

Analiza Sistem Software

Boot Loader

Student:

Bitai Oliver

An: IV Gr: 1.1

09.01.2018 Sectia:I.S.

Contents

[Introducere 2](#_Toc503380994)

[Obiectiv: 2](#_Toc503380995)

[Cerinte 2](#_Toc503380996)

[Cerinte functionale: 2](#_Toc503380997)

[Cerinte non-functionale: 3](#_Toc503380998)

[Cerinte care influenteaza arhitectura 3](#_Toc503380999)

[Componente 3](#_Toc503381000)

[Descompunerea in component 3](#_Toc503381001)

[Prezentare sistemului software 4](#_Toc503381002)

[Perspectiva de hardware arhitectura 4](#_Toc503381003)

[Perspectiva de software arhitectura 5](#_Toc503381004)

[Indicatori de calitate 5](#_Toc503381005)

[Fiabilitate 5](#_Toc503381006)

[Performanta 5](#_Toc503381007)

[Securitatea 6](#_Toc503381008)

[Tehnologi middleware 6](#_Toc503381009)

[Modele si stiluri arhitecturale 6](#_Toc503381010)

[Scenarii de validare 6](#_Toc503381011)

# Introducere

## Obiectiv:

Compania Continental Automotiv Romania, departamentul I ID (Interior Instrumentation & Driver) se ocupa de dezvoltare software si hardware ce tin de confortul si utilitatea elementelor din interiorul masini. Softwareul de boot loader reprezinta un interes foarte mare in acest domeniu fiindca elementele din interiorul masini (radio,clima,bord etc) precum si display-urile (navigatie,bord,head-up display etc) sunt controlate de microcontrolere care trebuie programate si eficientizate cat mai bine incat sa indeplineasca functionalitatea cat mai bine si aceste functionalitati sa fie realizate de catre microcontrolere cat mai mici. Prin urma anilor nu doar complexitatea functionalitatilor a crescut si numarul de microcontrolere care au functionalitati care se leaga intre ele, deci o comunicare sigura si asincrona de catre oricare microcontorler este foarte important.

Boot loaderul asigura incarcarea aplicatiei pe microcontroler intr-un mod sigur, ofera posibilitatea de testare prin anumite tool-uri a aplicatiei precum si in momentul alimentary a microcontrolerului pornirea in siguranta a aplicatiei.

# Cerinte

## Cerinte functionale:

* Fiind vorba despre un boot loader el nu necesita nici un sistem de operare sau alte software ajutatoare pentru a rula.
* El este la primul nivel de program care executa microcontrolerul.
* El are in sine toate configuratiile necesare pentru a putea fi rulat de microcontroller.
* In unele cazuri el este special construit/configurat pentru un singur tip de microcontroller.
* Trebuie conceput sa respecte o anumita structura.
* Boot loaderul nu interactioneaza cu alte dispositive sau microcontrolere dar interactioneaza cu aplicatii care trimit comenzi pe magistrala de comunicare.
* El are legatura doar cu o singura aplicatie pe care el poate valida si porni.
* In cazul in care in memoria flash gazeste o aplicatie validate boot loaderul isi termina executia si porneste aplicatia respectiva.
* Boot loaderul este capabil sa descararce aplicatii atat in memoria flash cat si in memoria RAM a microcontrolerului.
* In cazul unui actualizari de boot loader din motive de siguranta boot loaderul vechi nu poate fi sters/corupt pana in momentul in care noul boot loader este validat ca functional.
* In cazul in care in memoria flash gazeste o aplicatie validate boot loaderul isi termina executia si porneste aplicatia respectiva.

## Cerinte non-functionale:

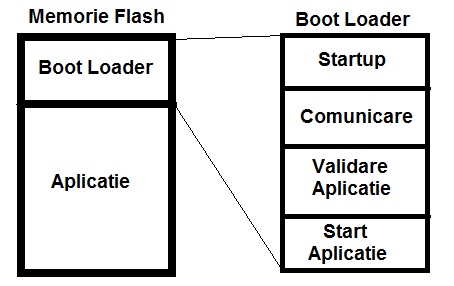
* Boot loaderul pentru a rula are nevoie de componente hardware minime de 16k bytes de memorie flash, 2k bytes de memorie RAM si de o frecventa de procesare de 16Mhz.
* El poate folosi si memori externe.
* In lipsa unui boot loader existent el poate fi programat in memoria flash a microcontrolerului doar cu ajutorul unui debugger sau alte dispositive care sunt capabile de programare flash cu interfata hardware direct legat la magistrala de date si adrese a flashului.
* El este plasat la inceputul memoriei flash.
* Prin functionare el nu poate sa isi corupe pe sine dar poate descarca aplicatii care pot sa faca o actualizare a boot loaderului.
* Boot loaderul trebuie ajustat pentru fiecare frecventa de microcontroller separat.
* Viteza executiei boot loaderului fiind vorba de o aplicatie embedded depinde de frecventa microcontrolerului dar are puncte de “asteptare” care sunt de acelasi durata independent de frecventa microcontrolerului (punctele de asteptare sunt folosite pentru a porni anumite functionalitati a boot loader-ului).

## Cerinte care influenteaza arhitectura

* Componentele hardware a microcontrollerului.
* Configuratia microcontrolerului
* Functia pe care o sa indeplineasca microcontorlerul
* Portabilitate

# Componente

## Descompunerea in component



**Boot Loader:** Este situate la inceputul memoriei flash si reprezinta softwarul care este rulat prima data de microcontroller.Rolul lui este - la pornirea sistemului – pornirea aplicatiei daca gaseste o aplicatie valida si realizarea comunicari cu magistrala daca primeste comenzi intr-un interval de timp scurt imediat dupa pornirea lui.

**Startup:** Este modulul din boot loader care asigura la pornirea sistemului incarcarea configuratiilor,datelor si a variabilelor necesare pentru rularea boot loaderului in memoria RAM.

**Comunicare:** Este modulul care se executa dupa startup si care sta in asteptare o perioada de timp foarte mic (10ms) in care asteapta comenzi de pe magistrala. Daca primeste comanda de activarea comunicatiei el ramane in aceasta stare pana cand nu primeste comanda de incheiere a comunicatiei si poate executa diferite comenzi precum descarcarea aplicatiei. Daca nu primeste in acest timp de asteptare nici o comanda de pe magistrala el preda executia podulului urmator.

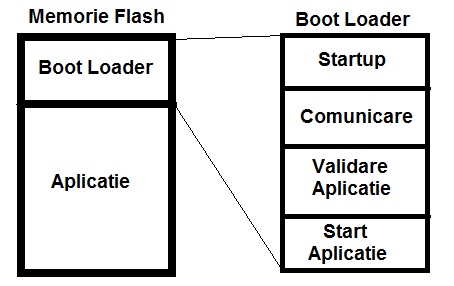
**Validare Aplicatie:** Acest modul are doua roluri: 1) in cazul in care o noua aplicatie era descarcat in timpul executari modulului de comunicare el poate valida aplicatia. 2) verifica daca exista o aplicatie care deja este validata si daca da preda executia la modulul start aplicatie, daca nu incheie executia boot loaderului.

**Start Aplicatie:** Pregateste aplicatia pentru a intra in rulare prin incarcarea anumitor parti a aplicatiei din flash in RAM dupa care incheie executia boot loaderului si preda executia aplicatiei.

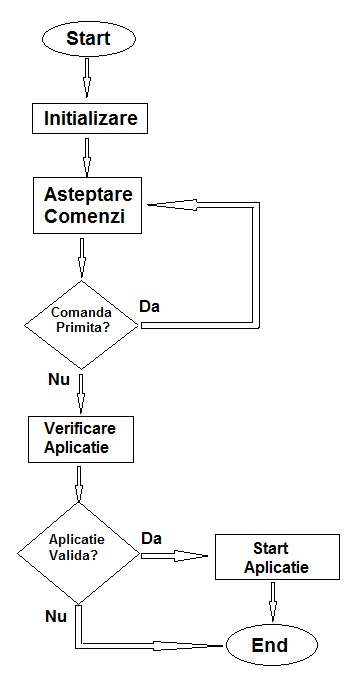
**Aplicatia:** Este un software care indeplineste mai multe functii si care este descarcat de catre boot loader. El realizeasa toate funcrionalitatile de care utilizatori beneficiaza, o data pornit pana la oprirea sistemului el ramane in executie.

# Prezentare sistemului software

## Perspectiva de hardware arhitectura



## Perspectiva de software arhitectura



# Indicatori de calitate

## Fiabilitate

Acest boot loader va fi in functionare intr-o masina in care v-a rula de foarte multe ori (ordine de 2-3000 de ori) in timpul de viata al masini. Boot loaderul este conceput,proiectat si realisat intr-un mod foarte bine gandit si testat incat fiabilitatea lui este foarte mare. Boot loaderul fiind softwareul care porneste aplicatia este inacceptabil ca el sa nu ofere fiabilitate maxima. Fiabilitatea o asigura in modul in care el este conceput precum si simplicitatea lui.

## Performanta

Datorita simplicitati boot loaderului el are o performanta de viteza foarte mare precum si o performanta buna de procesare a comenzilor si a datelor primite pe magistrala.

## Securitatea

Orice comunicare efectuata cu boot loaderul are un process bine definit care defineste atat formatul comenzilor si a datelor cat si teporizarea intre ele. Acest protocol este dezvoltat intern in corporatie si tinut sub confidentialitate.

# Tehnologi middleware

Se utilizeasa coda de mesaje care permite procesarea unui volum mare de mesaje/comenzi intr-o unitate de timp si ofera garantia ca mesajele/comenzile ajung la destinatar.

In unele cazuri la transferal unui volum de date mari se utilizeasa double-buffering pentru a fi sigur ca nici o bucata de date sa nu fie pierdut sau eronat.

# Modele si stiluri arhitecturale

Boot loaderul este construit conform modelului arhitectural utilizat in sisteme embedded.

# Scenarii de validare

* Boot loaderul se valideaza de sine statator pentru in cazul in care nu functioneaza practice microcontrolerul nu mai raspunde si nu face nimic.
* In comunicare pentru validarea mesajelor se folosesc check-sumuri pentru a verifica corectitudinea mesajelor.
* In comunicare pentru validarea mesajelor pentru orce mesaj se asteapta un raspuns care poate fi pozitiv sau negativ.