



STM32F7 – Independent watchdog (IWDG)

Independent watchdog (IWDG)

Revision 1.0



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

- Serves to detect and resolve malfunctions due to software failure:
 - Triggers a system reset when the counter reaches a given timeout value
 - 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 failure. It triggers a reset sequence when the counter reaches the timeout value. 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 can only be disabled by a reset.

One of the main benefits for applications is its ability to run independently from the main clock. Also, the IWDG can be configured through hardware or software using option bytes.

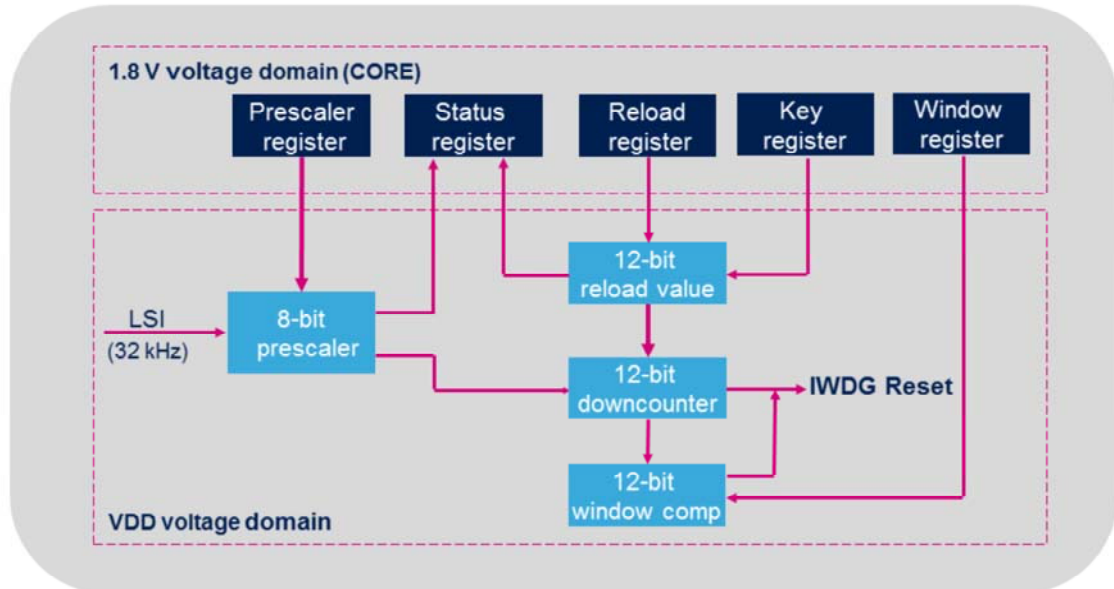
And the IWDG counter can be frozen during Standby or Stop mode.

- IWDG main features
 - 12-bit free-running downcounter
 - Clocked from an independent RC oscillator (LSI)
 - Conditional reset (if watchdog activated):
 - When the downcounter value becomes less than 0x000
 - When the downcounter is reloaded outside the window (if window option is enabled)



The IWDG features a 12-bit free-running downcounter. It is clocked by an independent low-speed internal RC oscillator at 32 kHz.

When the IWDG is activated, a reset is generated if the downcounter reaches zero or when the downcounter is reloaded outside the window (if this option is enabled).



The IWDG registers are located in the CORE voltage domain while its functions are in the VDD voltage domain. This architecture makes it possible for the IWDG to continue operating in Stop and Standby modes.

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

When the IWDG is started, the 12-bit counter starts counting down from the reset value of 0xFFF.

To refresh the IWDG counter, the Key value (0xAAAA) must be written in the Key register to reload the counter value.

If the downcounter reaches the end of the count value (0x000), a system reset is generated.

If the window option is enabled, the counter must be refreshed inside the window; otherwise, a system reset is generated.

Configuring IWDG hardware start

5

- IWDG hardware is enabled through device option bytes
- 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 window option is enabled)



The IWDG hardware is enabled by the device's option bytes. If enabled, the watchdog automatically starts at power on.

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

Configuring IWDG software start

6

- Enable IWDG by writing 0x0000 CCCC in IWDG_KR register
- Enable register access by writing 0x0000 5555 in IWDG_KR register
- Set IWDG prescaler by programming 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 IWDG_KR



The IWDG software start is configured in a few steps.

- The first step is to write the Key register with value 0x0000 CCCC which starts the watchdog.
- Then remove IWDG register protection by writing 0x0000 5555 to unlock the key.
- Set the IWDG 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 IWDG 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

7

- Setting IWDG time base:
 - IWDG time base prescaled from LSI clock (32 kHz)
 - 7 pre-dividers: 4 to 256 selectable by IWDG_PR register (and 12-bit watchdog counter reload value, RLR[11:0])
 - Setting the IWDG timeout by using the following formula:
 $t_{IWDG} (ms) = t_{LSI} (ms) \times 4 \times 2^{(IWDG_PR[2:0])} \times (IWDG_RLR[11:0] + 1)$
where $t_{LSI} (ms) = 1/32000 = 0.03125$
Min. and max. timeout values from 125 μ s to 32.8 s
- Checking IWDG reset source:
 - IWDGRSTF reset flag (in RCC_CSR register) to inform when a IWDG reset occurs (after device reset)



The IWDG time base is prescaled from the LSI 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 IWDG timeout. The IWDG time is based on the LSI period and its prescaler, as well as the selected watchdog counter reload value. Given the parameter limits, the IWDG timeout value can be between 125 μ s up to 32.8 s.

Once a reset is generated by the IWDG peripheral, the corresponding flag, IWDGRSTF, is set in the RCC_CSR register to inform the source of the reset.

Debug information 8

- IWDG counter can be stopped when the core is halted depending on the DBG_IWDG_STOP configuration bit in the DBG module.



When the microcontroller enters Debug mode (core halted), the IWDG counter either continues to work normally or stops, depending on the DBG_IWDG_STOP configuration bit in the DBG module (MCU APB1 freeze register 1).

Low-power modes

9

Mode	Description
Run	Active*
Sleep	Active*
Stop	Active*
Standby	Active*

* If IWDG enabled



The IWDG can be active in all running and low-power modes.