Secured Bootloader: program pentru controlul secvenței de inițializare și al actualizării aplicației în microcontrollere programabile

Conținut

[Obiectiv 3](#__RefHeading___Toc953_136633088)

[Structura proiectului 3](#__RefHeading___Toc955_136633088)

[Workflow 3](#__RefHeading___Toc957_136633088)

[Bundler 3](#__RefHeading___Toc959_136633088)

[Flasher 4](#__RefHeading___Toc961_136633088)

[Bootloader 4](#__RefHeading___Toc963_136633088)

## Obiectiv

Obiectivul proiectului este de a permite dezvoltatorilor de software embedded să delege sarcina programării unui microcontroller clienților în vederea ușurării procesului de actualizare al firmware-ului unui dispozitiv. Acest lucru trebuie să poată fi făcut intr-un mod sigur, deoarece clientul nu ar trebui să aibă access la codul sursă al aplicației care se încarcă pe dispozitiv. De asemenea procesul trebuie să poată fi realizat de o persoană nespecializată, așadar succesiunea de pași pentru a atinge obiectivul trebuie să fie simplă și prietenoasă.

## Structura proiectului

În cadrul lucrării de diploma se vor dezvolta trei mari componente:

1. Bootloader-ul preîncărcat pe microcontroller (denumit simplu: *bootloader*). Aplicație embedded.
2. Aplicația de programare a microcontroller-ului (denumită simplu: *flasher*). Aplicație desktop.
3. Aplicația de împachetare a firmware-ului care urmează a fi programat în microcontroller (denumită simplu: *bundler*). Aplicație desktop.

## Workflow

În cazul în care un dezvoltator de software embedded dorește să ofere clienților un update al firmware-ului unui dispozitiv, iar acest dispozitiv a fost pre-programat de care dezvoltator cu bootloader-ul dezvoltat în cadrul acestei lucrări, atunci clientul poate folosi flasher-ul ca să programeze noul firmware pe microcontroller, unde bootloader-ul va decripta bundle-ul și îl va scrie (și îi verifica integritatea) într-o zonă de memorie nevolatilă în vederea execuției ulterioare a codului.

### Bundler

În momentul în care decizia de a face o actualizare a unui firmware este luată de către dezvoltator acesta va trebui să creeze, cu ajutorul aplicației bundler, un bundle. Bundle-ul este pachetul care va ajunge la client. Acest bundle reprezintă de fapt codul binar al aplicației ce va urma să fie executată de către microcontroller, cu diferența că bundle-ul este criptat, deoarece clientul nu trebuie să poată vedea conținutul pachetului, acesta conținând informații confidențiale, proprietatea dezvoltatorului.

Așadar bundler-ul, va avea ca input:

* cheia de criptare
* un fișier conținând codul binar al firmware-ului

Iar ca output unul singur:

* bundle-ul destinat clientului

### Flasher

Clientul va primi un bundle de la dezvoltator pe care va trebui să îl încarce în aplicația flasher.

Această aplicație se va ocupa de toate detaliile necesare trimiterii cu succes a bundle-ului spre bootloader (eg. împărțirea în pachete, managementul erorilor de comunicație). Flasher-ul va comunica pe portul serial cu dispozitivul. Așadar dispozitivul trebuie in prealabil conectat la calculator prin portul serial. În urma procesului de flash, aplicația va raporta clientului statusul operației, fie ca programarea a reușit, fie motivele pentru care a eșuat (eg. bundle-ul nu a fost valid, decriptarea nu s-a putut efectua, etc).

Opțional, clientul va putea verifica validitatea bundle-ului primit prin intermediul unei valori hash, pusă la dispoziție chiar de dezvoltator.

Așadar flasher-ul va primi ca input:

* bundle-ul oferit de dezvoltator
* (opțional) o valoare hash, destinată verificării unui bundle corect transmis

Iar ca output:

* un status care reflectă rezultatul procesului de programare în microcontroller a firmware-ului

### Bootloader

Aplicația bootloader este cea mai importantă componentă a proiectului. În primul rând aceasta trebuie programată (prin metodele clasice, eg. JTAG) înaintea livrării dispozitivului către client, deoarece ea este responsabilă cu decriptarea și verificarea integrității bundle-ului primit de la flasher.

Când clientul dorește să înceapă secvența de actualizare a firmware-ului unui dispozitiv acesta va trebui să efectueze o serie de pași specificați în documentația dispozitivului (de exemplu să seteze un jumper într-o poziție). După efectuarea secvenței de pași, microcontroller-ul va porni bootloader-ul, iar aesta intră în modul de actualizare, mod în care așteaptă primirea bundle-ului pe portul serial. Când bundle-ul a fost primit de către microcontroller, bootloader-ul se va ocupa cu decriptarea și verificarea integrității transferului (de data aceasta între flasher si microcontroller). Dacă totul a decurs cu succes, firmware-ul este actualizat, iar flasher-ul este notificat ca atare. În caz contrar, scrierea firmware-ului este anulată, iar flasher-ului îi este notificat eșecul. În cazul în care utilizatorul nu execută pașii pentru pornirea modului de actualizare, bootloader-ul va porni aplicația existentă în memoria microcontroller-ului fără să aștepte niciun fel de input.

Din motive de securitate bootloader-ul nu va putea fi modificat prin el însuși, ci doar printr-un proces clasic de programare, doar firmware-ul executat de bootloader va putea fi modificat prin procesul descris aici.