 events can be selected simultaneously: this allows building periodical waveforms with multiple set/reset events per period. The programming registers are shadowed with preload registers, so that the waveforms can be modified on-the-fly without any risk of having abnormal transient waveforms.

- Each timing unit contains 2 capture registers, each having a programmable trigger selection
- Up to 28 events can be selected simultaneously:
 - the external events, EXTEVNT1..10 (10 events)
 - all other timing units in the HRTIM (e.g. Timers B to F for Timer A): Compare 1, 2 and output 1 set/reset events (16 events)
 - Timing unit update and software capture events
- Captured values are used to determine external timings and for Auto-delayed mode



In parallel with the compare units, each timer also embeds 2 capture units. The capture triggers the transfer from the current counter value into the capture register. This is useful for determining external timings and for the auto-delayed mode discussed on next slide. Typically it offers the possibility to measure the effective duty cycle in a cycle-by-cycle current controlled converter, where the PWM pulse is terminated when the current comparator trips.

The capture triggers can be selected from among 28 sources: external events, adjacent timing units, adjacent output waveforms, as well as update and software events. It is possible to simultaneously enable several triggers sources.

- Convenient whenever a 50% duty cycle PWM must be generated
 - Typically in resonant converters, where the operating frequency is continuously adjusted by control loops and the duty cycle must be maintained at 50%
- A write access to the Period register causes an automatic update of the Compare 1 value
 - Typically, a Compare 1 match event will be used to set/reset the PWM output(s)
 - Not user accessible (the content of CMP1 register is not affected)
- Makes sense mostly for continuous mode fixed frequency operation, but can be used in any other mode as long as the PER register needs to be accessed



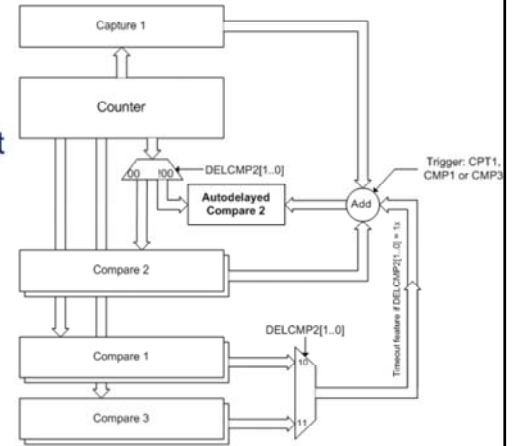
Half mode is a feature that allows reduction of computational burden when a variable frequency 50% duty cycle PWM waveform must be generated. This is typically the case for resonant converters, where the operating frequency is continuously adjusted by the control loop while the duty cycle must be maintained at 50%.

The Half-mode operating principle is to have a write access to the period register triggering a computation and an update of the Compare 1 register to half the value of the period.

Let's consider a PWM signal set on the period event and reset on a Compare 1 match. As soon as a new period value is written into the period register, the Compare 1 value is also updated to the period/2 value, so that the output set event will still happen right in the middle of the PWM waveform.

This is particularly useful for continuous mode fixed frequency operation, but it can be used in any other mode as long as the period register needs to be accessed.

- Allows to have an output changing upon a capture event with a programmable delay or after a timeout
- Computes on-the-fly compare value based on external event arrival time (capture register)
 - The compare register holds the delay vs. the expected event
 - Compare event is inhibited until
 - the capture occurs
 - if Timeout mode is enabled, the capture is forced by Compare 1 or Compare 3 if no capture occurs
- A new capture is possible
 - Once auto-delayed event has happened
 - After counter reset or roll-over



Auto-delayed mode offers the possibility to have set/reset timings referenced to external events, and not only related to the internal timebase. Still a timeout mechanism can ensure a default waveform when the external event is not present or disappears. This HRTimer computes a new compare value on-the-fly, based on external event arrival time, captured in a register, as soon as the event arrives.

The delay vs. the expected event is programmed in the compare register, but the compare event is inhibited until the capture is triggered.

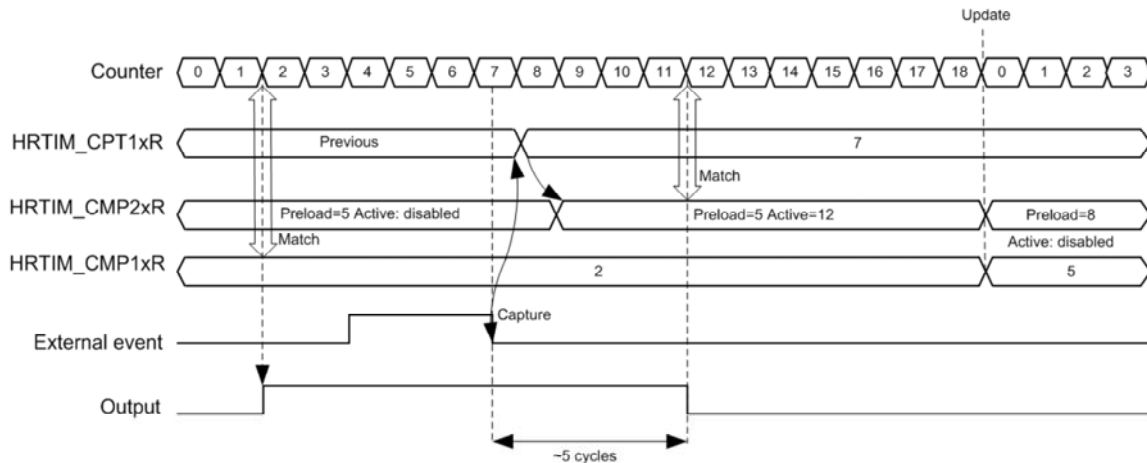
When the Timeout safety mechanism is enabled, the capture is forced when the counter equals the timeout threshold defined in the Compare 1 or the Compare 3 value.

A new capture is possible once the auto-delayed event has happened, or when a new period starts.

Auto-delayed application example

15

- In this example, the output waveform is defined as follows:
 - Set on Compare 1 match (regular Compare mode)
 - Reset on compare 2 (Auto-delayed mode), 4 cycles after a falling edge on external event



This slide gives a practical use case of Auto-delayed mode. The requirement is to build a waveform as follows:

- The output must be set on Compare 1 match, with a fixed timing vs the counter start value, here 2 clock cycles.
- The output must be reset 5 cycles after the external event falling edge.

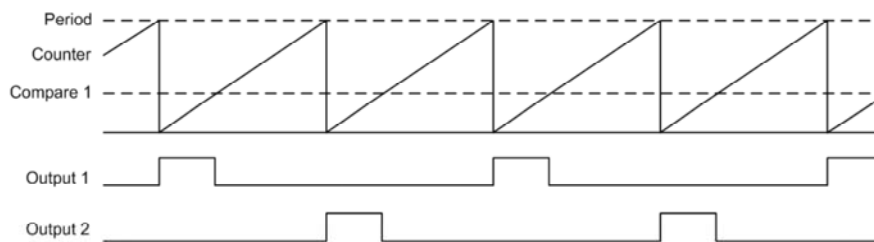
The output set occurs when the counter is equal to 2 and matches the HRTIM_CMP1R register value.

The HRTIM_CMP2R register is equal to 5 but is ignored until the external event falling edge occurs. The current counter value is 7. It is captured in the HRTIM_CPT1R register and added to the HRTIM_CMP2R preload value. This yields an effective Compare 2 value of 5 plus 7 equals 12.

When the counter matches this Compare 2 value, the output reset

event is finally generated, 5 cycles after the external event falling edge. The uncertainty of 0.5 cycle in the figure is the normal latency due to the capture signal resynchronization.

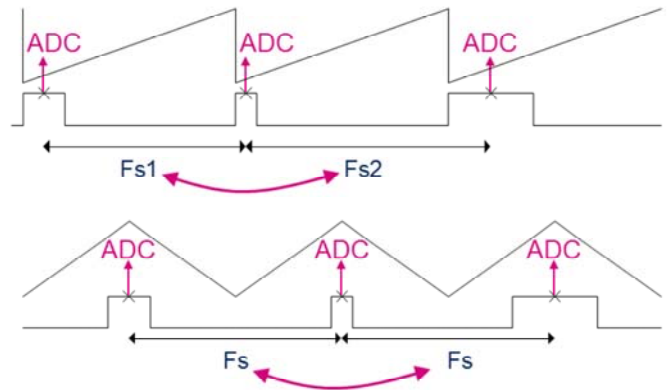
- To be used for push-pull converters, resonant converters, ...
- A single output is active at a single time, the others put in idle state
 - The crossbars of the 2 outputs must be programmed identically for perfect balancing, but it is not mandatory (this feature can be used for other purposes)
 - Available only in Continuous mode with roll-over
 - Example: Set on period, reset on Compare 1 match



The The HR Timer's push-pull mode simplifies the management of push-pull and resonant converters.

The two outputs of the timing unit are alternatively enabled and put in idle state to maintain the balance between the two PWM outputs. In this figure, two identical waveforms are programmed on Output 1 and Output 2. The push-pull circuitry disables the outputs every two cycles, out of phase, so that the signals can be built using a single compare value to set the PWM duty cycle. This is also possible without using the push-pull, but it would be at the expense of two other compare registers and a larger CPU burden. Finally, it must be noted that it is possible to unbalance the signals and generate asymmetrical waveforms: for instance a PWM using Compare 1 for Output 1 and a PWM using Compare 2 for Output 2.

- Useful for power supplies or motor-control applications, to control up to:
 - Two 3-phase brushless motor
 - Three DC or stepper motors with full-bridges
- Simplifies the ADC sampling
 - Constant sampling frequency
 - Sampling at the middle of the pulse



The HRTIMER counters can operate in Up-Down counting. This mode presents advantages for fixed switching frequency power converters and for motor control applications. The HRTIMER is able to control up to two 3-phase brushless motors, and three full bridge converters for driving steppers or DC motors.

The Up-down mode simplifies the ADC sampling implementation. At first, it allows a constant sampling period, since the output pulses are symmetrical to the top of the counting pattern (also called “crest” of the counting). This is shown on the bottom part of the figure, where we can see that the sampling frequency F_s is constant from one cycle to the other, while it is varying with the pulse-width using the up-only counting mode, on the upper part.

For many converters, the ideal ADC sampling point is in the middle of the output pulse, to avoid any conversion error due to the ripple on the output current or voltage.

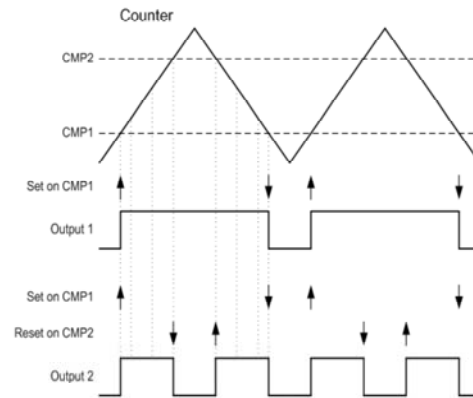
This ADC triggering point positioning requires some calculation in

Up-counting mode. In Up/Down counting mode, this is not necessary, as the ideal triggering point is always located on the crest, thus slightly reducing the ADC-related computational burden.

Programming the Up/Down mode

18

- The UDM bit allows to select Up-only and Up/Down mode
- The set/reset crossbar programming differs:
 - HRTIM_SETxyR sets the output during up-counting and resets it during down-counting
 - HRTIM_RSTxyR resets the output during up-counting and sets it during down-counting



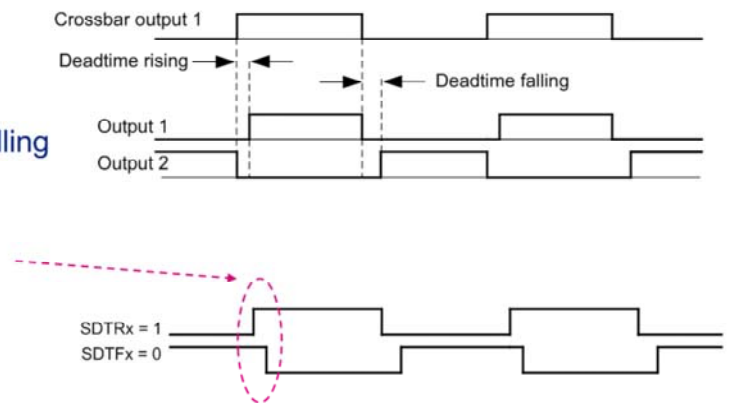
The UDM bit enables the Up/Down mode.

Most of the HRTIMER features are available in Up/Down mode, at the exception of Auto-delayed and Balanced-Idle modes. The HRTIMER programming mostly differs with the crossbar programming.

The events selected in the HRTIM_SETxyR register are coding for both the output set during up-counting and the output reset during down-counting. This allows pulses centered in the counting period, as shown on the figure for Output 1.

The events selected in the HRTIM_RESETxyR register are coding for both the output reset during up-counting and the output set during down-counting. This allows the creation of more complex waveforms, as shown on the figure for Output 2.

- Whenever there's a half-bridge, full-bridge, 3-phase inverter,...
- Versatile and safe deadtime setting
 - Deadtime is adjustable for both rising and falling edges (asymmetrical dead time)
 - Possibility to have negative deadtime for controlled overlap
 - Deadtime register is preloaded and can be updated during run-time
 - Typically for adaptive control, to optimize efficiency
 - For additional safety, possibility to have either deadtime sign or value write protected



A deadtime can be inserted after the crossbar to generate quasi complementary signals on the two outputs out of a single reference PWM. This allows safe driving of all topologies based on half-bridges, including full-bridges and 3-phase inverters, by inserting a deadtime during which the 2 power switches are turned-off. This avoids any problem of cross-conduction and shoot-through.

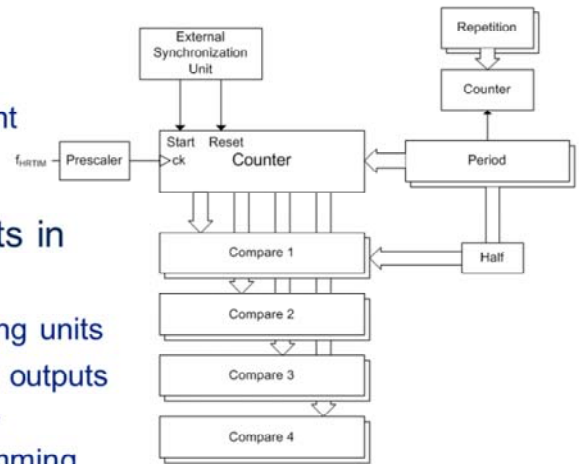
The deadtime value can be individually adjusted for both the rising and falling edges, asymmetrically, to handle any kind of driver or propagation delay. For very specific cases, it is even possible to program a negative deadtime and have some controlled outputs overlap.

The deadtime registers are preloaded, so that the timings can be adapted in real-time without any risk of spurious transitions. This is typically used for adaptive control to optimize the converter efficiency.

Last, for functional safety purposes, it is optionally possible to write protect the values or the signs of the deadtime with specific

control bits.

- Very same architecture as timing unit, simplified:
 - No outputs directly associated with:
 - No crossbar unit nor output related control
 - No capture unit, auto-delayed mode and external event management
- Primarily aims at synchronizing the 6 timing units in multi-timer configuration
 - The 6 events from the master are available for all timing units
 - They can be used to reset the counters or to set/reset outputs
 - Also holds counter enable bits from all timing units for synchronized starts and synchronization logic programming for interfacing with other high-resolution timers (internal or external)



The master timer comes with the 6 timing units, using the same architecture. Its programming is simplified since there are no directly associated outputs, nor captures, auto-delayed mode or external events management.

The master timer primarily aims at synchronizing the 6 timing units for converters requiring more than 2 outputs. The 6 events from the master (period, compare 1 to 4 and the global synchronization) are available on all timing units, and can be used to set/reset the outputs or reset the timers.

The master timer also holds the counter enable bits from all timing units. This allows a synchronized start of all timers with a single write access.

Last, it embeds the off-chip synchronization logic programming for interfacing with other HR Timer instances, in case of multiple MCU systems.

- Typical use case: cycle-by-cycle current control, zero current detection,...
- A set of 10 signals (among 30 sources) can be used to:
 - Set or reset the outputs
 - Reset the counters
 - Trigger
 - Burst mode start
 - Capture and auto-delayed compare
 - ADC start-of conversion
 - Delayed protections
- The 10 events are available on all timing units
 - Conditioning is done per event, filtering is done in each timing unit

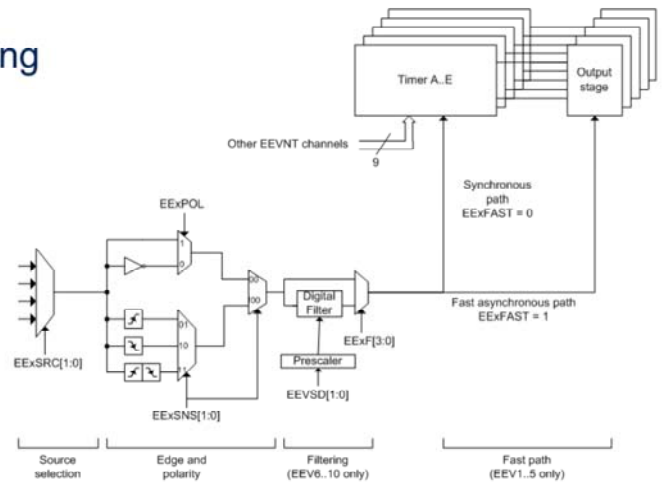


External events play a key role in the HR Timer. They dynamically modify the waveforms, typically to implement a cycle-by-cycle current control or to re-start the counter following a zero current detection.

A set of 10 external events (which can be chosen from among 30 on-chip or off-chip sources) are used to set, reset or toggle the outputs, reset the counters and trigger multiple features, such as the burst mode controller, the capture and auto-delayed compare, the ADC start of conversion or some protections.

The 10 external events are available for all timing units. While the conditioning (typically the edge sensitivity) is done globally, event per event, a different event filtering scheme can be applied to each timer.

- Determined by external conditions (interfacing circuitry, noise and oscillations,...)
- Source selection
- Programmable edge or level sensitivity
 - When level sensitive, the event is continuously generated as long as the level is active
- Digital filter (External events 6 to 10 only)



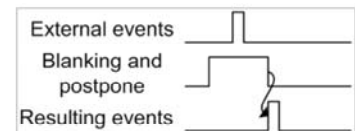
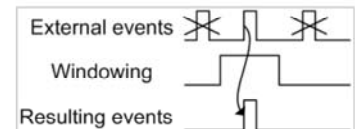
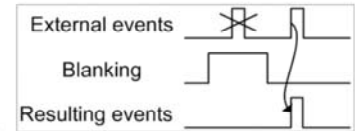
The external event conditioning circuitry is a preliminary signal detection stage, which is configured depending on the type of external interface or on the amount of noise and glitches present in the incoming signals.

The first stage is a 4-to-1 multiplexer to select the source, usually an input pin or a built-in comparator.

It is followed by the definition of the edge or level sensitivity. It must be noted that when the level-sensitive option is chosen, the external events are continuously issued as long as the level is active.

Last, a programmable digital filter stage allows removal of spurious transitions, at the cost of a higher response latency due to the filter propagation delay. This feature is available for external events 6 to 10 only.

- An active event is not necessary a valid event!
 - E.g. spurious over-current spikes due to diode recovery currents
- In each timing unit, possibility to apply if necessary a specific filtering scheme to each and every external event
- Blanking, to mask external events during a defined time
 - 12 blanking options: 4 within the timing units, 8 in other timers
- Windowing, to enable external events during a period
 - 3 windowing options: 2 within the timing units, 1 in other timers
- Possibility to postpone or latch events and have timeout



In many cases, an active event coming out of the conditioning stage is not necessarily a valid event. This is typically the case for current feedbacks. The freewheeling diodes recovery current might be a source of spurious over-current detection. It is therefore necessary to apply a specific filtering scheme to the external events used in a given timing unit.

Three filtering options are proposed.

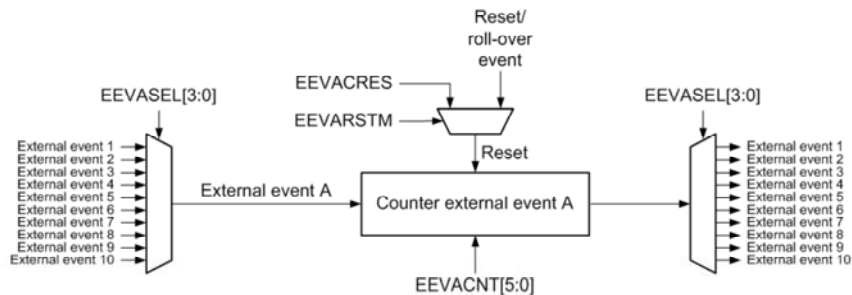
The blanking mode consists of masking the external events during a defined time window defined within the timing unit itself or using the other timing units. This is typically the case for leading-edge blanking.

The windowing mode lets the external events go through only during a specified period of time, again defined within the timing unit itself or using the other timing units.

An event postpone mode allows an event to be acknowledged and latched only if it occurs during a given time period, and delayed to the end of this period.

External Event counter (1/2)

- Event-based filtering: an external event must occur a given number of times before being considered valid
 - Typical use case: valley skipping mode for flyback converters
- Each timer unit has an external event counter, for any of the 10 events
- An event is generated once it has been counted N times (N=1 .. 32)

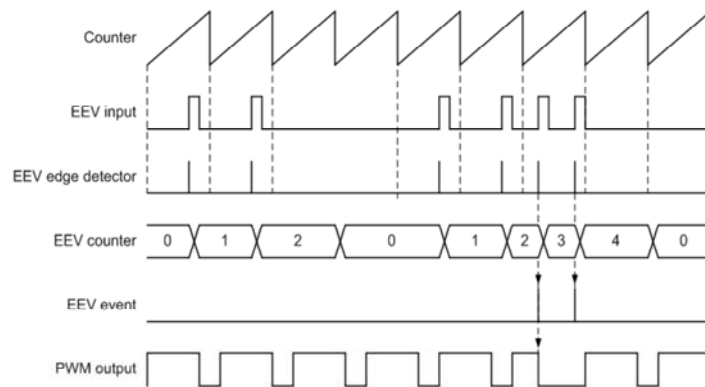


It is also possible to filter-out events based on the number of occurrences. This is typically the case for valley skipping mode in flyback converters: a new period must be started only after a given number of oscillations (valleys).

In this case, the external event counter allows the event to be taken into account only after a programmable number of occurrences from 1 to 32.

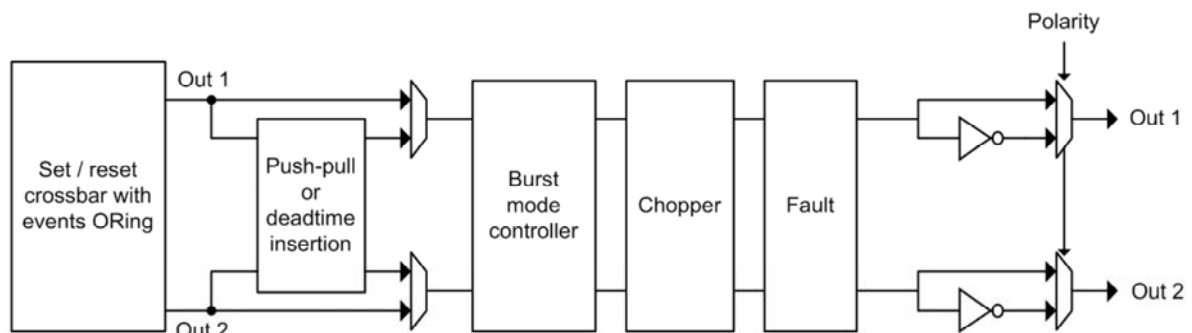
Each timer unit has its own external event counter, which can be applied to any of the 10 available external events.

- The external event counter has 2 operating modes:
 - Immediate mode: Event generated when N consecutive events occurred during a period
 - Event counter is reset on each and every new PWM period
 - Cumulative mode: event must occur at least once during multiple consecutive period
 - counter is reset only if the event did not appear during the last PWM period



Additionally, the external event counter has two operating modes. In the immediate mode, the event is generated as soon as N consecutive external events have occurred within a period. The event counter is reset at the end of the current PWM period. This mode can be used for implementing valley skipping for instance. In the cumulative mode, the external event counter is not reset at the end of the period, but only if there's no external event during a period. The event will be generated only if it has occurred during multiple consecutive periods, as shown on the right side of the figure, after 3 rising edges on the EEV input. On the contrary, on the left side of the figure, the external event counter is reset after the 3rd period, since there have been only 2 consecutive events without a 3rd one.

- Each output has 3 separate configuration bits
 - Enable (OEN), disable (ODIS), and disable status (ODS: Fault or Idle)
- 3 states
 - Run
 - Idle (at start or during burst and delayed protection)
 - Fault



Each of the ten HR Timer outputs have three configuration bits. The output is enabled using the OEN bit and disabled using the ODIS bits. All bits are grouped in a single register to have the possibility to enable and disable multiple outputs in a single write access. A disable status bit ODS indicates if the output was disabled because of a fault detection.

The bottom figure is an overview of the whole output stage, for a given timing unit. Following the set/reset crossbar, Output 1 and Output 2 reference signals can first be altered with the push-pull and deadtime insertion unit.

In Push-pull mode, Output 1 and Output 2 are alternatively in run or idle state. When deadtime is inserted, the Output 1 signal serves as reference to build the quasi-complementary signals and the Output 2 signal is discarded.

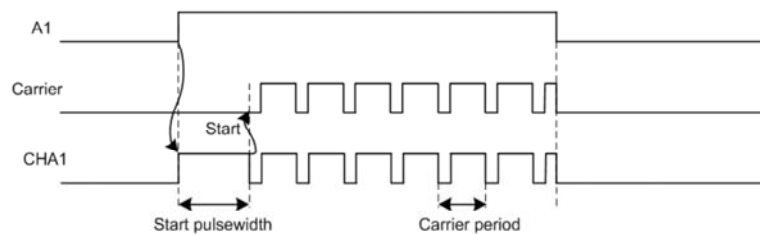
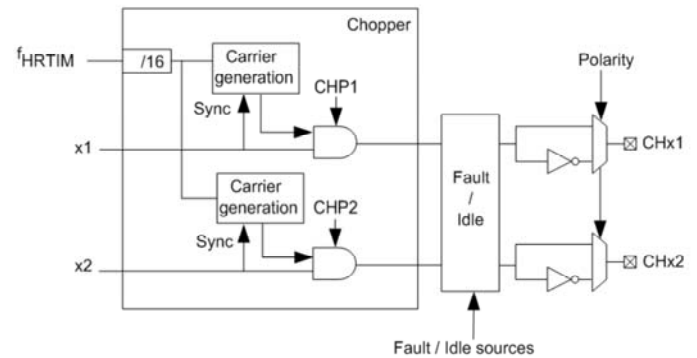
The burst mode controller allows both outputs to be automatically disabled and re-enabled for a limited period of time.

The chopper unit can superimpose a carrier signal on top of the active PWM waveform.

Finally the fault stage can disable completely the output to protect the power switches.

The polarity is programmed at the end. The entire HR Timer waveform can be programmed by just reasoning in terms of active / inactive switches, independently from the external gate driver and or interface sensitivity, before the polarity is applied.

- Allows driving isolation transformers without glue logic
- Parameters programmed for each timer
 - 16 frequencies (1.56 to 25 MHz)
 - Duty cycle (0/8 to 7/8)
 - 1st pulse length (40 to 630 ns)



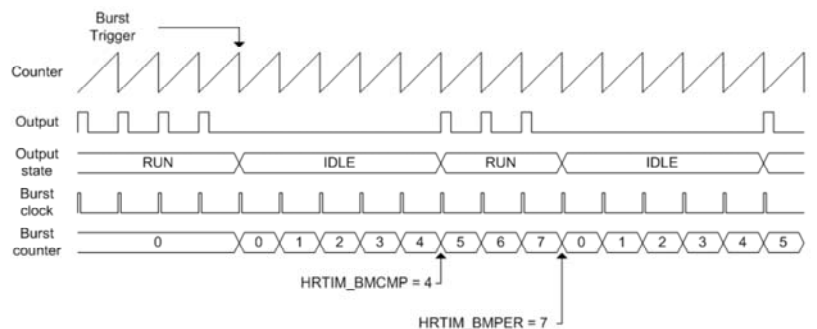
The chopper unit superimposes a programmable carrier waveform on top of the active PWM signal. A carrier generator is started by the incoming pulse and added to the incoming signal by means of an AND gate.

This allows to interface with isolation transformer drivers without having to add external glue logic.

The carrier waveform can be programmed with 3 parameters:

- the modulation frequency, from 1.56 to 25 MHz,
- the 1st pulse length to adjust the settling time,
- the duty cycle to set the sustain current.

- Handles automatically burst mode operation, whatever the number of timers involved in the converter, during light load operation
- Disables multiple outputs simultaneously
 - Programmable Idle state
 - Possibility to insert deadtime upon entry
 - Burst length and period programmable
 - Single-shot / continuous mode
 - Can be started on 32 events
 - Multiple burst clock options



Burst mode operation is of common use in power converters when operating under light loads. The burst mode controller allows the outputs alternatively in Idle and Run states, by hardware, so as to skip some switching periods with a programmable periodicity and length.

The burst mode controller can act on multiple timers in parallel. The outputs are disabled simultaneously, with a programmable idle state and the possibility to have a deadtime inserted upon burst mode entry, to avoid any deadtime violations.

The burst length is programmable as well as the repetition period when the burst mode is enabled in continuous mode.

The burst mode can be started on 32 events, and it has multiple burst clock options, either coming from the timer roll-over event or from independent user programmable period (based on the 400 MHz clock).

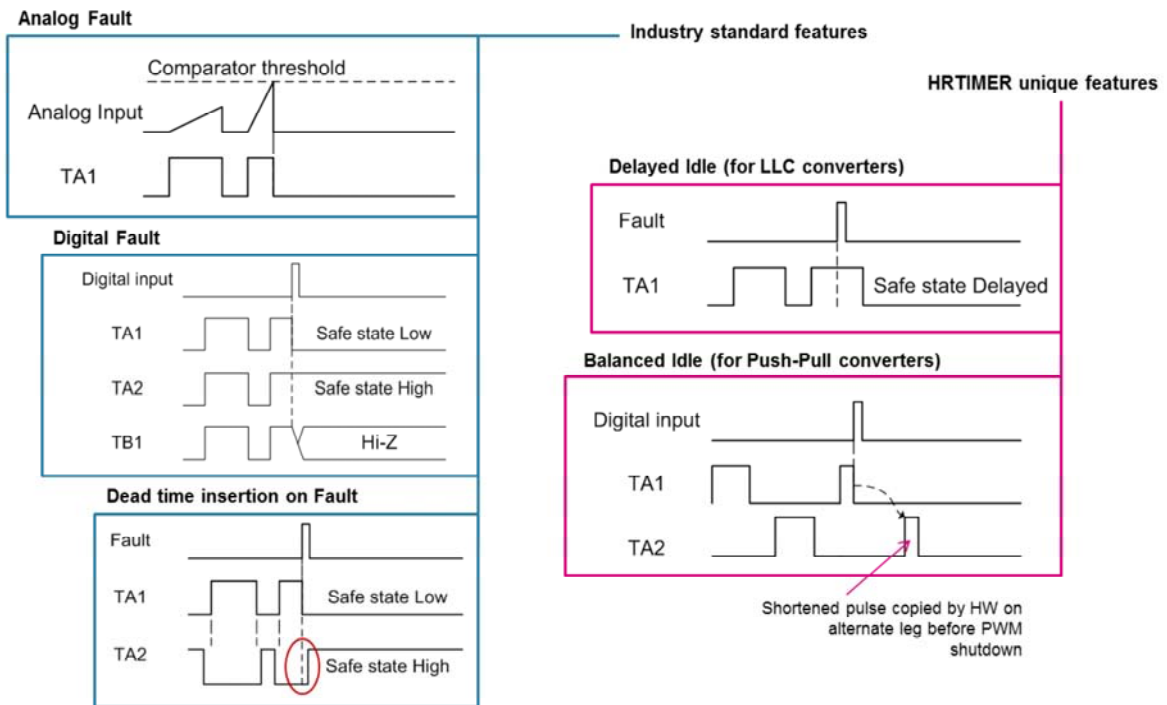
In this example, once the burst trigger arrives, the PWM is active for 3 periods every 8 cycles with pulses skipped for 5 periods. The burst mode controller counter is clocked and incremented on each

timer roll-over event.

shut-down.

Numerous fault management options

30



The HR Timer offers multiple fault protection options.

Some are industry standard features. The analog fault uses a comparator. When reaching a user-defined threshold, the comparator tripping causes the PWM outputs shutdown.

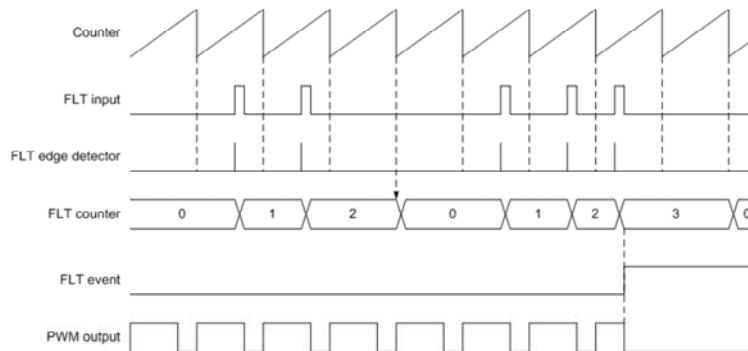
In digital fault mode, the system is put in safe state when receiving a digital fault trigger. For the cases where one of the outputs safe state is active, a deadtime can be automatically inserted right after the fault to guarantee a safe transition to the fault state without a deadtime violation.

The HR Timer however offers some unique protection features. For LLC converters, the delayed Idle maintains the current pulse on the output and the fault state is entered only after it is completed.

For the push-pull converters, the balanced idle mode guarantees a perfect balance between the two output pulse widths, when a fault is triggered, before the converter is actually entering the fault state. The output pulse which is shortened by the fault is automatically copied on the alternate output before stopping

operation.

- Possibility to add fault blanking with 2 options:
 - At turn-on (leading-edge blanking) or in between CMP3 and CMP4 events
- Fault counter with 2 operating modes
 - Immediate mode: fault generated when N consecutive faults occurred during a period
 - Cumulative mode: fault generated if a fault occurred once during multiple PWM period



The fault events can be filtered-out by two means.

It is possible to do fault blanking, so that faults are ignored during specific periods: at the beginning of the PWM cycle (this is also called leading-edge blanking), or during a window with programmable start and width, using the CMP3 and CMP4 compare registers.

Another option is to do filtering with the fault counter, do discard spurious fault information and consider only true errors. The fault counter can operate in two modes, similarly to the external event counter.

In the immediate mode, the fault is generated as soon as N consecutive external faults have occurred within a period. The fault counter is reset at the end of the current PWM period. In the cumulative mode, the external fault counter is not reset at the end of the period, but only when there's no fault during a period. The fault will be generated only if it occurred during

multiple consecutive periods, as shown on the right side of the figure, after 3 rising edges on the FLT input. On the contrary, on the left side of the figure, the fault counter is reset after the 3rd period, since there have been only 2 consecutive faults without a 3rd one.

Triggering ADC with HRTIM

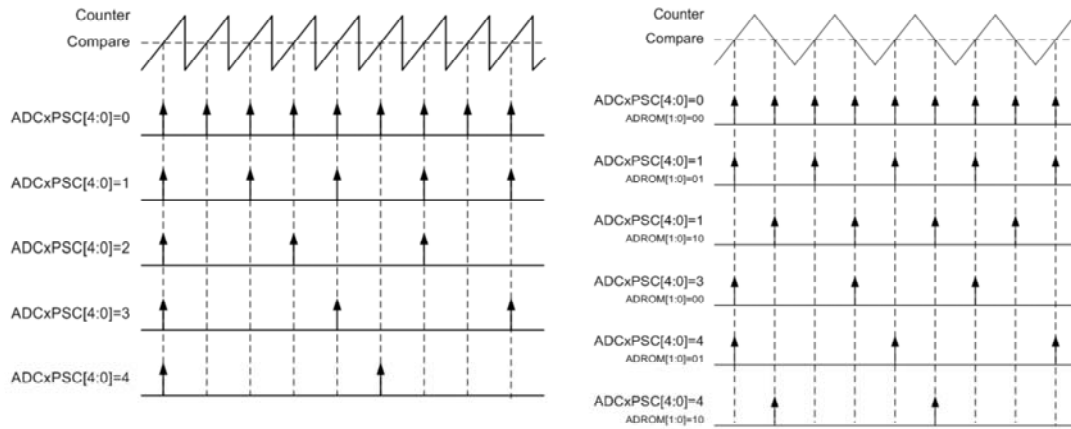
32

- The HR Timer has 10 ADC triggering channels
 - 2 channels per ADC, regular and injected
 - Allows to have up to 10 independent ADC processes in parallel for multiple converters
- Each triggering channel has 32 sources:
 - 4/5 events from each timing unit, including master, plus 5 external events
 - The triggering selection registers are preloaded for on-the-fly trigger source update
- Possibility to enable multiple sources simultaneously for triggers 1..4
 - Typical use case: interleaved converters
- One single source possible for triggers 5..10



The HR Timer has 10 ADC triggering channels, with 2 channels per ADC, for the regular and the injected sequencers. This allows up to 10 independent ADC processes in parallel for multiple converters. Each triggering channel has 32 sources, including 5 from the master timer, 5 from external events and 4 or 5 for each timing unit. It is possible to simultaneously enable multiple sources for the triggers 1 to 4, when the various trigger instants are clearly defined and do not overlap, as is typical in multiphase converters. The triggers 5 to 10 only allow one event at once. Last, the triggering selection registers are preloaded for on-the-fly trigger source updates.

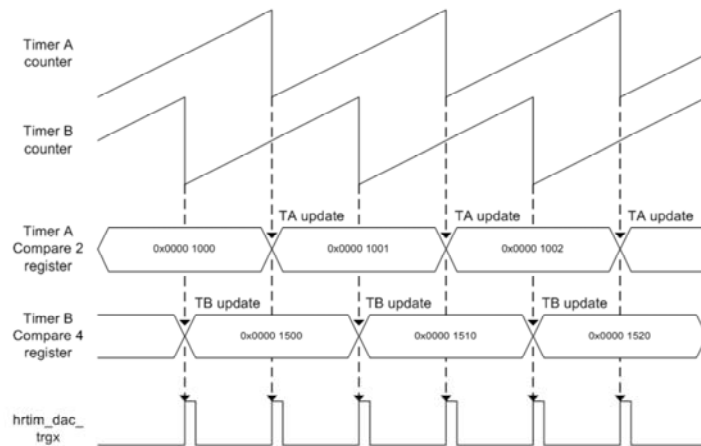
- For high switching-frequency application, it is possible to reduce the ADC triggering rate with the ADC post-scaler
- Each ADC trigger can be individually adjusted down to 1 out of 32



The ADC triggering rate can be reduced with the ADC post-scaler. This allows the decrease of the number of conversions and ADC interrupt service routines, for high PWM switching frequency applications.

Each of the 10 ADC triggers can be reduced down to 1 conversion per 32 counter periods, as shown on the 2 figures. The right side figure shows that the ADC triggers can also be positioned during the up or down phase of the counting period, using the ADROM[1:0] bits.

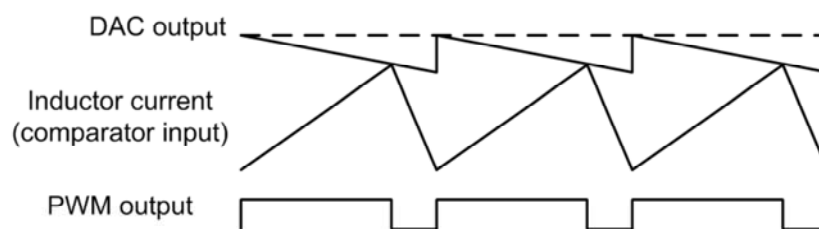
- DACs can be updated synchronously with the timer update events from master and timer units, at PWM period pace
- 3 outputs, with possibility of multiple sources per DAC



The HR Timer also offers DAC triggers. This gives the possibility to have the DAC values updated synchronously with the timer update events from the master and the slave timers.

This is typically used for peak-current mode, where the DAC is used as a threshold for a comparator. It allows the new DAC value to be applied right at the beginning of the next switching period. 3 DAC triggers are available, and it is possible to have multiple concurrent sources for the same trigger.

- This is done by applying a negative saw tooth shape (slope) on the current comparator, using the DAC
- This saw tooth must be synchronized with the PWM. The HRTIMER features two DAC triggers:
 - For resetting (synchronizing) the saw tooth
 - For decrementing the DAC values



The HRTIMER allows the ease implementation of slope compensation techniques. This is necessary for peak current-mode converters. In this case the PWM output is turned off by a comparator comparing the current level with a threshold set by the DAC. To guarantee the stability of the power supply in this configuration, it is necessary to apply a descending slope (a saw tooth) on the DAC output connected to the comparator, as shown on the figure.

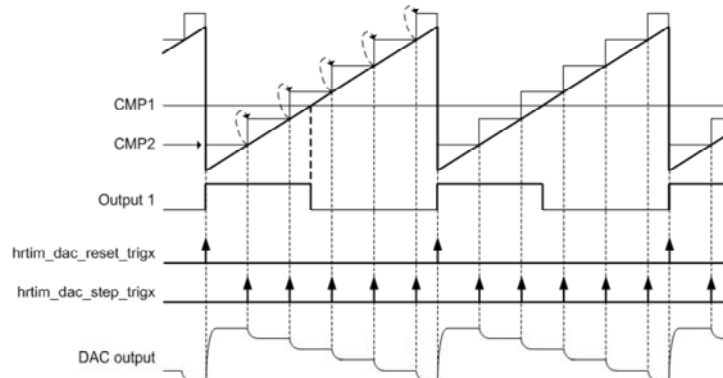
This saw tooth must be synchronized with the PWM generation. This is achieved with two DAC triggers, one to reset the saw tooth signal (reloading a start value), and another one to regularly decrement the DAC values.

The saw tooth start value and amplitude are defined in the DAC peripheral.

Dual channel DAC triggers

36

- 2 control bits allow the configuration of the DAC reset and DAC step events
 - Usually, DAC reset on period and DAC steps on CMP2
 - Other option possible (DAC square wave for hysteretic control)
- The compare 2 (CMP2) defines the number of steps



The HRTIMER's Dual channel DAC triggers circuitry can be programmed to adjust the trigger generation timings. In the default slope compensation configuration, the DAC reset trigger is sent on the PWM period beginning. The DAC step trigger (to decrement the DAC output voltage) is generated with the compare 2 matches the counter value. This register defines the number of steps during a period: the CMP2 value is automatically recomputed by the HRTIMER after each compare match, as shown on the figure, so that it can match multiple times per counting period. The compare 2 register value must only be programmed once, dividing the period value by the expected number of steps. Note that the staircase effect due to the small number of steps in the figure is not representative from a real application, and is just for illustration purposes. The high speed DAC allows a much higher number of steps and quasi linear waveforms.

- All working registers are duplicated (preload / active registers)
- 2 control bits allows to handle preload mechanism
 - Preload enable bits (PREEN) in each timer
 - Update disable bits (MUDIS, TxUDIS) to temporary suspend update mechanism (resumed when UDIS bits are reset)
- Numerous sources available
 - Within the timer itself (repetition, reset)
 - From other timers (master, Timers A to F)
 - From on-chip sources (general-purpose timers)
 - Following burst mode DMA



All HR Timer working registers are duplicated with a shadow register. This applies to the period and compare registers, and numerous other configuration registers. It allows glitch-free operation when updating multiple registers. The write accesses are done to the preload registers, which are transferred in the active registers when an update event occurs, usually at the beginning of a PWM cycle.

Two control bits allows handlong of the preload mechanism. A preload enable bit PREEN, in each timer, globally enables the register shadowing.

The update disable bits MUDIS and TxUDIS can temporary suspend the update event for complex HR Timer updates or on-the-fly HR Timer reconfigurations.

Numerous update trigger sources are available, within the timer itself, from other timers (master, Timers A to F), other on-chip sources (general-purpose timers) or following DMA burst mode.

Interrupts: 8 vectors and 100 sources

Interrupt event	Master	Timers A to F	Description
Synchronization event received	x		For chaining multiple HR Timers
Delayed protection triggered		x	Advanced fault protection schemes
Counter reset or roll-over event		x	PWM interrupt occurring on each and every cycle
Output 1 and Output 2 reset		x	Whenever an interrupt must be synchronized to a change of the output
Output 1 and Output 2 set		x	
Capture 1 and 2 events		x	Allows the capture value to be processed
Timing unit registers update	x	x	Register content transferred from preload to active
Repetition event	x	x	Periodical PWM interrupt for control loops
Compare 1 to 4 event	x	x	Indicates a compare match event
System fault	x	x	Indicates a MCU level fault
Fault 1 to 6	x	x	Indicates an external fault
Burst mode	x	x	Burst mode period
DLL ready	x	x	DLL calibration done



The HR Timer is able to generate interrupts from a total of 100 sources. They are dispatched on 8 interrupt vectors as following: one for each timing unit and the master timer, plus one vector dedicated for fault management.

The table lists all the sources and gives a brief description of each.

DMA: 7 requests and 91 sources

DMA request	Master	Timers A to F	Description
Synchronization event received	x		DMA transfer when chaining multiple HR Timers
Delayed protection triggered		x	DMA transfer in case of advanced fault trigger
Counter reset or roll-over event		x	DMA transfer on each and every PWM interrupt
Output 1 and Output 2 reset		x	DMA transfer on a change of the output
Output 1 and Output 2 set		x	
Capture 1 and 2 events		x	DMA transfer to store the captured values
Timing unit registers update	x	x	DMA transfer when register content has been transferred from preload to active
Repetition event	x	x	DMA transfer on PWM update (multiple periods)
Compare 1 to 4 event	x	x	DMA transfer on compare match

- The timer can be dynamically re-configured using burst mode DMA transfers
 - A single event can trigger the transfer of a table in the memory into selected HRTIM registers
 - Once the burst is completed, a global register update (transfer from preload to active) is performed as per selected update scheme



The HR Timer can also trigger DMA transfers from a total of 91 sources. They are dispatched on 7 DMA channels; one for each timing unit and the master timer.

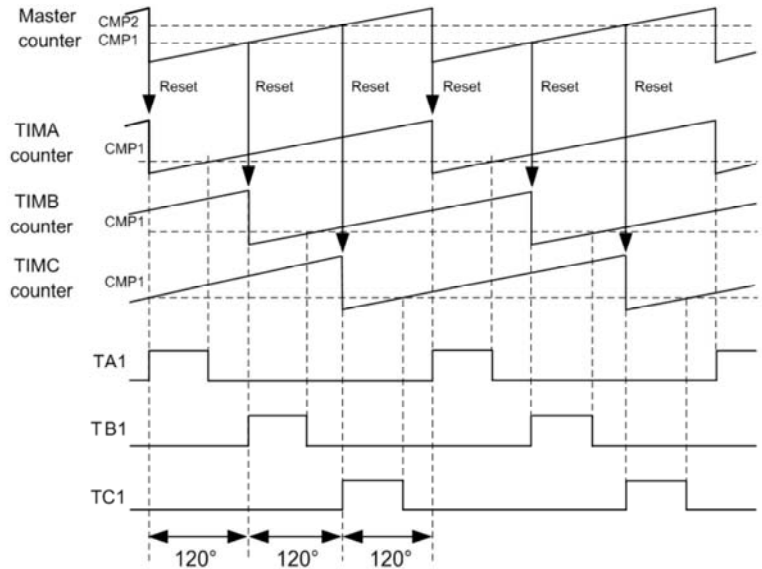
The table lists all the sources and gives a brief description of each. The DMA can also be used to dynamically re-configure the HR Timer during converter run-times. This is done using the DMA burst mode. Once it is triggered, the content of a table in the memory is transferred into the HR Timer control and configuration registers. The DMA burst control registers specify the registers to be updated, one by one.

Once the transfer is completed, a global register update is issued to have the new values transferred from the preload to the active registers.

Mode	Description
Run/ Low-power run	Active.
Sleep/ Low-power sleep	Active. Peripheral interrupts cause the device to exit Sleep mode.
Stop 0, Stop 1	Frozen. The High Resolution Timer is not functional and must be disabled.
Standby	Powered-down. The peripheral must be reinitialized after exiting Standby mode.
Shutdown	Powered-down. The peripheral must be reinitialized after exiting Standby mode.

The HR Timer peripheral can be active only in Run and Sleep modes. In Stop and Standby modes, the HR Timer must be disabled.

- Interleaved converter
 - 3 step-down converters running in parallel with a 120° phase-shift to minimize voltage output ripple and to spread input current demand
 - The master timer is responsible for shifting the 3 timers (A, B and C), using the counter reset
 - The duty cycle is adjusted by the control loop in each timer



This slide gives a practical use case of the HR Timer.

It shows how to generate the PWM signals for a triple interleaved converter. Three step-down converters are running in parallel with a 120° phase-shift to minimize the output voltage ripple and to spread the input current demand.

They must be synchronized to guarantee a constant phase-shift and eventually switch off one phase, adjusting the phase-shift to 180°.

This is done using the master timer. The events generated on master's period, Compare 1 and Compare 2 reset Timers A, B and C, respectively.

The pulse width on TA1, TB1 and TC1 outputs are then programmed using the Compare 1 register of each of the timing units, as per the control loop demand.

If we consider the TA1 output, it is built with the output set on master period event, output reset on Timer A Compare 1 event. The output TB1 is set on Master Compare 1 event and reset on Timer B Compare 1 event.

The output TC1 is set on Master Compare 2 event and reset on Timer C Compare 1 event.

Note that it is very simple to have an ADC trigger issued in the middle of the pulse width, using the spare compare registers in each timing unit.

Related peripherals and resources

42

- Refer to these trainings related to this peripheral:
 - RCC
 - Interrupts
 - DMA
 - Timers
 - DAC
 - ADC
- Documentation
 - HR Timer Cookbook (Application note AN4539)
 - STM32F334 application notes and user manuals (AN4885, AN4449, UM1733, UM1735, and UM1736)



The peripherals listed here influence HR Timer behavior. Please refer to the corresponding training for more information.

Multiple application notes and user manuals are available for the HR Timer for the STM32F334. They are partially valid also for the STM32G4 and can help developers better understand how to use the HR Timer.