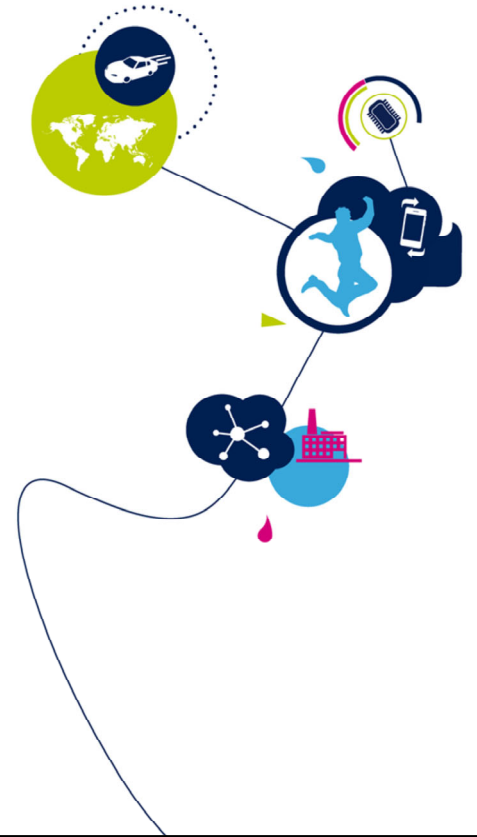


# STM32WB – IWDG

Independent watchdog

Revision 1.0



Hello and welcome to this presentation of the STM32 independent watchdog (IWDG). It covers the main features of this peripheral which can be used either as a watchdog to reset the microcontroller when a problem occurs, or as a free-running timer for application timeout management.

- Serves to detect and resolve malfunctions due to software failures:
  - Triggers a system reset when it is not refreshed within the expected time window
  - Always active even if the main clock fails
  - Cannot be disabled once activated, and needs to be refreshed

### Application benefits

- Totally independent process outside the main application
- Selectable hardware or software start
- Selectable low-power freeze in Standby or Stop modes



The independent watchdog is used to detect and resolve malfunctions due to software failures.

It triggers a reset sequence when it is not refreshed within the expected time-window.

Since its clock is an independent 32-kHz low-speed internal RC oscillator (LSI), it remains active even if the main clock fails.

Once enabled, it forces the activation of the low-speed internal oscillator, and it can only be disabled by a reset.

One of the main benefits for applications is its ability to run independently from the main clock.

- IWDG main features
  - Programmable timeout range from 125  $\mu$ s to 32.8 seconds
  - Programmable time-window width
  - Clocked from an independent RC oscillator (LSI)
  - Generates a reset when:
    - The timeout value is reached
    - The refresh occurs outside the window
  - Can be frozen in Debug, Stop or Standby mode
  - Can be configured to be automatically enabled



The independent watchdog offers a wide range of timeout values: from 125 microseconds to 32 seconds.

It is clocked by a 32-kHz RC oscillator which cannot be disabled when the independent watchdog is enabled.

It generates a reset when the programmed timeout value elapses, or when a watchdog refresh occurs outside a programmed time-window.

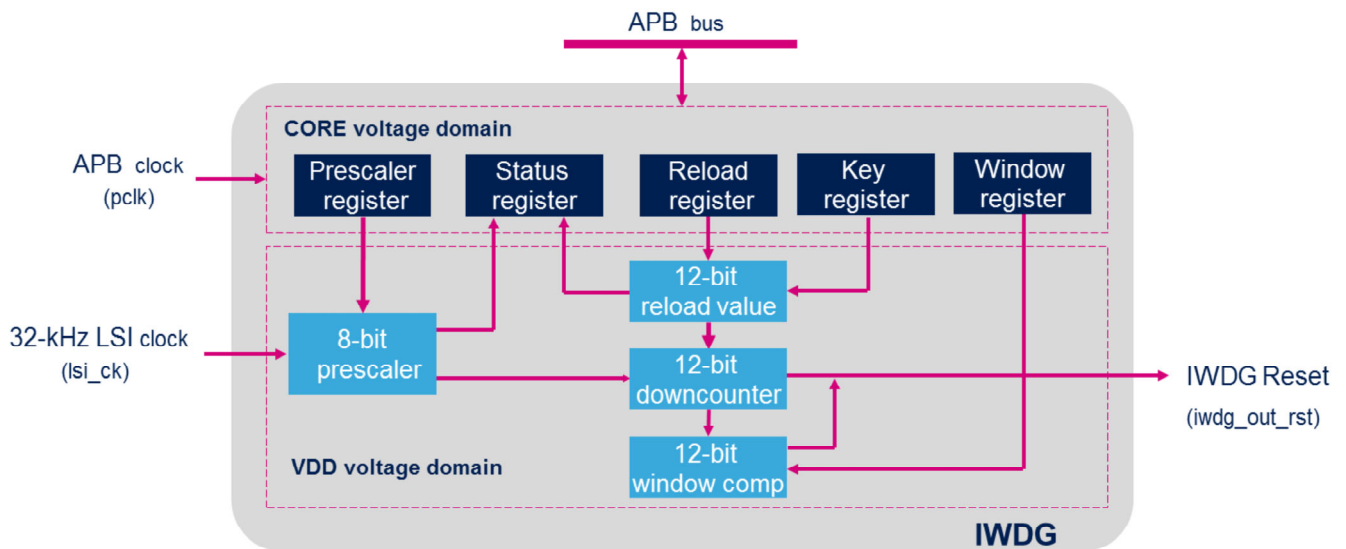
This window feature is optional, and not present in all independent watchdogs.

It is possible to automatically enable the independent watchdog after a system reset.

It is possible to define the behavior of the independent watchdog in Debug, Stop or Standby mode.

## Block diagram

4



The independent watchdog registers are located in the CORE voltage domain while its functions are in the VDD voltage domain.

Two clocks are needed:

The APB clock is required in order to access registers

The LSI clock is required for the functional part of the watchdog

This architecture allows the independent watchdog to work even in Stop and Standby modes.

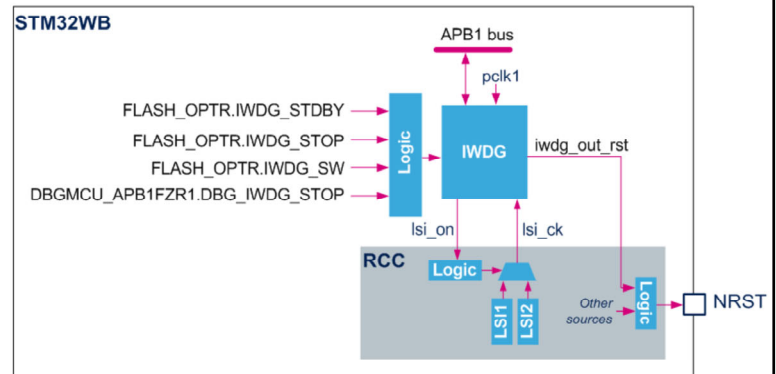
A programmable 8-bit prescaler is used to divide the LSI oscillator frequency.

The 12-bit downcounter defines the timeout value.

# Watchdog Integration

5

- IWDG is clocked by LSI1 or LSI2
- Can be automatically enabled after reset via IWDG\_SW option bit
- IWDG count-down can be frozen:
  - When the product is in Stop mode, via IWDG\_STOP option bit
  - When the product is in Standby mode, via IWDG\_STDBY option bit
  - When the CPU1 is in Debug mode, via DBG\_IWDG\_STOP



The STM32WB microcontroller includes an independent watchdog (IWDG).

The IWDG can be clocked by either LSI1 or LSI2. The IWDG will use whichever is enabled.

If both LSI are disabled and the IWDG is enabled, the LSI1 is forced on.

Note that LSI1 and LSI2 are both low-power, 32-kHz RC oscillators.

The IWDG is normally used by the CPU1 (Cortex M4).

It is possible to select the hardware or software start via option bytes.

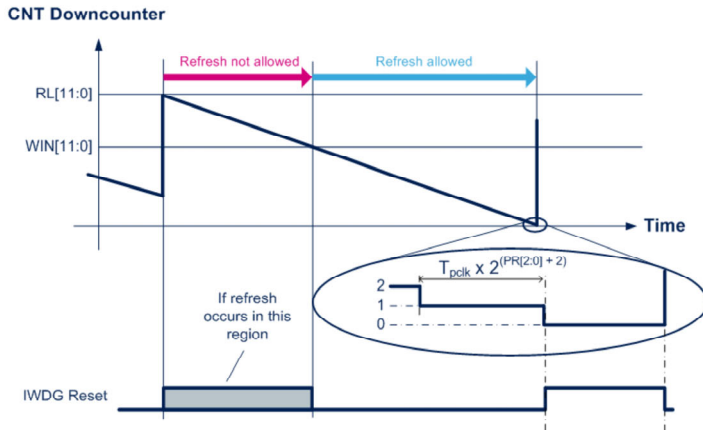
It is possible to select if the watchdog will freeze or not when the CPU1 is in Debug (core halted) mode.

Finally, it is also possible to control the behavior in Stop or Standby modes.

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

# IWDG functional description

6



If the software reloads the counter while the counter is greater than the value stored in WIN[11:0], then a reset is generated

To prevent a IWDG reset, refresh the watchdog while the counter value is lower than the time-window value WIN[11:0].



This diagram illustrates how the independent watchdog operates.

When the downcounter reaches zero, the watchdog reset is activated.

This happens when the application software did not refresh the window watchdog on time.

If the software refreshes the watchdog while the downcounter is greater than the value stored in the Window register, then a reset is generated as well.

To prevent a watchdog reset, the refresh must occur when the downcounter value is other than zero, and lower than the time-window value.

# Configuring IWDG hardware start

7

- At IWDG hardware start, the IWDG is automatically enabled after a system reset.
- After a system reset, the reload counter is set to 0xFFF and the prescaler to 0.
  - ➔ Timeout value of about 0.5 seconds
- Key register (IWDG\_KR) must be written by the software with 0x0000 AAAA at regular intervals before the counter reaches 0 and within the window (if the window option is enabled)



The independent watchdog hardware is enabled by the device's option bytes.

If the hardware mode is enabled, after every system reset, the watchdog automatically loads the counter with 0xFFF, and starts to count down.

The prescaler is set to 0, providing a division by 4 on the input clock.

To prevent any reset, the Key register must be refreshed at regular intervals before the counter reaches 0 and within the time window, if this option has been selected.

Considering that the LSI1 or LSI2 clock frequencies are at exactly 32 kHz, the application has about 0.5 seconds to refresh the IWGD before the generation of a watchdog reset.



# Configuring IWDG software start

8

- Enable IWDG by writing 0x0000 CCCC in register IWDG\_KR
- Enable register access by writing 0x0000 5555 in register IWDG\_KR
- Set IWDG prescaler by programming register IWDG\_PR
- Set the reload register (IWDG\_RLR)
- Wait for the registers to be updated (IWDG\_SR = 0x0000 0000)
- **Window option enabled:** Write the window value in the IWDG\_WINR register. This automatically refreshes the counter value IWDG\_RLR.
- **Window option disabled:** Refresh the counter value with IWDG\_RLR by writing 0x0000 AAAA in register IWDG\_KR



The independent watchdog software start is configured in only a few steps.

- The first step is to write the Key register with value 0x0000 CCCC which starts the watchdog.
- Then remove the independent watchdog register protection by writing 0x0000 5555 to unlock the key.
- Set the independent watchdog prescaler in the IWDG\_PR register by selecting the prescaler divider feeding the counter clock.
- Write the reload register (IWDG\_RLR) to define the value to be loaded in the watchdog counter.

After accessing the previous registers, it is necessary to wait for the IWDG\_SR bits to be reset in order to confirm that the registers have been updated.

- Two options are now available: enable or disable the



independent watchdog window option.

- To enable the window option, write the window value in the IWDG\_WINR register.
- Otherwise, refresh the counter by a writing 0x0000 AAAA in the Key register to disable the window option.

# IWDG settings and reset flag

9

- Setting IWDG time-base:
  - IWDG time-base prescaled from LSI1 or LSI2 clocks (32 kHz)
    - 7 pre-dividers: 4 to 256 selectable by IWDG\_PR register (and 12-bit watchdog counter reload value, RLR[11:0])
  - Set the IWDG timeout by using the following formula:

$$t_{\text{IWDG}} = t_{\text{LSI}} \times 4 \times 2^{\text{PR}} \times (\text{RL} + 1)$$

where  $t_{\text{LSI}} = 1/32000 = 31.25 \mu\text{s}$ , PR and RL are fields of IWDG registers

- The cause of the IWDG reset can be identified via RCC registers



The IWDG time-base is prescaled from the LSI1 or LSI2 clock at 32 kHz. The IWDG\_PR prescaler register can divide the LSI clock frequency by 4 up to 256. The watchdog counter reload value is a 12-bit value written in the IWDG\_RLR register.

A formula can be used determine the independent watchdog timeout. The independent watchdog time is based on the LSI period and its prescaler, as well as the selected watchdog counter reload value.

Note that the microcontroller's Reset and Clock Controller (RCC) provides registers identifying the source of the reset. In this way, the application can check if a reset is caused by an independent watchdog.

# Low-power modes

10

Mode	Description
Run	Active*
Low-power run	Active*
Sleep	Active*
Low-power sleep	Active*
Stop 0/Stop 1/Stop 2	Active*
Standby	Active*
Shutdown	Not working. The IWDG is reset when exiting from Shutdown

\* If IWDG enabled



The IWDG can be active in all modes, except in Shutdown mode.

When the STM32WB microcontroller exits from Shutdown mode, the IWDG registers are set to their initial values.