Offset address: 0x8000 trong bộ nhớ flash.

Text

Description automatically generated with medium confidence

Otadata slot chứa dữ liệu cho việc OTA update. Bootloader kiểm tra otadata để biết app nào được chạy. nếu otadata trống, bootloader chạy factory app.

Type field

Partiton type field được xác định: app (0x00) và data (0x01). ESP IDF bootloader bỏ qua các partition type ngoài app (0x00) và data(0x01).

SubType

Khi type là **app**, subtype field có thể được xác định là factory (0x00), ota\_0 (0x10), … ota\_15(0x1F) hoặc test (0x20).

* factory (0x00) là app partition mặc định. Bootloader sẽ chạy factory app trừ khi có một partition otadata, trong trường hợp này thì bootloader đọc otadata để xác định ota nào được chạy. OTA không được update vào factory partition.
* ota\_0 (0x10) … ota\_15 (0x1F) là các OTA app slot. Khi OTA chạy, otadata cấu hình xem app slot nào được bootloader nên chạy. Khi sử dụng OTA, một chương trình cần ít nhất 2 OTA slot (ota\_0 và ota\_1).
* test (0x20): test firmware.

Khi type là **data**, subtype field có thể được xác định là ota (0x00), phy (0x01), nvs (0x02), nvs\_key (0x04) …

* ota là otadata chứa thông tin về OTA app slot được chọn hiện tại. ota nên có kích thước 0x2000 byte

Over the air (OTA)

Một chương trình app nên có factory app và ít nhất 2 ota app (ota\_0 và ota\_1)

Hàm OTA hoạt động bằng cách viết một firmware app mới vào bất kỳ OTA app slot nào hiện tại đang không được chọn để boot. Một khi firmware app được xác nhận, OTA Data partition (otadata) được update để chỉ định firmware mới nên được dùng trong lần boot tiếp theo.

Khi khởi động ban đầu, OTA data partition nên không chứa dữ liệu gì (tất cả các byte được xóa thành 0xFF). Trong trường hợp đó, esp-idf software sẽ chạy factory app nếu nó nằm trong partition table. Nếu không có factory app (do người dùng không cài) thì OTA slot đầu tiên sẽ được chạy.

Sau khi OTA firmware được update, OTA data partition được update để xác định OTA app slot nào được chạy lần tới.

App rollback

Mục đích chính của application rollback là giữ cho thiết bị được chạy sau khi update. Tính năng này cho phép bạn quay trở lại application trước trong trường hợp application mới bị lỗi. Khi quá trình rollback được cho phép và một OTA update cung cấp một version app mới, 1 trong 3 thứ có thể xảy ra:

* Nếu application mới hoạt động tốt, hàm *esp\_ota\_mark\_app\_valid\_cancel\_rollback()* đánh dấu app đó là ESP\_OTA\_IMG\_VALID. Sau đó sẽ không có hạn chế trong việc chạy application đó.
* Nếu application mới không hoạt động được do có lỗi, một quá trình rollback về application trước được yêu cầu, hàm *esp\_ota\_mark\_app\_invalid\_rollback\_and\_reboot()* đánh dấu app lỗi đó là ESP\_OTA\_IMG\_INVALID và khởi động lại. Nhờ đó, bootloader sẽ không chọn app lỗi đó để chạy và chạy chương trình trước.
* Nếu CONFIG\_BOOTLOADER\_APP\_ROLLBACK\_ENABLE được set, và một reset xảy ra mà không gọi 1 trong 2 hàm thì sau đó application được rollback.

OTA có 2 cách:

* Sử dụng APIs được cung cấp bởi app\_update
* Sử dụng APIs được cung cấp bởi esp\_https\_ota, OTA qua HTTPS.