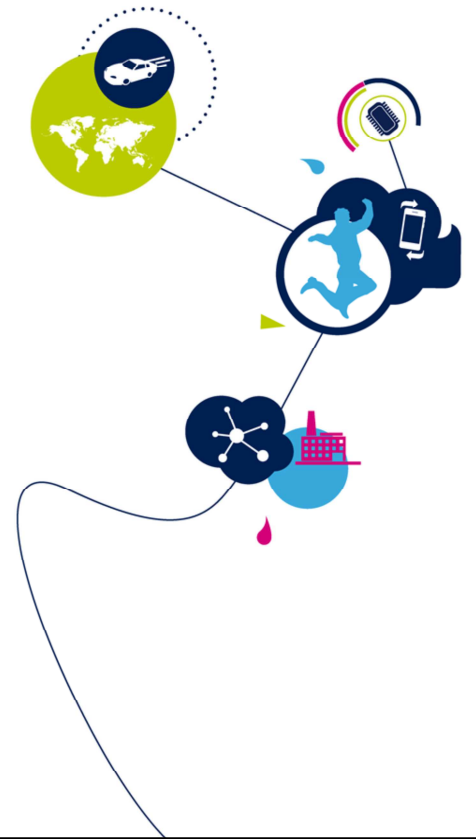


STM32MP1 – IWDG

Independent watchdog
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 hardware or software failures:
 - Generates a system reset when it is not refreshed within the expected time window
 - Always active even if the main clock fails
 - Generates an early interrupt
 - 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 hardware or software failures.

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

An early interrupt can be generated as well.

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

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

- IWDG main features
 - Programmable timeout range from 187 μ s to 131 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
 - Generates an early interrupt, at a programmable position
 - Can be frozen in Debug, Stop or Standby mode
 - Can be configured to be automatically enabled at a timeout value of 131 sec.



The independent watchdog offers a wide range of timeout values: from 187 microseconds to 131 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.

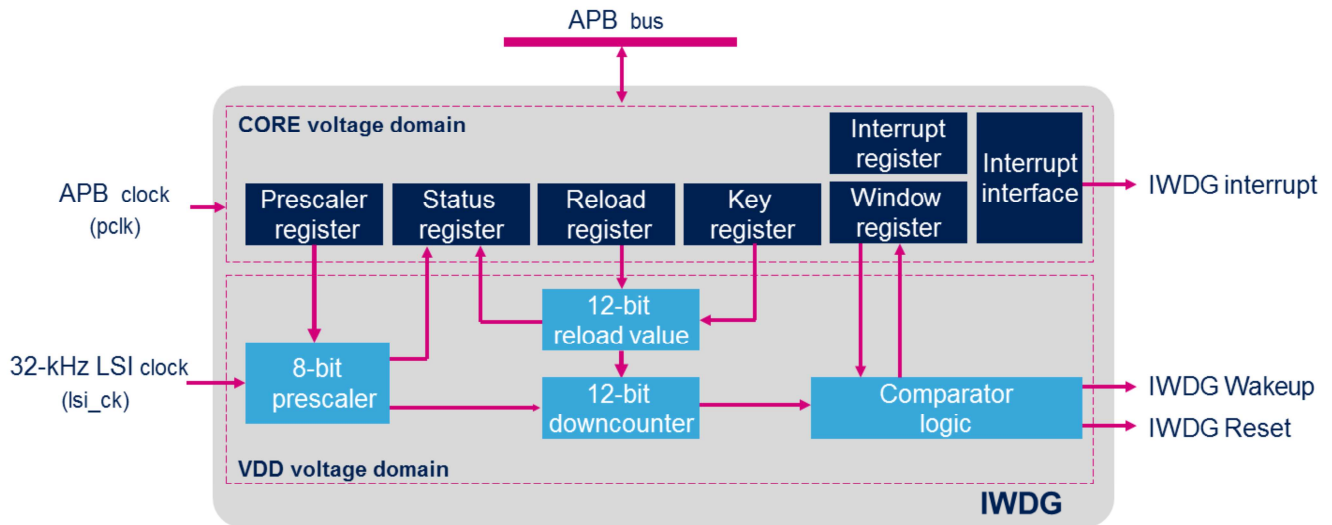
The IWDG also generates an early interrupt, at a programmable position, before the reset happens.

It is possible to enable automatically the independent watchdog after a system reset. In this case, the default timeout value is set to 131 seconds.

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.

It this way, the independent watchdog continuous counting down even if the CORE voltage is switched off.

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.

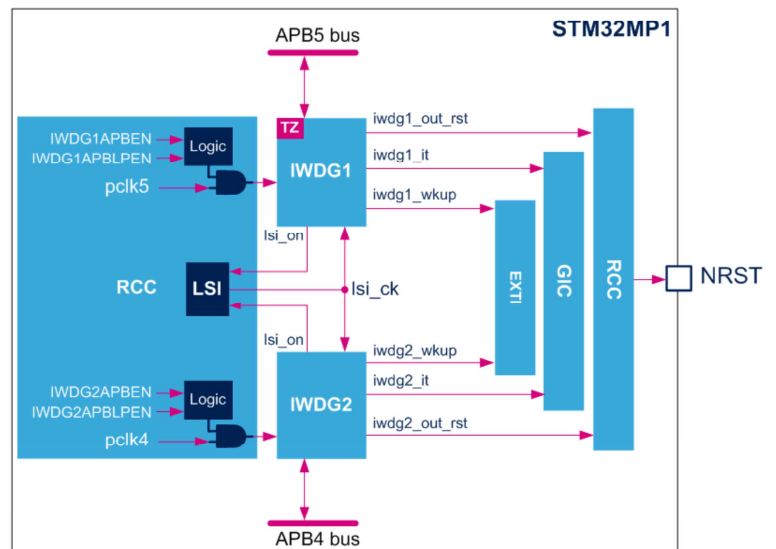
The 12-bit downcounter defines the timeout value.

The comparator logic is used for the window function and for the interrupt generation.

WatchDog Integration

5

- IWDG1 and IWDG2 are dedicated to the Cortex A7 core
- Can generate a system reset
- Can wake-up the system
- Can generate an interrupt



The STM32MP1 includes two independent watchdogs (IWDG), dedicated to the Cortex A7 core.

Both are using the low-speed internal oscillator as a watchdog clock.

When an independent watchdog is enabled, it forces the activation of the low-speed internal oscillator.

IWDG1 is TrustZone aware, and is connected to the secure APB5 bus.

The enabling of the APB clock is controlled by the RCC block. The APB clock is needed to refresh and to configure the watchdog.

The wakeup signals are connected to the EXTI block, allowing the system to exit from Stop mode when an early interrupt is generated.

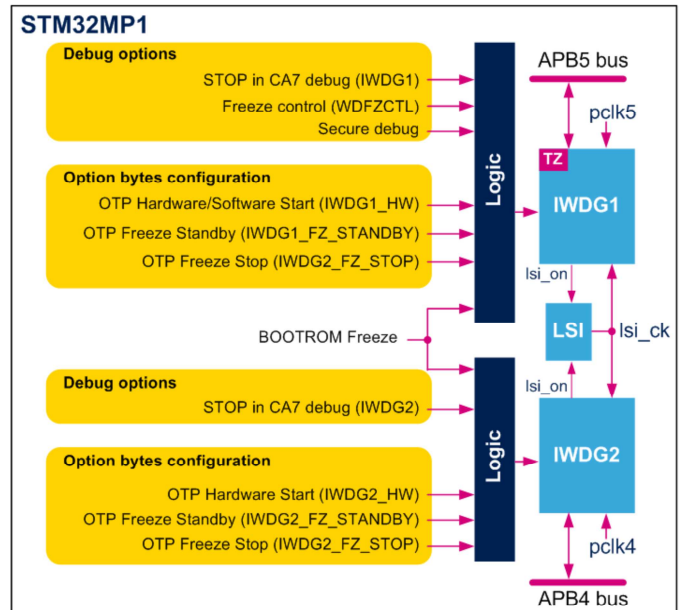
The interrupt signals are connected to the global

interrupt controller (GIC) of the Cortex A7 core.
Finally the reset signals generated by the watchdogs are connected to the RCC block, and can perform a system reset.

Behavior in Stop, Standby and Debug

6

- Possibility to freeze IWDG1 or IWDG2 when the CA7 goes to CStop.
- Possibility to freeze IWDG1 or IWDG2 when the system goes to Standby.
- Possibility to start automatically IWDG1 or IWDG2 (hardware mode).
- Possibility to freeze IWDG1 or IWDG2 in debug mode



The application can control the behavior of both watchdogs in Stop, Standby or in debug mode.

In addition, it is possible to enable automatically the watchdogs after a system reset (hardware mode).

For each independent watchdog, it is possible to select the hardware or software mode via option bytes.

For each independent watchdog, it is possible to define if it is frozen or not when the Cortex A7 goes to CStop, or when the system goes to standby.

The IWDG1 has some specific options linked to secure mode.

- When the secure debug is disabled, the IWDG1 can be frozen when both Cortex A7 cores are in debug mode.
- When the secure debug is enabled, the behavior of the IWDG1 freeze depends on WDFZCTL bit:
 - Either the IWDG1 is frozen when at least a one

- Cortex A7 cores is in debug mode,
Or the IWDG1 is frozen when both cores are in debug mode.

The IWDG2 can also be frozen when one of the Cortex A7 core is in debug mode.

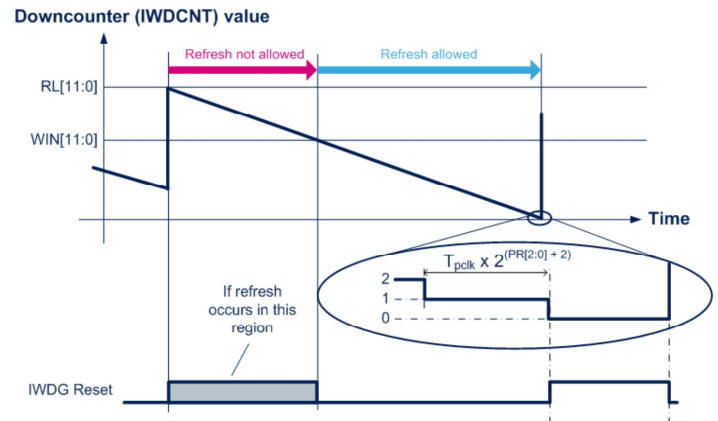
Finally when the BootRom is executed, both IWDG are frozen, preventing the BootRom to modify the watchdog settings.

IWDG Reset Generation

7

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.

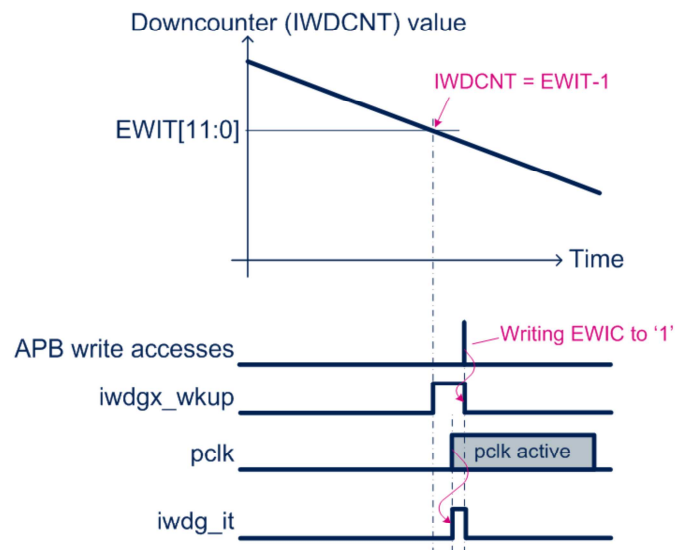
If the software refreshes the watchdog while the downcounter is greater than the value stored in the Window register, then a reset is generated as well.

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

IWDG interrupt generation

8

- The application programs the position where the interrupt shall be generated.
- When the downcounter (IWDCNT) reaches the programmed threshold, the wakeup signal is activated.
- The interrupt signal is activated when the APB clock is available.
- The interrupt is cleared by the application



As shown in the figure, the application can program at which position the interrupt shall be generated.

When the downcounter (IWDCNT) reaches the programmed threshold the wakeup signal is activated in order to wake-up the system from Stop in the case the system was in Stop mode.

The interrupt signal is activated as soon as the APB clock is available, allowing the CA7 to serve this interrupt.

At the end of the interrupt service routine, the application must clear the interrupt by writing the EWIC bit to '1'.

Configuring IWDG hardware start

9

- In IWDG hardware start, the IWDG is automatically enabled after a system reset.
- 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 0xFFFF, and starts to count down.

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.

Configuring IWDG software start 10

- 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

11

- Setting IWDG time base:
 - IWDG time base prescaled from LSI clock (32 kHz)
 - 7 pre-dividers: 4 to 1024 selectable by IWDG_PR register (and 12-bit watchdog counter reload value, RLR[11:0])
 - Setting the IWDG timeout by using the following formula:

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

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

- An IWDG reset can be identified via RCC registers



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.

The watchdog counter reload value is a 12-bit value written in the IWDG_RLR register.

The independent watchdog time is based on the LSI period and its prescaler, as well as the selected watchdog counter reload value.

It offers a wide range of timeout values.

Note that the Reset and Clock Controller (RCC) of the product provides registers giving the source of the reset. In that way the application can check if a reset is caused by an independent watchdog.

Low-power modes

12

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

* If IWDG enabled

The IWDG can be active in all modes.

**WB 12 + revoke sentences **