

Utilizzo di **bsdiff** e **bspatch** su macOS per aggiornamenti firmware differenziali

Introduzione

bsdiff è un tool open source che consente di creare file **delta patch** binari, ovvero file contenenti solo le differenze tra due versioni di un file binario (es. firmware). **bspatch** è il tool complementare che applica tali patch per rigenerare il nuovo file a partire da quello vecchio.

Utile nei sistemi embedded a risorse limitate dove è necessario trasmettere aggiornamenti firmware **riducendo la banda** e la **memoria necessaria**.

Installazione su macOS

Metodo consigliato: Homebrew

Assicurati di avere [Homebrew](#) installato. Poi esegui:

```
brew install bsdiff
```

Verifica installazione:

```
which bsdiff  
which bspatch
```

Struttura di utilizzo

Generare una patch:

```
bsdiff old_firmware.bin new_firmware.bin patch_file.delta
```

Applicare una patch:

```
bspatch old_firmware.bin new_output.bin patch_file.delta
```

Esempio pratico

Supponiamo di avere due versioni firmware:

```
ls  
firmware_v1.0.bin  
firmware_v1.1.bin
```

Generazione patch:

```
bsdiff firmware_v1.0.bin firmware_v1.1.bin firmware_update.delta
```

Applicazione patch:

```
bspatch firmware_v1.0.bin firmware_reconstructed.bin  
firmware_update.delta
```

Verifica differenze:

```
cmp firmware_v1.1.bin firmware_reconstructed.bin
```

Oppure:

```
diff <(xxd firmware_v1.1.bin) <(xxd firmware_reconstructed.bin)
```



Verifica integrità della patch

Per garantire che la patch sia corretta:

```
sha256sum firmware_v1.1.bin  
sha256sum firmware_reconstructed.bin
```

Gli hash devono coincidere.



Integrazione con MCU embedded (Nordic nRF5)

Passaggi suggeriti:

1. **Genera la patch** con `bsdiff` su macOS.
2. **Trasmetti la patch** via BLE, UART, etc., al tuo dispositivo.
3. **Integra bspatch** (versione minimale) nel tuo firmware:
 - Porting di `bspatch.c` nel tuo progetto.
 - Gestione diretta della flash (lettura/scrittura).
 - Buffering ottimizzato per risparmio RAM.
4. **Verifica CRC finale** del firmware aggiornato prima del boot.

⚠ `bspatch` usa bzip2. Se il target ha RAM limitata, considera versioni modificate (senza compressione o con zlib/miniz).

⚡ Prestazioni e ottimizzazione

- Le patch sono **estremamente leggere** se le modifiche al firmware sono localizzate.
- Un tipico `bsdiff` su firmware da 256 KB può generare una patch da **2–10 KB**.
- Tempo di patching su PC: **<1 secondo**.
- Tempo su MCU: **dipende dalla flash e RAM**, tipicamente 1–3 secondi.

🔗 Riferimenti

- [bsdiff originale](#)
- [bsdiff su GitHub \(fork\)](#)
- Tool alternativi: xdelta3, zsync, Memfault reflasher