# 1.Autosar Memory Stack

- Lưu vehicle information, calibration data

- Các module trong memstack:

A screenshot of a computer

Description automatically generated

A white paper with black text

Description automatically generated

## Memory requirement:

* Application có thể access nhiều kiểu non-voltatile memory khác nhau như external flas, eeprom, internal fls
* Sau khi ECU reprogramming, các dữ liệu cố định or thống số xe nên được giữ an toàn
* 1 ECU có thể sử dụng cho nhiều variant khác nhau của xe phụ thuộc vào tính nắng hoặc là thị trường (MAP)
* Memory stack nên có chức năng kiểm tra tính nhất quán và toàn vẹn của data được lưu trong memory lúc hoạt động
* …….

A close up of a document

Description automatically generated

## NvRAM Manager

### Overview

- Trong AUTOSAR, NvRAM Manager (NvM) xử lý toàn NonVolatile Memory của ECU. SWC chỉ tương tác với NvM để read hoặc write NonVolatile Memory

- NvM độc lập với phần cứng bên dưới. Nó chỉ xử lý NonVolatile Memory thông quá khối bộ nhớ trừu tượng (Fee, Ea). Mỗi khối có một số xử lý (Block ID) và administrative data

- Application không thể tự truy NonVolatile Memory. Nó cần NvM để cung cấp tất cả các giao diện cần thiết. Ngay cả với các giao diện đó, **Application chỉ có thể tương tác với bản sao RAM của khối**. NvM thực hiện kiểm tra tính nhất quán và đồng bộ hóa dữ liệu giữa RAM và bộ nhớ NV.

NvM tương tác với các mô-đun AUTOSAR sau:  
**EcuC** — để khởi tạo, khởi động và tắt  
**SchM** — để lập lịch trong quá trình hoạt động bình thường của ECU  
**MemIf** — để truy cập các thiết bị bộ nhớ trừu tượng  
**Dem** — để ghi nhật ký lỗi liên quan đến hoạt động của NvM  
**Crc** — để kiểm tra tính toàn vẹn và tính nhất quán trên dữ liệu  
Ngoài các mô-đun này, NvM cũng tương tác với các mô-đun như BswM, Det và Csm. Tất cả các SWC (ứng dụng) sử dụng bộ nhớ không biến động đều tương tác với NvM thông qua RTE.

A diagram of a network

Description automatically generated

### Feature

A screenshot of a computer

Description automatically generated

- NvRAM Block: là entite structure (toàn bộ khối cấu trúc) cần dùng để quản trị và lưu dữ NV data. NvRAM block bao gồm các basic storage object

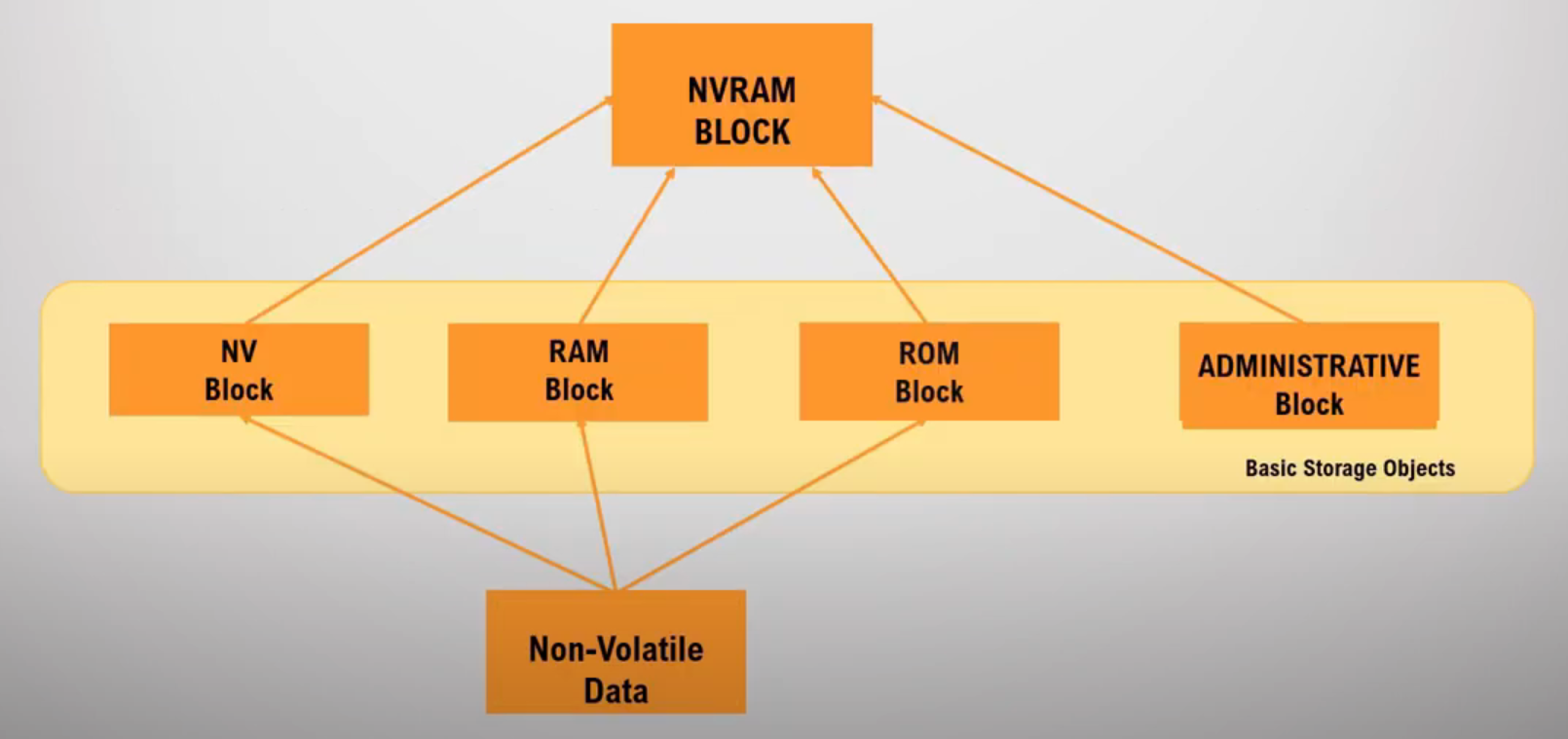
- NV data : data được lưu trong non-volatile memory

- Basic storage object là đơn vị nhỏ nhất của NvRAM block, 1 vài basic storage object tạo nên NvRAM block. Basic storage object có thể được cư trú ờ các vùng memory khác nhau ( RAM, ROM, NV memory)

### Basic storage Object

A white background with black text

Description automatically generated



#### NV block

- 1 phần **bắt buộc** của NvRAM block được **đặt trong Non-Volatile Memory**. Bao gồm non-volatile data và optionally (CRC, Nv block header)

A yellow rectangular sign with black text

Description automatically generated

#### Ram block ( 7.1.3.2 autosar spec)

* RAM block is a basic storage object nằm trong RAM – not mandatory
* consisting of user data and (optionally) a CRC value and (optionally) a NV block header.c

A diagram of a block diagram

Description automatically generated

* 2 kiểu:
* Permanent Ram block: belongs to NV block, có thể được access bởi 1 SWC. Address của permanent ram block được fixed trong NVM configuration
* Temporary ram block :
  + Address sẽ không được biết trong lúc configuration, sẽ được truyền vào NVM module trong runtime.
  + Nếu nhiều SWCs muốn access vào cùng **1 NV block**, thì mỗi SWC sẽ sử dụng its own temporary ram block

#### Rom block

* Lưu trọng ROM
* Lưu các data không thể (ko nên) bị modified trong run-time (default data, calibration data)
* Sử dụng để cung cấp các default data khi data không được đọc từ NV memory
* Không quan tâm tới CRC, content của block là constant.
* A square of a blue color

  Description automatically generated with medium confidence

A close up of a paper

Description automatically generated

Vd: Select assist map:

* Sau khi ECU khởi động sẽ đọc giá trị MAP selection từ NV Block ( data flash)
* Nếu không đọc được, hoặc đọc giá trị ko đúng thì default MAP sẽ được sử dụng. Default map chính là calib data được lưu trong ROM block.

#### Administrative block

- mandatory part của NV block

- lưu trong RAM

- lưu các attribute/ error/ status information của NVRAM

# CHATGPT - Which different from ROM and Non volatile memory

ROM (Read-Only Memory) and non-volatile memory (NVM) are both types of memory that retain data even when the power is turned off. However, they differ significantly in terms of their characteristics, usage, and how they are programmed.

**ROM (Read-Only Memory)**

1. **Characteristics:**
   * **Non-rewriteable:** ROM is typically non-rewriteable, meaning once data is written to ROM, it cannot be easily modified or erased.
   * **Permanent Storage:** It is used primarily for storing firmware, bootloader, or other essential programs that need to be preserved across power cycles.
   * **Fast Access:** ROM generally provides fast access times since it is designed for frequent read operations.
2. **Types of ROM:**
   * **Mask ROM:** Manufactured with data already programmed during fabrication.
   * **EPROM (Erasable Programmable ROM):** Can be erased and reprogrammed using UV light exposure.
   * **EEPROM (Electrically Erasable Programmable ROM):** Can be electrically erased and reprogrammed in-circuit, providing more flexibility compared to EPROM.
3. **Applications:**
   * **Embedded Systems:** Used in microcontrollers, CPUs, and other embedded systems to store firmware and critical instructions.
   * **Bootloaders:** Often used to store initial boot code that initializes the system upon power-up.
   * **Read-Only Data:** Suitable for storing data that does not change frequently and needs to be reliably available.

**Non-Volatile Memory (NVM)**

1. **Characteristics:**
   * **Rewriteable:** NVM allows data to be written, modified, and erased multiple times throughout its lifespan.
   * **Various Technologies:** Includes flash memory (NAND, NOR), EEPROM, Ferroelectric RAM (FeRAM), and others, each with different properties like speed, endurance, and cost.
   * **Slower Access:** Typically slower access times compared to ROM due to programming and erase operations.
2. **Applications:**
   * **Data Storage:** Used for storing application data, configurations, user preferences, and other dynamic information that may change over time.
   * **File Systems:** Integral part of file systems in computers and embedded systems where data needs to persist across power cycles.
   * **Critical Updates:** NVM allows firmware updates and system configuration changes without requiring physical replacement of memory chips.

**Key Differences**

* **Programmability:** ROM is typically programmed during manufacturing and is non-rewriteable or has limited reprogramming capabilities (e.g., EPROM). In contrast, NVM is designed for frequent read/write operations and allows data to be modified in-circuit.
* **Use Cases:** ROM is used for permanent storage of firmware and essential system instructions, while NVM is used for dynamic data storage and allows for flexibility in storing and updating application data.
* **Access Speed:** ROM generally provides faster access times compared to NVM, which may have slower access times due to the nature of programming and erasing operations.
* **Cost and Complexity:** ROM tends to be simpler and cheaper to manufacture for large-scale production of fixed data. NVM technologies vary in cost and complexity, offering different trade-offs between speed, endurance, and cost.

In summary, while both ROM and NVM are types of non-volatile memory used in electronics, they serve different purposes and are suited for different types of data storage and access requirements in computing and embedded systems.