Hello and welcome to this presentation of the STM32 System Memories Protection. It will cover the different means for protecting code and/or data from external and/or internal attacks.
Software providers may need to protect their software intellectual propriety from malicious users or from intrusive attacks.

For this purpose, STM32WB microcontrollers provide several features for protecting code and/or data located in either Flash memory, SRAM2 or Backup registers. These features can prevent the reading or writing of code and/or data through the JTAG debugger, end-user code, or SRAM Trojan code.

A new protection memory feature is dedicated to the root security services (RSS) and the wireless stack running on the Cortex-M0+ core. This CPU has an exclusive access to the protected segments.
The following means are provided for code protection purposes:

- **Cortex-M0+ security**
  - Protection of upper part of Flash memory and SRAM2 for exclusive access of Cortex M0+
  - It prevents code and data access of RSS and wireless stack by user application running on Cortex-M4.
  - This protection is always active.

- **RDP: ReaDout Protection.**
  - It prevents Flash memory access through the JTAG for ALL Flash memory areas.

- **PcROP: Proprietary code readout protection.**
  - It prevents Read access of configurable Flash memory areas performed by the CPU executing malicious 3rd-party code (Trojan horse).

- **WRP: Write protection.**
  - It prevents accidental or malicious write/erase operations.
RDP, PCROP and WRP are configurable via the STM32WB option bytes.
The Cortex M0+ security features protect firmware code and data running on this core against user applications running on the Cortex M4 core. This ensures the secure execution of the root security services (RSS) and wireless stack, as well as prevents debug access. The CortexM0+ security features are always enabled and cannot be removed, even by a RDP level regression.

For a detailed description of CortexM0+ security protection features or the RSS, please refer to the dedicated modules proposed in this training.
Cortex-M0+ security protects upper parts of Flash and SRAM2 memories. Size of each areas are automatically set during wireless stack install or update.

Secure Flash Start Address (SFSA) is the lower boundary of protected Flash memory. It is aligned on 4KBytes granularity.

Secure Backup Ram Start Address (SBRSA) and Secure Non-Backup Ram Start Address (SNBRSA) are respective lower address of protected parts of SRAM2a and SRAM2b memories. Size can be set with a granularity of 1KBytes.

However settings of these option bytes are under the responsibility of RSS. They are set during a first install or an update of the wireless stack. They cannot be modified by the user.
Let’s take a closer look at the details of the readout protection feature.
The STM32WB readout protection feature offers three levels of protection for all SRAM2 and Flash memory as well as the backup registers:

- **Level 0** means “no protection”. This is the factory default. Read, Write and Erase operations are permitted in the SRAM2 and Flash memory as well as the backup registers. Option bytes can be modified.

- **Readout protection Level 1**
  - If the selected boot mode is user Flash memory (Boot0 = 0), and if no debugger access is detected (no JTAG):
    - All operations (R/W/Erase) are permitted on Flash memory, SRAM2, and Backup registers. Option bytes can be modified.
  - If the selected boot mode is not user Flash memory (Boot0 = 1), or if a debugger access is detected (JTAG):
    - All operations (R/W/Erase) to Flash memory, SRAM2, and Backup registers are blocked (hard fault generated). Option bytes can be modified.
execution until the next power-on reset. Please note that option bytes can still be modified in Level 1.
Level 2 provides the same protection features for the SRAM2, Flash memory and Backup registers as described for Level 1. However, there are three major differences.

1. The JTAG/SWD debugger connection is disabled (even at the ST factory, to ensure that there are no backdoors);
2. the Boot mode is forced to User Flash memory REGARDLESS of what the boot 0/1 settings are, and Level 2 is permanent. Once set to Level 2, there is no going back;
3. RDP/WRP option bytes can no longer be changed, as well as ALL the other option bytes.
Changing the level of RDP protection is only permitted when the current protection level is ‘1’. RDP level 2 is permanent.
Changing the RDP protection level from ‘1’ to ‘0’ will automatically erase the non-secure part of the Flash memory, SRAM2 and backup registers. The secure part of the Flash is not impacted and the security remains unchanged.
Let’s look at the transitions possible between each readout protection level.
As already mentioned, STM32WB MCUs have three RDP levels:
1. Level 0 means there is no memory protection and option bytes can be modified.
   From Level 0, the device can move to Level 1 or Level 2.
2. Level 1 ensures the memory protection while keeping debug access enabled.
   From Level 1, the device can move to Level 0 or Level 2. Regression to Level 0 will cause a Flash memory Mass Erase.
3. Level 2 ensures the memory protection the same as Level 1, but completely disables JTAG/SWD debug access.
   Level 2 is a permanent state, and moving to another
RDP level is not possible.
This table summarizes the different types of access authorized for the Flash memory, backup registers and SRAM2 according to the readout protection (RDP) level, configured boot mode and debug access, as previously discussed. In summary:

- When RDP is set to Level 0, no protection mechanism is active and all memories can be read and modified.
- When RDP is other than level 0:

  If the device is configured to boot from the User Flash memory, THEN:
  => The User Flash memory, backup registers and SRAM2 can be read or modified regardless of the RDP level.
  => The System Flash memory can be read only.
  => The Option bytes can be read only when the RDP is set to Level 2.
Otherwise, if the device is not configured to boot from the User Flash memory or if a debugger access is detected, THEN:
=> Almost all memories are not accessible excepted the System Flash memory, which can only be read in Level 1, and Option bytes which can be read or modified in Level 1.
Proprietary Code Readout Protection (PCROP)

Let’s take a closer look at the details of the Proprietary Code Readout Protection (PCROP) and how it’s different from RDP.
PcROP means: Proprietary code readout protection

Why PcROP?

Proprietary code readout protection is basically a way to protect the confidentiality of 3rd-party software intellectual property code independently of the RDP level setting. Third-parties may develop and sell specific software IPs for STM32 microcontrollers and original equipment manufacturers may use them when developing their own application code. Proprietary code readout protection helps protect the confidentiality of 3rd-party IPs and protects software intellectual property against malicious users.

In other words, PcROP consists in preventing malicious software or debuggers from reading sensitive code. The protected area is execute-only and can only be reached by the STM32 CPU, as an instruction code, while all other accesses (DMA, debug and CPU data...
read, write and erase) are strictly prohibited. This means that the code to be protected must be compiled using a specific compiler option: For example: “–execute_only” (for Keil tools)
The proprietary code readout protected areas in Flash memory is defined through the option bytes.

The PcROP feature is improved on the STM32WB devices. Two separate PcROP areas can now be set independently (one per bank), each one defined by a start and end address with a granularity of 64 bits. Note that once a PcROP area is configured, its size can only be increased.

Once the PcROP areas have been defined, the only way to disable this protection feature is to change the RDP protection level from '1' to '0', which erases the entire Flash memory.

The Erase policy of PCROP areas in case of RDP level regression is defined through PCROP_RDP option bit. By setting the PCROP_RDP bit in the option bytes, the code in the PcROP areas will NOT be lost and the
protection will not be removed.

To further explain the ‘execute only’ meaning of the PcROP:
• The PcROP is a sub-state of the RDP. The PcROP is designed to prohibit other code executing on the STM32 from reading the Flash memory. This is not the same as the RDP, where the protection targets external worlds. When the PcROP is enabled, the AHB only allows the Instruction bus to work, so code can only be executed. The Data bus can’t access that Flash memory.
• Once the development phase is completed, the PCROP can then be turned into a RDP setting, Level 1. In this case, the external world is limited to read-only. But the PcROP settings for specific sectors stills applies to all masters trying to read that code.
Now, let's take a closer look at the details of the write protection settings of the STM32WB.
The Flash memory write protection mechanism is designed to prevent unwanted write access to defined areas in Flash memory, such as the bootloader or calibration constants that do not change. The write protection areas are defined through the option bytes. The user can define up to four different write-protected Flash memory areas independently (two per bank). Each of the four Flash memory areas are defined by a start and end address with a page granularity (4 Kbytes).

The size of the write areas can be modified whenever the RDP level is not set to Level 2. Erase operations are treated as write operations on write protected areas, meaning they are not allowed.

**Flash write protection**

- **Settings & constraints**
  - The write protected area is defined through the option bytes
  - The STM32WB allows 2 WRP areas to be configured with page granularity (4 Kbytes)
  - The WRP area size can be modified (option byte changed) whenever the RDP is not Level 2.

- **Properties**
  - When a WRP area is defined/enabled, write/erase operations are not permitted on this area.
In addition to this training, you may find these three modules useful.

- Refer to these trainings linked to this feature:
  - STM32WB-Memory-Flash
  - Flash memory architecture
  - STM32WB-System-CM0+ security
  - Description and configuration of the Cortex-M0+ feature
  - STM32WB Security Root Security Services (RSS)
  - Description of RSS functionalities (Wireless stack install/update & CK3)