Hello, and welcome to this presentation of the STM32 system window watchdog. It will cover the main features of this peripheral used to detect software faults.
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.
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].
The STM32H7 includes two window watchdogs:

- WWDG1 is located in the D1 Domain and is dedicated to CPU1
- WWDG1 can generate an interrupt for the CPU1 and an interrupt for the CPU2 via the EXTI.
- WWDG2 is located in the D2 Domain and is dedicated to CPU2
- WWDG2 can generate an interrupt for the CPU2 and an interrupt for the CPU1 via the EXTI.
- Can generate a system reset

The STM32H7 includes two window watchdogs: WWDG1 and WWDG2. WWDG1 is dedicated to CPU1 usage, and the WWDG2 is dedicated to CPU2 usage. WWDG1 is connected to the APB3 bus of the D1 Domain. WWDG2 is connected to the APB1 bus of the D2 Domain. The WWDG1 early interrupt output is connected to the NVIC of the CPU1 but also to the EXTI in order to wake up and interrupt the CPU2 if the application requires such a feature. In the same way, the WWDG2 early interrupt output is connected to the NVIC of the CPU2 but also to the EXTI in order to wake up and interrupt the CPU1 if the application requires such a feature. For each window watchdog, it is possible to select if the
watchdog will freeze when the associated CPU is in Debug (core halted) mode. Please refer to the Microcontroller debug unit (DBGMCU) description for details.
The window watchdogs are frozen when the system is in Stop or Standby mode, but can remain active when the corresponding CPU is in CSleep mode. Both watchdogs have the capability to perform a system reset handled by the RCC block.
This diagram illustrates how the window watchdog operates.
When the 7-bit downcounter rolls over from 0x40 to 0x3F, it initiates a reset. This happens if the application software does not refresh the window watchdog on time. The early 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].
To enable the window watchdog clock, the corresponding window watchdog enable bit in the RCC block must be set to 1.

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

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

The downcounter uses the APB clock PCLK divided by 4096, and again divided by a division ratio selected by the application.

It can be 1, 2, 4, 8, 16, 32, 64 or 128 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 which
parts causes the reset, thanks to status flags provided by the RCC block. The window watchdogs can be one of the sources.
The Early Wakeup Interrupt 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 interrupt occurs whenever the downcounter value reaches 0x40.
It is enabled by setting the EWI bit in the WWDG_CFR register.
The EWI interrupt is cleared by writing “0” to the EWIF bit in the WWDG_SR register.
The window watchdog is active in Run and Sleep modes. It is not available in Stop or Standby modes. In Sleep mode, the window watchdog clock can be disabled by clearing the corresponding low-power enable bit located in the RCC block.