

LD Linker scripts

The linker scripts in `esp-open-sdk/sdk/ld/` dictate how the GCC linker organizes code sections (`.text`, `.data`, `.bss`, `.irom0.text`) and maps them to physical flash offsets and memory addresses.

Understanding the Naming Conventions

- **v6**: Refers to the version of Espressif's SDK linker layout definitions.
 - **new vs. old**: Refers to the Espressif bootloader structure version:
 - **old (v1.1 bootloader era)**: Used legacy flash mapping rules (limited `irom0` size).
 - **new (v1.2+ bootloader era)**: Expanded the mapped `irom0` size per application slot to support larger code bases.
 - **512 / 1024 / 2048**: Refers to the target flash size in **Kilobytes** (512 KB, 1 MB, 2 MB).
 - **app1 vs. app2**: Used in **OTA (Over-The-Air)** two-slot layouts (`user1.bin` and `user2.bin`). `app1` links code to execute from the first flash partition, while `app2` links code to execute from the second partition.
-

Breakdown of the Linker Scripts

1. Non-OTA Script

- **eagle.app.v6.ld**: Purpose: ** Non-OTA (No-Bootloader) applications. Maps `eagle.flash.bin` directly to offset `0x00000` and `eagle.irom0text.bin` to offset `0x10000` or `0x20000`. It assumes the internal hardware ROM boots the application directly without checking an OTA header or selecting between slots.

2. OTA Scripts (Bootloader-Based)

- **eagle.app.v6.new.512.app1.ld / app2.ld** : Purpose: 512 KB Flash using the “new” bootloader format. Split into two equal OTA partition slots (~236 KB max code size per app).
- **eagle.app.v6.new.1024.app1.ld / app2.ld** : Purpose: 1 MB (and larger) Flash using the “new” bootloader format. Split into two ~492 KB OTA slots (`user1.bin` / `user2.bin`).
- **eagle.app.v6.new.2048.ld**: Purpose: Single large application layout for 2 MB+ flash modules designed to work alongside special bootloaders or expanded static allocations.
- **eagle.app.v6.old.*.ld**: Purpose: Legacy backward-compatibility layouts for projects targeted at very early Espressif bootloader releases.

3. Core ROM Address Symbols

- **eagle.rom.addr.v6.ld**: Purpose: Hardware ROM symbol map. Behavior: Does not define

flash partitions. Instead, it provides the physical memory addresses of functions pre-baked into the ESP8266 internal hardware ROM (such as axtls crypto functions, low-level SPI functions, printf, and malloc). This file is included by other linker scripts.

From:

<https://wiki.oscardegroot.nl/> - HomeWiki

Permanent link:

<https://wiki.oscardegroot.nl/doku.php?id=esp:esp8266:open-sdk:firmware-ld-linking>

Last update: **2026/09/21 15:38**

