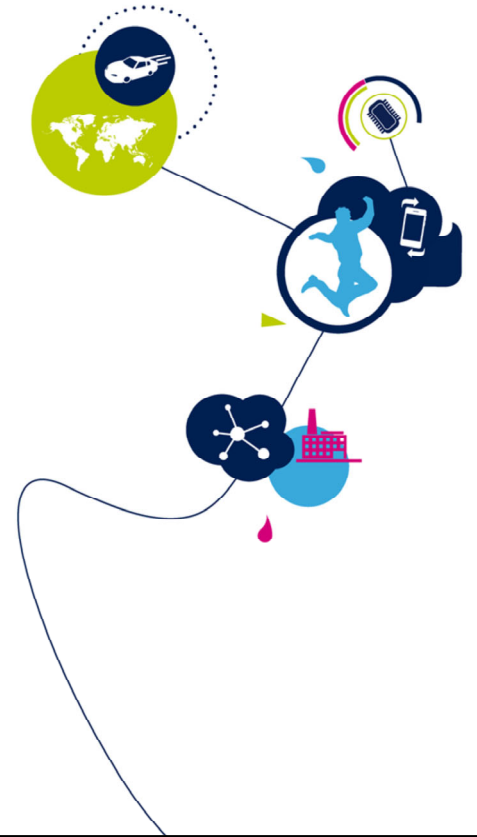


STM32WB – WWDG

System Window Watchdog

Revision 1.0



Hello, and welcome to this presentation of the STM32 system window watchdog (WWD). It will cover the main features of this peripheral used to detect software faults.

- Used to detect the occurrence of software faults

- WWDG counter must be refreshed within a time window
- Generates a system reset when a programmed time period expires
- Can be programmed to detect abnormally late or early application behavior
- Can't be disabled once activated and needs to be refreshed

Application benefits

- Best suited for applications which require the watchdog to react within an accurate time window.
- Configurable time window
- Early Wakeup Interrupt (EWI) available before reset happens



The window watchdog is used to detect the occurrence of software faults.

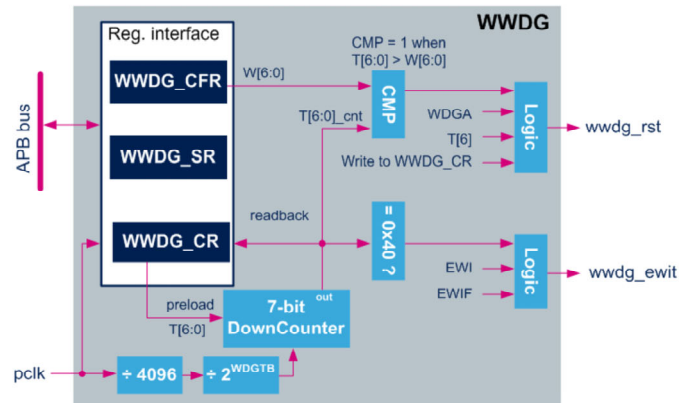
The window watchdog can be programmed to detect abnormally late or early application behavior.

It is best suited for applications required to react within an accurate timing window.

Once enabled, it can only be disabled by a device reset.

An Early Wakeup Interrupt can be generated before a reset happens to perform a system recovery or manage certain actions before a system restart.

- WWDG main features
 - Programmable timeout value
 - Programmable time window width
 - Reset generation:
 - When the timeout value is reached
 - When it is refreshed outside the time-window
 - Early wakeup interrupt (EWI)
 - Generated before the timeout value is reached



The window watchdog offers several features:

- The user can program the timeout value and the window width according to application needs.
- It can generate a reset under two conditions:
 - when the downcounter value becomes less or equal to 0x3F, or
 - when the watchdog is refreshed outside the time-window.
- It can generate an early wakeup interrupt when the downcounter reaches 0x40.

The early wakeup interrupt can be used to reload the downcounter in order to avoid a reset generation, or to manage system recovery and context backup operations.

As shown in the figure, the window watchdog uses the APB clock (pclk) as reference clock for its time-base.

The pclk is provided by the RCC block.

This clock is divided by 4096, and by a value programmed by the application.

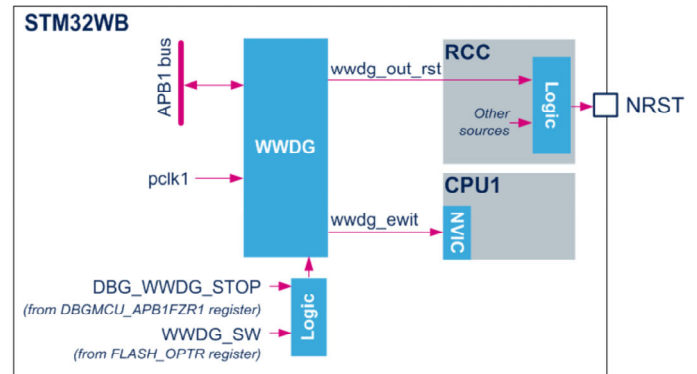
The application can also program the reload value of the downcounter bits T[6:0].

The window width is controlled by bits W[6:0].

Watchdog Integration

4

- The WWDG is dedicated to CPU1
- The WWDG can be automatically enabled after reset via WWDG_SW option bit
- The WWDG can be frozen when the CPU1 is in Debug mode via DBG_WWDG_STOP
- The WWDG can generate:
 - An early wakeup interrupt for the CPU1
 - A system reset



The STM32WB's embedded WWDG is dedicated for use by the CPU1 only.

The WWDG is connected to the APB1 bus and clocked by the APB1 clock.

The WWDG early wakeup interrupt output is connected to the NVIC of the CPU1.

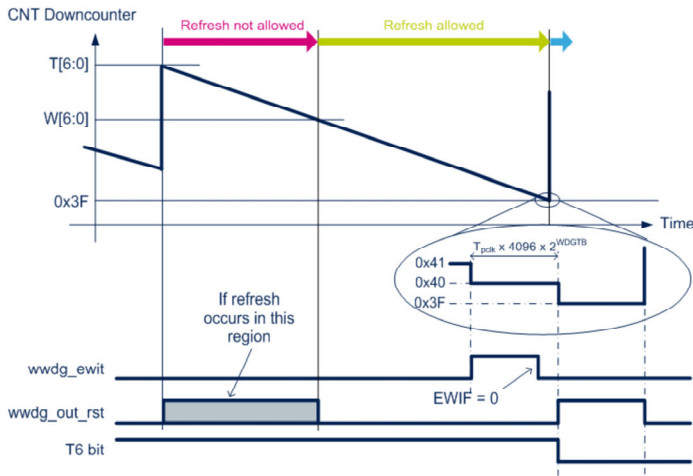
It is possible to select the hardware or software start, via an option byte. In hardware mode, the WWDG can be automatically enabled after reset.

The window watchdog is frozen when the system is in Stop, Standby or Shutdown mode, but can remain active when the CPU1 is in CSleep mode.

The WWDG performs a system reset handled by the RCC block, when a timeout occurs or when the WWDG is refreshed outside the allowed window.

WWDG functional description

5



If the software reloads the counter while the counter is greater than the value stored in W[6:0], then a reset is generated.

To prevent a reset, write the reload value T[6:0] when the down-counter is lower than W[6:0] value, and greater than 0x3F.

When the down-counter transitions from 0x40 to 0x3F, it initiates a reset.



This diagram illustrates how the window watchdog operates. When the 7-bit downcounter rolls over from 0x40 to 0x3F, the watchdog asserts the reset. This happens if the application software does not refresh the window watchdog on time.

The early wakeup interrupt, if enabled can be generated when the downcounter reaches 0x40.

If the software refreshes the watchdog while the downcounter is greater than the value stored in bits W[6:0], a reset is generated.

This happens when the application refreshes the watchdog too early. No interrupt is generated in this case.

To prevent a window watchdog reset, the watchdog refresh must happen while the downcounter value is lower than the time-window value, and greater than 0x3F.

This is illustrated by the green area.

The refresh operation consists on reloading the downcounter with bits T[6:0].

- Enabling the WWDG in software mode:
 - In the RCC block:
 - Set the WWDGEN bit to '1' in order to provide the APB clock to the watchdog
 - Set the WWDGSMEN bit to '1' in order to keep the watchdog running in Sleep mode
 - In the WWDG block:
 - Set WDGA bit to '1'
- Enabling the WWDG in hardware mode:
 - The WWDG is running as soon as the CPU1 is in Run or Sleep mode.



The WWDG can work either in hardware or software mode. In software mode, the application needs to enable the APB1 watchdog clocks via the RCC, and set the bit WDGA to '1' in the WWDG, in order to enable the watchdog.

Note that once the APB1 clock for the watchdog is enabled, the application cannot disable it. Only a system reset can disable the watchdog clock.

The low-power enable bit can be set as well if the application wishes to keep the window watchdog activated, even if the CPU1 is in Sleep mode.

In hardware mode, there is no need to enable the watchdog: the WWDG is counting-down when the CPU1 is in Run or Sleep. The bits WWDGEN and WWDGSMEN are forced to '1' by the hardware.

WWDG settings and reset flag

7

- Setting the WWDG time-base:
 - WWDG time-base pre-scaled from pclk clock
 - 4096 internal divider and 4 pre-dividers: 1, 2, 4, 8 selectable by register WWDG_CFR
 - Setting the WWDG timeout by using the following formula:
$$t_{\text{WWDG}} = t_{\text{PCLK}} \times 4096 \times 2^{\text{WDGTB}} \times (T[5:0] + 1)$$

In hardware mode, a reset is generated after 262144 cycles of APB clock (t_{PCLK}), if the watchdog is not refreshed

- Checking the WWDG reset source:
 - Reset flags in the RCC block indicate when a WWDG reset occurs (after device reset)



The downcounter uses the APB1 clock, divided by 4096, and again divided by a division ratio selected by the application. This division ratio can be 1, 2, 4 or 8, as defined in the WWDG_CFR register.

The formula shown in this slide lets you determine the watchdog timeout value.

When a system reset occurs, it is possible to identify the cause of the reset thanks to status flags provided by the RCC block.

The window watchdog can be one of the sources.

Interrupt event	Description
EWI	Early Wakeup Interrupt. It can be used in specific safety operations or when data logging must be performed before the actual reset is generated.

- An EWI occurs when the downcounter value reaches 0x40
- An EWI is enabled by setting the EWI bit in the WWDG_CFR register
- An EWI is cleared by writing “0” to the EWIF bit in the WWDG_SR register



The Early Wakeup Interrupt (EWI) can be used in order to perform emergency tasks before the reset occurs, such as:

- Data logging,
- Data protection,
- Watchdog refresh in order to prevent the reset, or
- Other emergency tasks...

The EWI occurs whenever the downcounter value reaches 0x40.

It is enabled by setting the EWI bit in the WWDG_CFR register.

The EWI is cleared by writing “0” to the EWIF bit in the WWDG_SR register.

Low-power modes

9

Mode	Description
Run Low-power Run	Active.
Sleep, Low-power Sleep	Always active in hardware start mode. Can be disabled, in software start mode via WWDGSMEN bit in the RCC.
Stop0, Stop1, Stop2	Not available.
Standby	Not available.
Shutdown	Not available.



The window watchdog is active in Run and Sleep modes. It is not available in Stop, Standby or Shutdown mode. In Sleep mode, when the watchdog is used in software start mode, it can be disabled by clearing the low-power enable bit located in the RCC block.