



مرکز آموزش نیرا سیستم

nirasystem.com

FLASH

Ali Mirghasemi



Introduction

- The FLASH peripheral on the STM32F4 microcontroller series is a crucial component responsible for non-volatile storage of program code and data.
- It is a type of embedded Flash memory that allows developers to store and retrieve information even when the power is turned off.
- This is particularly essential for microcontrollers, as it enables them to retain their firmware and configuration settings.



Application

- The primary application of the FLASH peripheral is to store the program code that the microcontroller runs.
- This includes the firmware written by the developer, which contains instructions for the microcontroller to execute.
- Additionally, the FLASH memory may be used for storing configuration settings, calibration values, or any other data that needs to persist across power cycles.



Registers

- **Flash Control Register (FLASH_CR)**
 - This register controls various aspects of the Flash memory, such as enabling or disabling the Flash, initiating the programming or erasing operations, and configuring the size of the data width.
- **Flash Status Register (FLASH_SR)**
 - This register provides information about the status of ongoing Flash operations, indicating whether a write or erase operation is complete.
- **Flash Option Control Register (FLASH_OPTCR)**
 - This register is used to configure various Flash options, such as read protection, write protection, and other specific features.



Write Cycle

- **Unlocking the Flash**
 - Before any write operation, the Flash memory needs to be unlocked using the `HAL_FLASH_Unlock()` function.
- **Erasing the Sector**
 - If needed, the Flash sector must be erased before writing new data. This is done using the `HAL_FLASHEx_Erase()` function.
- **Clear Data Cache and Instruction Cache**
 - If an erase operation in Flash memory also concerns data in the data or instruction cache, you have to make sure that these data are rewritten before they are accessed during code execution
- **Programming the Data**
 - The actual data is then programmed into the Flash memory using the `HAL_FLASH_Program()` function.
- **Locking the Flash**
 - Once the write cycle is complete, it's important to lock the Flash memory again to prevent accidental writes. This is done using the `HAL_FLASH_Lock()` function.