Memory Types in Embedded Systems

Flash | SRAM | EEPROM

Memory is the heart of any embedded system. Each type plays a different role in how data is processed, stored, and preserved. Let’s explore these memory types, their characteristics, and how they shape embedded device behavior.

Sanath Thilakarathna
Mechatronics Engineer | Lecturer | Researcher | Roboticist

Why Memory Matters ?

In embedded systems, memory isn't just about storage capacity, it's about making the system responsive, efficient, and durable.

- Faster memory means quicker execution of tasks and lower latency.
- Non-volatile memory ensures your device retains critical data after shutdown.
- Energy efficient memory extends battery life.
- The right memory balance reduces cost without compromising reliability. The system's responsiveness, behavior on power loss, and long term operation all depend on how memory is chosen and used.

Sanath Thilakarathna
Mechatronics Engineer | Lecturer | Researcher | Roboticist

Flash Memory

Flash memory is a type of non-volatile memory primarily used to store firmware, the core software that controls hardware functions.

- It retains information even after power is removed, making it ideal for program storage.
- Flash is typically organized in blocks and sectors, and erasing/writing happens in blocks, not bytes.
- It supports tens of thousands of write/erase cycles, but excessive writing shortens lifespan.
- Internal Flash is found in most microcontrollers (e.g., STM32, ESP32, ATmega328P) and can range from kilobytes to several megabytes.
- Used to store bootloaders, application code, and sometimes constants like lookup tables or fonts.

Sanath Thilakarathna
Mechatronics Engineer | Lecturer | Researcher | Roboticist

SRAM (Static RAM)

SRAM is the fastest memory in a microcontroller and is used for real-time data storage during program execution.

- Unlike Flash, it is volatile, all data is lost when power is turned off.
- It does not require refreshing (unlike DRAM), which gives it lower latency and higher performance.
- SRAM is used to store stack data (function calls, return addresses), heap (dynamically allocated memory), and global/static variables.
- It consumes more power per bit and occupies more silicon area, so MCUs typically have less SRAM than Flash.
- Critical for applications requiring fast response like signal processing, motor control, and data buffering.

Sanath Thilakarathna
Mechatronics Engineer | Lecturer | Researcher | Roboticist

EEPROM

EEPROM (Electrically Erasable Programmable Read-Only Memory) is designed for small, non-volatile data storage.

- Unlike Flash, EEPROM can be written and erased one byte at a time, making it flexible for storing small data units.
- It's perfect for storing parameters that might change over time, such as device calibration values, configuration settings, or user preferences.
- EEPROM is slower than SRAM and less dense than Flash, but it supports a higher number of write cycles (up to 1 million).
- Found in many 8-bit MCUs (e.g., ATmega series), and also available as external I2C/SPI memory chips.
- Not suited for fast or large-volume data operations but invaluable for persistent settings.

Sanath Thilakarathna
Mechatronics Engineer | Lecturer | Researcher | Roboticist

Comparing Flash, SRAM, and EEPROM

Each memory type serves a specific function in an embedded system. Understanding their differences is essential for efficient design:

- **Flash:** Non-volatile, block-based, used for firmware and static data.
- **SRAM:** Volatile, fast access, used for runtime data and temporary buffers.
- **EEPROM:** Non-volatile, byte-addressable, for configuration and calibration. Trade-offs exist in terms of endurance, speed, size, and cost. Good embedded design balances these factors based on application needs.

Sanath Thilakarathna
Mechatronics Engineer | Lecturer | Researcher | Roboticist

MCU Memory Map Example

Let’s look at ATmega328P (used in Arduino UNO) as an example:

- **Flash (32KB):** Stores your compiled firmware and bootloader.
- **SRAM (2KB):** Executes logic, handles variables, buffers.
- **EEPROM (1KB):** Stores saved values like high scores, calibration offsets, or settings. This simple memory hierarchy is replicated across almost all microcontrollers, regardless of size or complexity.

Sanath Thilakarathna
Mechatronics Engineer | Lecturer | Researcher | Roboticist

Design Insights

Efficient embedded design depends on smart memory allocation:

- Flash is great for data that changes rarely but must persist.
- SRAM is limited and fast, use it for time-critical variables and buffers.
- EEPROM is perfect for persistent data that may change occasionally.
- Monitoring stack/heap usage in SRAM is crucial to avoid overflows.
- Reducing unnecessary writes to Flash or EEPROM increases system longevity. Optimizing memory usage can lead to better power savings, faster response times, and more reliable devices.

Sanath Thilakarathna
Mechatronics Engineer | Lecturer | Researcher | Roboticist

Follow for More

Interested in microcontrollers, embedded design, or firmware engineering?

- Follow me for real-world insights, system-level design breakdowns, and technical deep dives.
- Let’s share knowledge and build better embedded systems together!

#Flash #SRAM #EEPROM #IoT #SystemDesign #EngineeringTips

Sanath Thilakarathna
Mechatronics Engineer | Lecturer | Researcher | Roboticist