



Elektrobit

# EB tresos Bootloader

Internal Requirements

1.1.1, RELEASED



Elektrobit Automotive GmbH  
Am Wolfsmantel 46  
91058 Erlangen, Germany  
Phone: +49 9131 7701 0  
Fax: +49 9131 7701 6333  
Email: [info.automotive@elektrobit.com](mailto:info.automotive@elektrobit.com)

## **Technical support**

## **Legal disclaimer**

Confidential and proprietary information

ALL RIGHTS RESERVED. No part of this publication may be copied in any form, by photocopy, microfilm, retrieval system, or by any other means now known or hereafter invented without the prior written permission of Elektrobit Automotive GmbH.

All brand names, trademarks and registered trademarks are property of their rightful owners and are used only for description.

Copyright 2019, Elektrobit Automotive GmbH.

# Table of Contents

1. Requirements of EB tresos Bootloader for Essentials .....	6
1.1. Software structure .....	6
1.1.1. Memory segment .....	6
1.1.2. Example of memory mapping .....	7
1.1.3. Default Memory segment structure .....	7
1.1.3.1. Default Application validity .....	8
1.1.3.2. Default Fingerprint .....	10
1.1.4. Dual Memory Bank .....	11
1.2. Programming sequence .....	12
1.2.1. Startup sequence .....	12
1.2.2. Programming sequence adaptation .....	16
1.3. CRC calculation .....	19
1.4. Request/Response format .....	19
1.4.1. Response management .....	19
1.4.2. DiagnosticSessionControl (10 hex) service .....	21
1.4.2.1. Programming Session .....	21
1.4.2.1.1. Request message definition .....	22
1.4.2.1.2. Positive response message definition .....	22
1.4.2.1.3. Supported negative response codes (NRC_) .....	22
1.4.2.2. Default Session .....	23
1.4.2.2.1. Request message definition .....	23
1.4.2.2.2. Positive response message definition .....	23
1.4.2.2.3. Supported negative response codes (NRC_) .....	23
1.4.2.3. Extended Session .....	24
1.4.2.3.1. Request message definition .....	24
1.4.2.3.2. Positive response message definition .....	24
1.4.2.3.3. Supported negative response codes (NRC_) .....	25
1.4.3. ECUReset (11 hex) service .....	25
1.4.3.1. Request message definition .....	25
1.4.3.2. Positive response message definition .....	26
1.4.3.3. Supported negative response codes (NRC_) .....	26
1.4.4. EraseMemory (31 hex) .....	26
1.4.4.1. Logical Block erasing .....	26
1.4.4.1.1. Request message definition .....	26
1.4.4.2. Address erasing .....	27
1.4.4.2.1. Request message definition .....	27
1.4.4.3. Full erasing .....	28
1.4.4.3.1. Request message definition .....	28
1.4.4.4. Positive response message definition .....	28

1.4.4.5. Supported negative response codes (NRC_) .....	28
1.4.5. SecurityAccess (27 hex) .....	29
1.4.5.1. GetSeed .....	29
1.4.5.1.1. Request message definition .....	29
1.4.5.1.2. Positive response message definition .....	30
1.4.5.1.3. Supported negative response codes (NRC_) .....	30
1.4.5.2. SendKey .....	30
1.4.5.2.1. Request message definition .....	30
1.4.5.2.2. Positive response message definition .....	31
1.4.5.2.3. Supported negative response codes (NRC_) .....	31
1.4.6. RequestDownload (34 hex) .....	31
1.4.6.1. Request message definition .....	32
1.4.6.2. Positive response message definition .....	32
1.4.6.3. Supported negative response codes (NRC_) .....	32
1.4.7. TransferData (36 hex) .....	33
1.4.7.1. Request message definition .....	33
1.4.7.2. Positive response message definition .....	33
1.4.7.3. Supported negative response codes (NRC_) .....	34
1.4.8. RequestTransferExit (37 hex) .....	34
1.4.8.1. Request message definition .....	34
1.4.8.2. Positive response message definition .....	35
1.4.8.3. Supported negative response codes (NRC_) .....	35
1.4.9. TesterPresent (3E hex) .....	35
1.4.9.1. Request message definition .....	35
1.4.9.2. Positive response message definition .....	36
1.4.9.3. Supported negative response codes (NRC_) .....	36
1.4.10. CheckMemory (31 hex) .....	36
1.4.10.1. Request message definition .....	36
1.4.10.2. Positive response message definition .....	37
1.4.10.3. Supported negative response codes (NRC_) .....	37
1.4.11. CheckProgrammingDependencies (31 hex) .....	38
1.4.11.1. Request message definition .....	38
1.4.11.2. Positive response message definition .....	38
1.4.11.3. Supported negative response codes (NRC_) .....	39
1.4.12. PreCondition Check (31 hex) .....	39
1.4.12.1. Request message definition .....	39
1.4.12.2. Positive response message definition .....	40
1.4.12.3. Supported negative response codes (NRC_) .....	40
1.4.13. ControlDTCSettings (85 hex) .....	40
1.4.13.1. Request message definition .....	41
1.4.13.2. Positive response message definition .....	41
1.4.13.3. Supported negative response codes (NRC_) .....	41

1.4.14. CommunicationControl (28 hex) .....	41
1.4.14.1. Request message definition .....	42
1.4.14.2. Positive response message definition .....	42
1.4.14.3. Supported negative response codes (NRC_) .....	42
1.4.15. Read Data by Identifier .....	43
1.4.16. Write Fingerprint .....	43
1.4.17. Write Data by Identifier .....	43
1.5. EB specific programming sequence .....	43
1.5.1. Open programming session .....	44
1.5.2. Unlock ECU (optional) .....	44
1.5.3. Erasing Flash .....	45
1.5.4. Write segment identifier (optional) .....	46
1.5.5. Check reprogramming information .....	47
1.5.6. Program FLASH memory .....	48
1.5.7. Close programming .....	49
1.5.8. Check application validity Start routine .....	50
1.5.9. Close programming session .....	51
1.5.10. Define DID .....	52
1.5.10.1. Read all segment identifier .....	52
2. Glossary .....	53
2.1. EB tresos Bootloader .....	53

# 1. Requirements of EB tresos Bootloader for Essentials

This document list all EB tresos Bootloader for Essentials requirements.

## 1.1. Software structure

### 1.1.1. Memory segment

Each project shall define its memory mapping.

The bootloader memory is split in segments.

The following requirements apply to memory structure,;

Id:	IREQ_OEMIND_001	
Version:	1	
Description:	A segment shall be defined as application or calibration data.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_002	
Version:	2	
Description:	When BlockId are used to perform erasing, Bootloader shall accept the download of a single or multiple segments in single download sequence EraseMemory/RequestDownload/TransferData/RequestTransferExit/CheckMemory.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_003	
Version:	1	
Description:	Bootloader shall accept only segment that are aligned on Hardware memory sector:	

	<ul style="list-style-type: none"> <li>▶ A memory segment shall start at the first address of a memory sector</li> <li>▶ A memory segment shall end at the last address of a memory sector</li> </ul>	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	guide, tcs_sw	

### 1.1.2. Example of memory mapping

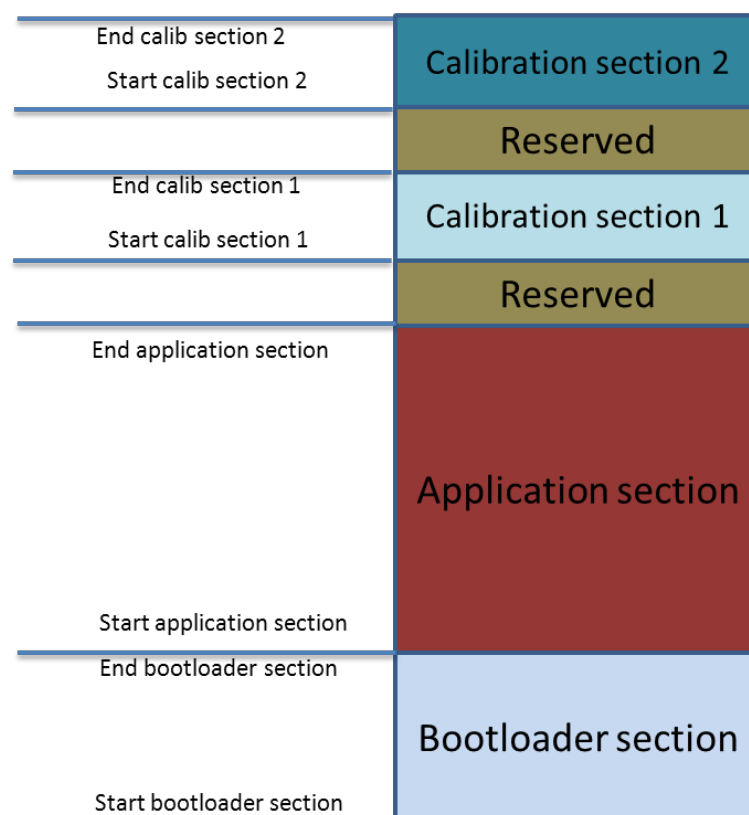


Figure 1.1. Example of memory mapping

### 1.1.3. Default Memory segment structure

Depending of the project needs, customer can decide to use this default memory segment structure if no other requirement exist for the project, or decide to use project specific segment structure e.g. if some OEM requirement shall apply

The Bootloader configuration allow to select:

- ▶ Default or custom application validity check shall be used
- ▶ Default or custom fingerprint shall be used

The following requirements apply if default implementation is used:

Id:	IREQ_OEMIND_086	
Version:	1	
Description:	Bootloader shall manage for every segments: <ul style="list-style-type: none"> <li>▶ A fingerprint used to identify the segment version</li> <li>▶ Validity flags to determine if the segment is valid or not.</li> </ul>	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

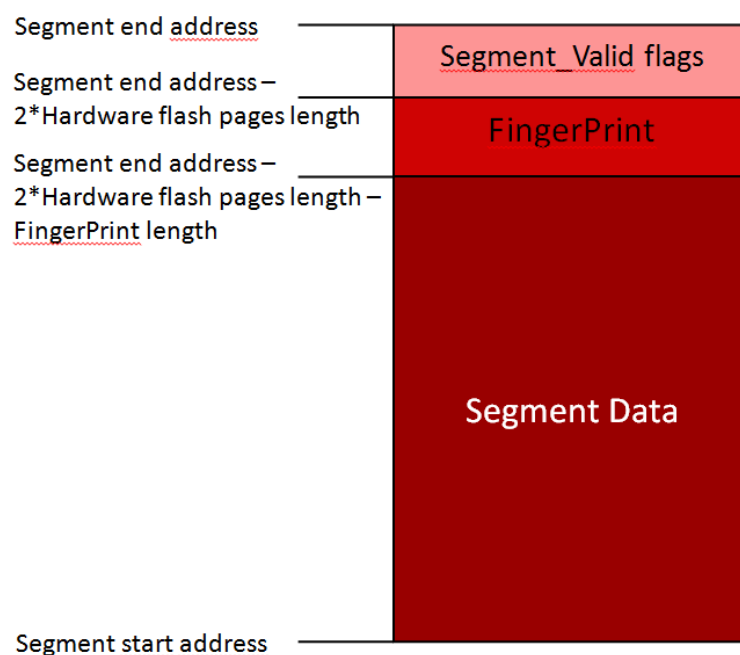


Figure 1.2. Default segment details

### 1.1.3.1. Default Application validity

Id:	IREQ_OEMIND_004	
Version:	1	
Description:	If EB default Application validity is configured, Bootloader shall manage a segment valid area for every segment:	





	<ul style="list-style-type: none"> <li>▶ The segment valid area is divided in two parts a valid flag and a revoked flag</li> <li>▶ Both flags need to be written separately at a specific time during the reprogramming sequence</li> <li>▶ The validity flag is set when the segment checksum is considered as correct</li> <li>▶ The revoked flag is set before starting the erasing to invalidate the segment</li> <li>▶ Two hardware flash pages are then required to store these flags.</li> </ul>
Provides coverage to: (Id-Version   Variants)	FBL-069-1
Needs coverage of:	dsn, tcs_sw

Id:	IREQ_OEMIND_021
Version:	1
Description:	<p>The following value shall be used for the segment valid area:</p> <ul style="list-style-type: none"> <li>▶ Invalid: 0xFFFFFFFFFFFFFFFF(*) / It's the FLASH erased value. The flag is set to this value when the segment is erased</li> <li>▶ Revoked: 0x5245564F4B454420 (meaning "REVOKED " in ASCII) / This flag means that the segment has been invalidate before erasing start. If this flag is set the segment shall not be consider as valid</li> <li>▶ Valid: 0x50524F4752414d4d (meaning "PROGRAMM" in ASCII) / This means the segment is valid. This flag is set to this value once the self-check has been run successfully on the segment</li> </ul>
Provides coverage to: (Id-Version   Variants)	FBL-069-1
Needs coverage of:	dsn

(\*) this value depends on the FLASH hardware technology

Id:	IREQ_OEMIND_022
Version:	1
Description:	The Validity flag shall be placed at the segment end address minus the Flash memory page (min 0x8)
Provides coverage to: (Id-Version   Variants)	FBL-069-1
Needs coverage of:	dsn

Id:	IREQ_OEMIND_023	
Version:	1	
Description:	The Revoked flag shall be placed at the segment end address minus twice the Flash memory page (min 0x8)	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn	

Id:	IREQ_OEMIND_024	
Version:	1	
Description:	"Revoked flag" shall be set to "Invalid" before the segment is erased.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_025	
Version:	1	
Description:	"Validity" flag shall be set to "Valid" when reprogramming sequence is completed and validated (downloading complete + CheckMemory successful)	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

### 1.1.3.2. Default Fingerprint

Id:	IREQ_OEMIND_005	
Version:	1	
Description:	<p>If EB default Fingerprint is configured, Bootloader shall manage a fingerprint area for every segment:</p> <ul style="list-style-type: none"> <li>▶ The fingerprint is a 16bytes long area to store segment information</li> <li>▶ It needs either 16 Bytes or a full flash page if the page size is superior to 16Bytes.</li> <li>▶ It is the customer's responsibility to define the value of this flag</li> </ul>	



Provides coverage to: (Id-Version   Variants)	FBL-069-1
Needs coverage of:	dsn, tcs_sw

Id:	IREQ_OEMIND_026
Version:	1
Description:	If EB default Fingerprint is used, FingerPrint shall be stored into the segment just before the Segment_Valid flag. Note: If the hardware flash page is larger than 16bytes, the flag start address shall be set aligned to the flash page.
Provides coverage to: (Id-Version   Variants)	FBL-069-1
Needs coverage of:	dsn

Id:	IREQ_OEMIND_027
Version:	1
Description:	If EB default Fingerprint is used, FingerPrint shall be written after the segment has been erased.
Provides coverage to: (Id-Version   Variants)	FBL-069-1
Needs coverage of:	dsn, tcs_sw

## 1.1.4. Dual Memory Bank

When Dual Memory Bank is used, the EB tresos Bootloader supports the programming of an inactive memory bank while running the software on the active memory bank.

Each project shall decide if the dual memory banks should be used.

Id:	IREQ_OEMIND_087
Version:	1
Description:	The EB tresos Bootloader shall provide callbacks for the activation and usage of the dual memory bank.
Provides coverage to: (Id-Version   Variants)	FBL-074-1
Needs coverage of:	dsn, tcs_sw

## 1.2. Programming sequence

### 1.2.1. Startup sequence

Bootloader startup sequence shall behave as defined in the following sequence:

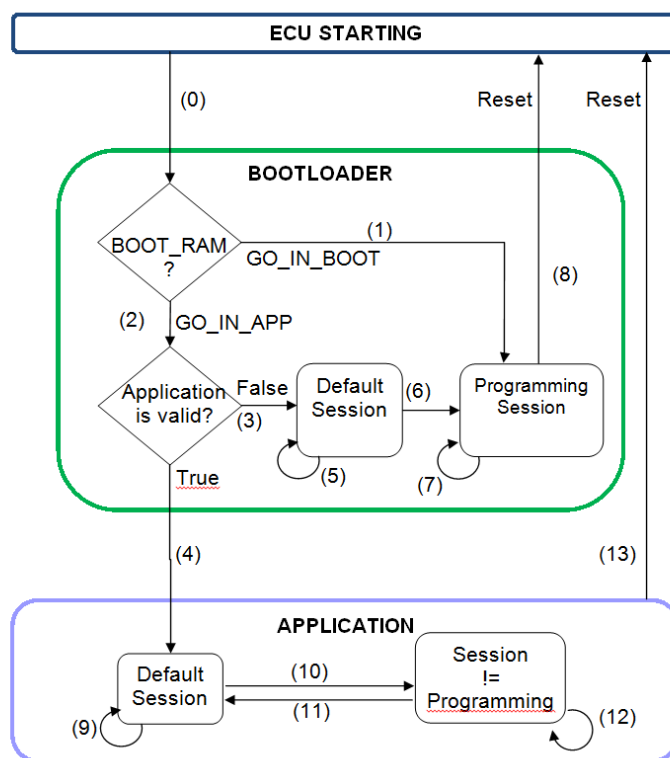


Figure 1.3. Startup Sequence

Id:	IREQ_OEMIND_006	
Version:	1	
Description:	Transition (0): <ul style="list-style-type: none"> <li>► Conditions: ECU power supply or reset</li> <li>► Actions: Initialization</li> </ul>	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_007	
Version:	1	
Description:	Transition (1): <ul style="list-style-type: none"> <li>► Conditions: BOOT_RAM flag equals “GO_IN_BOOT”.</li> <li>► Actions: Program stays in Bootloader Bootloader shall switch in Programming session Response to the Programming session request shall be sent BOOT_RAM shall then be set to “GO_IN_APP”</li> </ul>	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_008	
Version:	1	
Description:	Transition (2): <ul style="list-style-type: none"> <li>► Conditions: BOOT_RAM flag equals “GO_IN_APP”</li> <li>► Actions: Bootloader shall evaluate application validity (Read all segment SEGMENT_VALID flag)</li> </ul>	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_009	
Version:	1	
Description:	Transition (3): <ul style="list-style-type: none"> <li>► Conditions: Application is not valid</li> <li>► Actions: Program shall stay in Bootloader Bootloader shall stay in UDS Default session</li> </ul>	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_010	
Version:	1	
Description:	Transition (4):	

	<ul style="list-style-type: none"> <li>► Conditions: Application is valid</li> <li>► Actions: Program shall jump in Application Application shall stay in UDS Default session</li> </ul>
Provides coverage to: (Id-Version   Variants)	FBL-069-1
Needs coverage of:	dsn, tcs_sw

Id:	IREQ_OEMIND_011
Version:	1
Description:	Transition (5): <ul style="list-style-type: none"> <li>► Conditions: DSC(DefaultSession) request is received</li> <li>► Actions: Bootloader shall stay in UDS Default session</li> </ul>
Provides coverage to: (Id-Version   Variants)	FBL-069-1
Needs coverage of:	man, tcs_sw

Id:	IREQ_OEMIND_012
Version:	1
Description:	Transition (6): <ul style="list-style-type: none"> <li>► Conditions: DSC(ProgrammingSession) request is received</li> <li>► Actions: Bootloader shall switch in UDS Programming session</li> </ul>
Provides coverage to: (Id-Version   Variants)	FBL-069-1
Needs coverage of:	man, tcs_sw

Id:	IREQ_OEMIND_013
Version:	1
Description:	Transition (7): <ul style="list-style-type: none"> <li>► Conditions: DSC(ProgrammingSession) request is received</li> <li>► Actions: Bootloader shall switch in UDS Programming session</li> </ul>
Provides coverage to: (Id-Version   Variants)	FBL-069-1
Needs coverage of:	man, tcs_sw

Id:	IREQ_OEMIND_014	
Version:	1	
Description:	Transition (8): <ul style="list-style-type: none"> <li>► Conditions: DSC(XXX) request is received with XXX different from ProgrammingSession</li> <li>► Actions: ECU shall send positive response ECU shall reset</li> </ul>	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_015	
Version:	1	
Description:	Transition (8): <ul style="list-style-type: none"> <li>► Conditions: ProgrammingSession timeout is expired</li> <li>► Actions: ECU shall reset</li> </ul>	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Application transitions:

Transition (9):

- **Conditions:**
  - DSC(DefaultSession) request is received
- **Actions:**
  - Application shall stay in UDS Default session

Transition (10):

- **Conditions:**
  - DSC(XXX) request is received with XXX different from ProgrammingSession and different from DefaultSession.
- **Actions:**
  - Application shall switch in UDS XXX session

Transition (11):

- ▶ **Conditions:**
  - ▶ DSC(DefaultSession) request is received
- ▶ **Actions:**
  - ▶ Application shall switch in UDS Default session

Transition (12):

- ▶ **Conditions:**
  - ▶ DSC(XXX) request is received with XXX different from ProgrammingSession and different from DefaultSession
- ▶ **Actions:**
  - ▶ Application shall switch in UDS XXX session (or stay in XXX session if already active)

Transition (13):

- ▶ **Conditions:**
  - ▶ DSC(ProgrammingSession) request is received.
- ▶ **Actions:**
  - ▶ Application shall not transmit response (it shall be done in Bootloader)
  - ▶ Application shall set BOOT\_RAM to "GO\_IN\_BOOT"
  - ▶ ECU shall reset

## 1.2.2. Programming sequence adaptation

The programming sequence is based on UDS norm ISO 14229-1. This sequence can be adapted by customer.

Id:	IREQ_OEMIND_062
Version:	1
Description:	EB tresos Bootloader support the following download sequence (The mandatory diagnostic request are indicated with '*', other requests can be configured or not depending of the project needs): DiagnosticSessionControl(Extended session) RoutineControl(PreCondition Check) ControlDTCSetting(DisableDTCStorage) CommunicationControl(Disable Functional/Nm Communication) DiagnosticSessionControl(Programming session)* SecurityAccess(GetSeed)* SecurityAccess(CompareKey)* WriteDataByIdentifier(FingerPrint) RoutineControl(EraseMemory)* Request-Download* TransferData* RequestTransferExit* RoutineControl(CheckMemory)* RoutineControl(CheckProgrammingDependencies) EcuReset*



Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

The programming sequence is managed through a state machine that can be controlled by integration code:

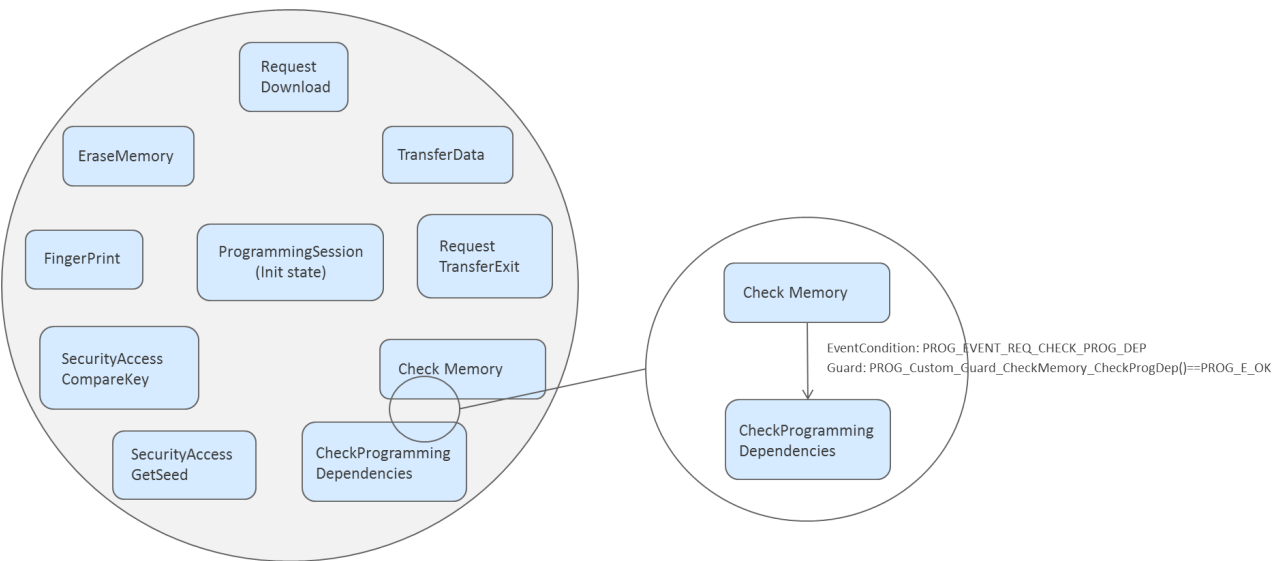


Figure 1.4. Programming state machine

For every transition between states, the state machine will require the authorization to perform the transition (and so accept the diagnostic request) by calling a callback function that shall be completed by Customer.

Id:	IREQ_OEMIND_063	
Version:	1	
Description:	For every transition requiring a custom check the Bootloader state machine shall call a guard Api PROG_Custom_Guard_CurrentState_RequestedState to get authorization to perform the transition	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

The Transition can be accepted (e.g from CheckMemory to CheckProgrammingDependencies states):

```
tProgStatus PROG_Custom_Guard_ChecMemory_CheckProgDep(void)
{
    /*Transition is accepted*/
    Return PROG_E_OK
}
```

```
}
```

Transition can be rejected (e.g from CheckMemory to CheckProgrammingDependencies states):

```
tProgStatus PROG_Custom_Guard_ChecMemory_CheckProgDep(void)
{
    /*Transition is rejected + a negative response shall be sent */
    PROG_Send_NRC(UDS_NRC_24);
    Return PROG_E_NOT_OK
}
```

The following table described the transitions that are allowed/forbidden or that can be customized at integration time.

Old request	New request									
	DSC02	SA (Seed)	SA (Key)	FP	RC (Erase-Memory)	RD	TD	RTE	RC (Check-Memory)	RC (Check-PrgDep)
DSC02										
SA (Seed)										
SA (Key)		(1)		(3)	(3)	(3)				(3)
FP		(1)								
RC (Erase-Memory)		(1)								
RD		(1)								
TD		(1)								
RTE		(1)							(2)	
RC (Check-Memory)		(1)								
RC (Check-PrgDep)		(1)								

Table 1.1. States transitions

Transition always accepted	A guard is called	Transition always rejected	Transition is: ▶ accepted if the successive CheckMemory are identical to the first one received
----------------------------	-------------------	----------------------------	--

			▶ rejected if the successive CheckMemory are different from the first one received
--	--	--	--

Table 1.2.

- (1) If ECU is already unlocked, no state change happen
- (2) If RTE processing has failed, transition to CheckMemory routine state is always rejected
- (3) If CompareKey processing has failed, transition to these states is always rejected

State machine restricts transitions between some states due to security reason, ISO14229 requirements or transition that can lead to software errors.

The following states can only be accessed when ECU is unlocked: FingerPrint(FP), EraseMemory(RC Erase), RequestDownload(RD), TransferData, RequestTransferExit, CheckMemory, CheckProgrammingDependencies.

In case of diagnostic request execution failure (sending of a negative response), the current state is kept (no reset of the state machine).

## 1.3. CRC calculation

The CRC use in the bootloader for Checking each segments can be:

- ▶ 16 bits CRC CCITT
- ▶ 32 bits CRC Ethernet

The CRC is computed on all received data between an erase routine and a CheckMemory routine.

## 1.4. Request/Response format

### 1.4.1. Response management

The diagnostic requests and responses are implemented according to UDS (ISO14229) norm.

Responses data that can change depending of the OEM requirement can be customized through dedicated callback that shall be implemented by customer.

#### Examples:

- For EraseMemory routine response, if the routine execution succeed, the following callback can be implemented to add a status byte in the response:

```
void PROG_Custom_EraseRoutine_Succeed(u8 * Data, u8 * DataLength)
{
    /*Add status*/
    Data[4]= 0x00;
    *DataLength=0x05;
}
```

- For EraseMemory routine response, if the routine execution failed and a positive response shall be sent, the following callback can be implemented to add a status byte in the positive response:

```
tProgStatus PROG_Custom_EraseRoutine_Failed(u8 * Data, u8 * DataLength, tUdsStatus * eUdsSta
{
    /* Send positive response with negative status*/
    *eUdsStatus=UDS_ACK;
    /*Add status*/
    Data[4]= 0x01;
    *DataLength=0x05;
}
```

- For EraseMemory routine response, if the routine execution failed and a negative response shall be sent, the following callback can be implemented:

```
tProgStatus PROG_Custom_EraseRoutine_Failed(u8 * Data, u8 * DataLength, tUdsStatus * eUdsSta
{
    /* Set NRC code to be use for the negative response */
    eUdsStatus = UDS_NRC_72;
}
```

This adaptation is required for the routines indicated in the following requirements.

Id:	IREQ_OEMIND_064	
Version:	1	
Description:	Before sending response to EraseMemory routine, Bootloader shall call a call-back allowing modifying the response before sending.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	



Needs coverage of:	dsn, tcs_sw
--------------------	-------------

Id:	IREQ_OEMIND_065	
Version:	1	
Description:	Before sending response to CheckMemory routine, Bootloader shall call a call-back allowing modifying the response before sending.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_066	
Version:	1	
Description:	Before sending response to PreConditionCheck routine, Bootloader shall call a callback allowing modifying the response before sending.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_067	
Version:	1	
Description:	Before sending response to CheckProgrammingDependencies routine, Bootloader shall call a callback allowing modifying the response before sending.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

## 1.4.2. DiagnosticSessionControl (10 hex) service

### 1.4.2.1. Programming Session

Id:	IREQ_OEMIND_068
Version:	1
Description:	The Service DSC 02 (0x10 02) shall follow the format described below

Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	man, tcs_sw	

#### 1.4.2.1.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	DiagnosticSessionControl	DSC	0x10
2	7	SuppresPosRspMsgIndicationBit	SPRMIB	0b0 - 0b1
2	6 to 0	DiagnosticSessionType: ProgrammingSession	DS	0x02

Table 1.3.

#### 1.4.2.1.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	DiagnosticSessionControl PR	DSCPR	0x50
2	-	echo of diagnosticSessionType parameter from the request message		
3-4	-	sessionParameterRecord P2	SPR_P2	value from UDS configuration
5-6	-	sessionParameterRecord P2*	SPR_P2*	value from UDS configuration divided by 10

Table 1.4.

#### 1.4.2.1.3. Supported negative response codes (NRC\_-)

Hex	Description
12	subFunctionNotSupported
13	incorrectMessageLengthOrInvalidFormat
22	conditionsNotCorrect

Table 1.5.

### 1.4.2.2. Default Session

Id:	IREQ_OEMIND_069	
Version:	1	
Description:	The Service DSC 01 (0x10 01) shall follow the format described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	man, tcs_sw	

#### 1.4.2.2.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	DiagnosticSessionControl	DSC	0x10
2	7	SuppresPosRspMsgIndicationBit	SPRMIB	0b0 - 0b1
2	6 to 0	DiagnosticSessionType: DefaultSession	DS	0x01

Table 1.6.

#### 1.4.2.2.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	DiagnosticSessionControl PR	DSCPR	0x50
2	-	echo of diagnosticSessionType parameter from the request message		
3-4	-	sessionParameterRecord P2	SPR_P2	value from UDS configuration
5-6	-	sessionParameterRecord P2*	SPR_P2*	value from UDS configuration divided by 10

Table 1.7.

#### 1.4.2.2.3. Supported negative response codes (NRC\_-)

Hex	Description
12	subFunctionNotSupported

Hex	Description
13	incorrectMessageLengthOrInvalidFormat
22	conditionsNotCorrect

Table 1.8.

### 1.4.2.3. Extended Session

This request is optional

Id:	IREQ_OEMIND_082	
Version:	1	
Description:	The Service DSC 03 (0x10 03) shall follow the format described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	man, tcs_sw	

#### 1.4.2.3.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	DiagnosticSessionControl	DSC	0x10
2	7	SuppresPosRspMsgIndicationBit	SPRMIB	0b0 - 0b1
2	6 to 0	DiagnosticSessionType: ExtendedSession	DS	0x03

Table 1.9.

#### 1.4.2.3.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	DiagnosticSessionControl PR	DSCPR	0x50
2	-	echo of diagnosticSessionType parameter from the request message		
3-4	-	sessionParameterRecord P2	SPR_P2	value from UDS configuration



Byte	Bit	Service and Parameter name	Mnemonic	Value
5-6	-	sessionParameterRecord P2*	SPR_P2*	value from UDS configuration divided by 10

Table 1.10.

#### 1.4.2.3.3. Supported negative response codes (NRC\_-)

Hex	Description
12	subFunctionNotSupported
13	incorrectMessageLengthOrInvalidFormat
22	conditionsNotCorrect

Table 1.11.

### 1.4.3. ECUReset (11 hex) service

Id:	IREQ_OEMIND_070	
Version:	1	
Description:	The Service ER 01 (0x11 01) shall follow the format described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	man, tcs_sw	

#### 1.4.3.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	ECUReset	ER	0x11
2	7	SuppresPosRspMsgIndicationBit	SPRMIB	0b0 - 0b1
2	6 to 0	Hardware reset	HR	0x01

Table 1.12.

### 1.4.3.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	ECUReset PR	ERPR	0x51
2	-	Hardware reset	HR	0x01

Table 1.13.

### 1.4.3.3. Supported negative response codes (NRC\_-)

Hex	Description
12	subFunctionNotSupported
13	incorrectMessageLengthOrInvalidFormat
22	conditionsNotCorrect

Table 1.14.

## 1.4.4. EraseMemory (31 hex)

### 1.4.4.1. Logical Block erasing

Id:	IREQ_OEMIND_071	
Version:	1	
Description:	If erasing shall be performed using a block Id, the Routine control EraseMemory (0x31 01 FF 00) shall follow the format described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

#### 1.4.4.1.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	RoutineControl	RC	0x31

Byte	Bit	Service and Parameter name	Mnemonic	Value
2	7	SuppresPosRspMsgIndicationBit	SPRMIB	0b0 - 0b1
2	6 to 0	RoutineControlType = StartRoutine	RCTP	0x01
3-4	-	RoutineIdentifier : Memory erase high byte/low byte	RI	0xFF/0x00
5	-	RoutineControlOptionRecord : Block Identifier size	RCEOR	1
6	-	RoutineControlOptionRecord : Block Identifier	RCEOR	00-FF

Table 1.15.

#### 1.4.4.2. Address erasing

Id:	IREQ_OEMIND_072	
Version:	1	
Description:	If erasing shall be performed using Address/Length, the Routine control Erase-Memory (0x31 01 FF 00) shall follow the format described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

##### 1.4.4.2.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	RoutineControl	RC	0x31
2	7	SuppresPosRspMsgIndicationBit	SPRMIB	0b0 - 0b1
2	6 to 0	RoutineControlType = StartRoutine	RCTP	0x01
3-4	-	RoutineIdentifier : Memory erase high byte/low byte	RI	0xFF/0x00
5-8	-	RoutineControlOptionRecord : Start address of the zone to erase	RCEOR	0x00000000-0xFFFFFFFF
6	-	RoutineControlOptionRecord : Length of the zone to erase	RCEOR	0x00000000-0xFFFFFFFF

Table 1.16.

Note: An additional 1 byte parameter ALFID "Address Length Format Identifier" defining the length of address and length parameter can be optionnaly supported in the request. This ALFID will be located in Byte 5. This is done with an EB tresos Studio configuration parameter and is statically defined.

### 1.4.4.3. Full erasing

Id:	IREQ_OEMIND_073		
Version:	1		
Description:	If erasing shall be performed on all memory (all configured memory segments), the Routine control EraseMemory (0x31 01 FF 00) shall follow the format described below. This requiremnt is similar to IREQ_265.		
Provides coverage to: (Id-Version   Variants)	FBL-069-1		
Needs coverage of:	dsn, tcs_sw		

#### 1.4.4.3.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	RoutineControl	RC	0x31
2	7	SuppresPosRspMsgIndicationBit	SPRMIB	0b0 - 0b1
2	6 to 0	RoutineControlType = StartRoutine	RCTP	0x01
3-4	-	RoutineIdentifier : Memory erase high byte/low byte	RI	0xFF/0x00

Table 1.17.

#### 1.4.4.4. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	RoutineControl PR	RCPR	0x71
2	-	echo of byte 2 from the request message		
3-4	-	echo of byte 3-4 from the request message		

Table 1.18.

#### 1.4.4.5. Supported negative response codes (NRC\_-)

Hex	Description
12	subFunctionNotSupported
13	incorrectMessageLengthOrInvalidFormat

Hex	Description
22	conditionsNotCorrect
24	requestSequenceError
31	requestOutOfRange
33	securityAccessDenied
72	GeneralProgrammingFailure
78	ResponsePending
7E	subFunctionNotSupportedInActiveSession
7F	ServiceNotSupportedInActiveSession

Table 1.19.

## 1.4.5. SecurityAccess (27 hex)

### 1.4.5.1. GetSeed

Id:	IREQ_OEMIND_074	
Version:	1	
Description:	The Service SecurityAccess(GetSeed) (0x27 0x) shall follow the format as described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	man, tcs_sw	

#### 1.4.5.1.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	SecurityAccess	SA	0x27
2	7	SuppresPosRspMsgIndicationBit	SPRMIB	0b0 - 0b1
2	6 to 0	SecurityAccessType : Seed request	SAT_RSD	01, 03, 05, 07-7D

Table 1.20.



#### 1.4.5.1.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	SecurityAccess PR	SAPR	0x67
2	-	echo of SecurityAccessType parameter from the request message		
3-6	-	SecuritySeed : Seed#1 (high byte) ... Seed#4 (low byte)	SECSEED	0x00-0xFF... 0x00-0xFF

Table 1.21.

#### 1.4.5.1.3. Supported negative response codes (NRC\_-)

Hex	Description
12	subFunctionNotSupported
13	incorrectMessageLengthOrInvalidFormat
22	conditionsNotCorrect
24	requestSequenceError
7E	subFunctionNotSupportedInActiveSession
7F	ServiceNotSupportedInActiveSession

Table 1.22.

### 1.4.5.2. SendKey

Id:	IREQ_OEMIND_075		
Version:	1		
Description:	The Service SecurityAccess(SendKey) (0x27 0x) shall follow the format as described below		
Provides coverage to: (Id-Version   Variants)	FBL-069-1		
Needs coverage of:	dsn, tcs_sw		

#### 1.4.5.2.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	SecurityAccess	SA	0x27

Byte	Bit	Service and Parameter name	Mnemonic	Value
2	7	SuppresPosRspMsgIndicationBit	SPRMIB	0b0 - 0b1
2	6 to 0	SecurityAccessType : Key	SAT_SK	02, 04, 06, 08-7E
3-6	-	SecurityKey :Seed#1 (high byte)...Seed#m (low byte)	KEY1 - KEY4	0x00 - 0xFF...0x00 - 0xFF

Table 1.23.

#### 1.4.5.2.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	SecurityAccess PR	SAPR	0x67
2	-	echo of SecurityAccessType parameter from the request message		

Table 1.24.

#### 1.4.5.2.3. Supported negative response codes (NRC\_-)

Hex	Description
12	subFunctionNotSupported
13	incorrectMessageLengthOrInvalidFormat
22	conditionsNotCorrect
24	requestSequenceError
35	invalidKey
7E	subFunctionNotSupportedInActiveSession
7F	ServiceNotSupportedInActiveSession

Table 1.25.

### 1.4.6. RequestDownload (34 hex)

Id:	IREQ_OEMIND_076
Version:	1

Description:	The Service RD (0x34) shall follow the format as described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

#### 1.4.6.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	RequestDownload	RD	0x34
2	-	DataFormatIdentifier	DFI	0x00
3	-	AddressAndLengthFormatIdentifier	ALFID	0x44
4-7	-	MemoryAddress	MA	0x00000000 - 0xFFFFFFFF
8-11	-	MemorySize	MS	0x00000000 - 0xFFFFFFFF

Table 1.26.

#### 1.4.6.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	RequestDownload PR	RDPR	0x74
2	-	LengthFormatIdentifier	LFID	0x10
3-4	-	MaxNumberOfBlockLength	MNROB	0x800-0xFFFF

Table 1.27.

#### 1.4.6.3. Supported negative response codes (NRC\_-)

Hex	Description
13	incorrectMessageLengthOrInvalidFormat
22	conditionsNotCorrect



Hex	Description
33	securityAccessDenied
70	uploadDownloadNotAccepted
7F	ServiceNotSupportedInActiveSession

Table 1.28.

### 1.4.7. TransferData (36 hex)

Id:	IREQ_OEMIND_077	
Version:	1	
Description:	The Service TransferData (0x36) shall follow the format as described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

#### 1.4.7.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	TransferData	TD	0x36
2	-	BlockSequenceCounter	BSC	0x00 - 0xFF
3-n	-	TransferRequestParameterRecord : transferRequestParameter#1 ... transferRequestParameter#n	TRPR	0x00-0xFF...- 0x00-0xFF

Table 1.29.

BlockSequenceCounter is a running counter. It starts from 1 and is increased on each TransferData request. It switches from 255 to 0 and goes on.

BlockSequenceCounter is reset to 1 on each RequestDownload request.

#### 1.4.7.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	TransferData PR	TDPR	0x76

Byte	Bit	Service and Parameter name	Mnemonic	Value
2	-	echo of BlockSequenceCounter parameter from the request message		

Table 1.30.

### 1.4.7.3. Supported negative response codes (NRC\_-)

Hex	Description
13	incorrectMessageLengthOrInvalidFormat
24	requestSequenceError
31	requestOutOfRange
71	transfertDataSuspend
72	generalProgrammingFailure
73	WrongBlockSequenceCounter
78	ResponsePending
7F	ServiceNotSupportedInActiveSession

Table 1.31.

### 1.4.8. RequestTransferExit (37 hex)

Id:	IREQ_OEMIND_078	
Version:	1	
Description:	The Service RequestTransferExit (0x37) shall follow the format as described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	man, tcs_sw	

#### 1.4.8.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	RequestTransferExit	RTE	0x37

Table 1.32.

### 1.4.8.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	RequestTransferExit PR	RTEPR	0x77

Table 1.33.

### 1.4.8.3. Supported negative response codes (NRC\_-)

Hex	Description
13	incorrectMessageLengthOrInvalidFormat
22	conditionsNotCorrect
24	requestSequenceError
31	requestOutOfRange
33	securityAccessDenied
70	uploadDownloadNotAccepted
78	ResponsePending
7F	ServiceNotSupportedInActiveSession

Table 1.34.

## 1.4.9. TesterPresent (3E hex)

Id:	IREQ_OEMIND_079	
Version:	1	
Description:	The Service TesterPresent (0x3E) shall follow the format as described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	man, tcs_sw	

### 1.4.9.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	TesterPresent	TP	0x3E

Byte	Bit	Service and Parameter name	Mnemonic	Value
2	-	SubFunction	BSC	0x00 / 0x80

Table 1.35.

#### 1.4.9.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	TesterPresent PR	TPPR	0x7E

Table 1.36.

#### 1.4.9.3. Supported negative response codes (NRC\_-)

Hex	Description
12	subFunctionNotSupported
13	incorrectMessageLengthOrInvalidFormat

Table 1.37.

### 1.4.10. CheckMemory (31 hex)

Id:	IREQ_OEMIND_080	
Version:	1	
Description:	The start Routine control CheckMemory (0x31 01 xx xx) shall follow the format as described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

#### 1.4.10.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	RoutineControl	RC	0x31

Byte	Bit	Service and Parameter name	Mnemonic	Value
2	7	SuppresPosRspMsgIndicationBit	SPRMIB	0b0 - 0b1
2	6 to 0	RoutineControlType = StartRoutine	RCTP	0x01
3-4	-	RoutineIdentifier : CheckMemory high byte/low byte	RI	0xXX/0xXX
5-n	-	RoutineControlOptionRecord : 16 bits or 32 bits Checksum	RCEOR	0x00000000-0xFFFFFFFF

Table 1.38.

#### 1.4.10.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	RoutineControl PR	RCPR	0x71
2	-	echo of byte 2 from the request message		
3-4	-	echo of byte 3-4 from the request message		
5*	-	routineStatusRecord	RSR	Customer specific value

Table 1.39.

\*Optional: the status can be added, at integration time, by customer if required for its project (see Response management section)

#### 1.4.10.3. Supported negative response codes (NRC\_-)

Hex	Description
12	subFunctionNotSupported
13	incorrectMessageLengthOrInvalidFormat
22	conditionsNotCorrect
24	requestSequenceError
31	requestOutOfRange
72	GeneralProgrammingFailure
78	ResponsePending
7E	subFunctionNotSupportedInActiveSession
7F	ServiceNotSupportedInActiveSession

Table 1.40.

## 1.4.11. CheckProgrammingDependencies (31 hex)

Id:	IREQ_OEMIND_085	
Version:	1	
Description:	The start Routine control CheckProgrammingDependencies (0x31 01 FF 01) shall follow the format as described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	man, tcs_sw	

### 1.4.11.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	RoutineControl	RC	0x31
2	7	SuppresPosRspMsgIndicationBit	SPRMIB	0b0 - 0b1
2	6 to 0	RoutineControlType = StartRoutine	RCTP	0x01
3-4	-	RoutineIdentifier : CheckProgrammingDependencies high byte/low byte	RI	0xFF/0x01

Table 1.41.

### 1.4.11.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	RoutineControl PR	RCPR	0x71
2	-	echo of byte 2 from the request message		
3-4	-	echo of byte 3-4 from the request message		
5*	-	routineStatusRecord	RSR	Customer specific value

Table 1.42.

\*Optional: the status can be added, at integration time, by customer if required for its project (see Response management section)

### 1.4.11.3. Supported negative response codes (NRC\_-)

Hex	Description
12	subFunctionNotSupported
13	incorrectMessageLengthOrInvalidFormat
22	conditionsNotCorrect
24	requestSequenceError
31	requestOutOfRange
72	GeneralProgrammingFailure
78	ResponsePending
7E	subFunctionNotSupportedInActiveSession
7F	ServiceNotSupportedInActiveSession

Table 1.43.

### 1.4.12. PreCondition Check (31 hex)

This request is optional

Id:	IREQ_OEMIND_081	
Version:	1	
Description:	The start Routine control PreConditionCheck (0x31 01 xx xx) shall follow the format as described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	man, tcs_sw	

#### 1.4.12.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	RoutineControl	RC	0x31
2	7	SuppresPosRspMsgIndicationBit	SPRMIB	0b0 - 0b1
2	6 to 0	RoutineControlType = StartRoutine	RCTP	0x01
3-4	-	RoutineIdentifier : PreConditionCheck high byte/low byte	RI	0xXX/0xXX

Table 1.44.

### 1.4.12.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	RoutineControl PR	RCPR	0x71
2	-	echo of byte 2 from the request message		
3-4	-	echo of byte 3-4 from the request message		
5*	-	routineStatusRecord	RSR	Customer specific value

Table 1.45.

\*Optional: the status can be added, at integration time, by customer if required for its project (see Response management section)

### 1.4.12.3. Supported negative response codes (NRC\_-)

Hex	Description
12	subFunctionNotSupported
13	incorrectMessageLengthOrInvalidFormat
22	conditionsNotCorrect
24	requestSequenceError
31	requestOutOfRange
7E	subFunctionNotSupportedInActiveSession
7F	ServiceNotSupportedInActiveSession

Table 1.46.

## 1.4.13. ControlDTCSettings (85 hex)

This request is optional

Id:	IREQ_OEMIND_083	
Version:	1	
Description:	The ControlDTCSettings request (0x85 02 xx xx) shall follow the format as described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	



Needs coverage of:

man, tcs\_sw

#### 1.4.13.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	ControlDTCSettings	CDTCS	0x85
2	-	DTCSettingType: Off	DTCSTP	0x02
3-n*	-	DTCSettingControlOptionRecord	DTCSCOR	0xXX

Table 1.47.

\*Optional: these parameters are unused by Bootloader, Bootloader only provide positive response without performing any action (No DTC support in Bootloader)

#### 1.4.13.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	ControlDTCSettings PR	CDTCSPR	0xC5
2	-	echo of byte 2 from the request message		

Table 1.48.

\*Optional

#### 1.4.13.3. Supported negative response codes (NRC\_-)

Hex	Description
12	subFunctionNotSupported
13	incorrectMessageLengthOrInvalidFormat
7E	subFunctionNotSupportedInActiveSession
7F	ServiceNotSupportedInActiveSession

Table 1.49.

### 1.4.14. CommunicationControl (28 hex)

This request is optional

Id:	IREQ_OEMIND_084	
Version:	1	
Description:	The CommunicationControl request (0x28 xx xx) shall follow the format as described below	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	man, tcs_sw	

#### 1.4.14.1. Request message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	CommunicationControl	CC	0x28
2*	-	controlType:	CTRLTP	xx
3*	-	communicationType	CTP	xx

Table 1.50.

\*these parameters are unused by Bootloader, Bootloader only provide positive response without performing any action (No communication control in Bootloader)

#### 1.4.14.2. Positive response message definition

Byte	Bit	Service and Parameter name	Mnemonic	Value
1	-	CommunicationControl PR	CCPR	0x68
2	-	echo of byte 2 from the request message		

Table 1.51.

#### 1.4.14.3. Supported negative response codes (NRC\_-)

Hex	Description
12	subFunctionNotSupported
13	incorrectMessageLengthOrInvalidFormat
7E	subFunctionNotSupportedInActiveSession
7F	ServiceNotSupportedInActiveSession

Table 1.52.

### 1.4.15. Read Data by Identifier

ReadDataByIdentifier is project/ECU specific and shall be implemented by customer: a callback will be called for every configured DID

### 1.4.16. Write Fingerprint

This request is optional

Fingerprint is project/ECU specific and shall be implemented by customer: a callback will be called on reception of the request in order customer can perform the verification and writing in memory of the fingerprint data

### 1.4.17. Write Data by Identifier

WriteDataByIdentifier is project/ECU specific and shall be implemented by customer: a callback will be called for every configured DID

## 1.5. EB specific programming sequence

The following sequence is done for a complete reprogramming using EB tester tool:

- ▶ Open programming UDS session (=> jump from application to bootloader): Diagnostic session control Programming session (0x10 02)
- ▶ Unlock the ECU: Security Access (0x27)
- ▶ Loop on memory segment (handled by TesterTool):
  - ▶ Erase FLASH memory application: Routine control: EraseMemory (0x31 01 FF 00)
  - ▶ Write segment identification: Write Data by Identifier: write FingerPrint (0x2E F1 84)
  - ▶ Application/Calibration downloading:
    - ▶ Check reprogramming information: Request download (0x34)
    - ▶ Program FLASH memory: Transfer Data (0x36)
    - ▶ Close FLASH memory reprogramming: Request Transfer Exit (0x37)
  - ▶ Check the application validity (CRC calculation): Routine control Check Memory (0x31 01 FF 01)
- ▶ End loop
- ▶ - ECU reset and jump to application (two possibilities):

- ▶ Diagnostic session control: default session (0x10 01)
- ▶ ECU reset : Hardware Reset (0x11 01)

The loop on memory segment from flash erasing to Programming Check could be done as many times as necessary to download all ECU memory segments (application and Calibrations segments).

It is also possible to program only one segment, such as the calibration (without reflashing the application). It's up to the TesterTool to manage the related segments.

### 1.5.1. Open programming session

See chapter "Request/Response format" for DiagnosticSessionControl request/response details.

### 1.5.2. Unlock ECU (optional)

See chapter "Request/Response format" for SecurityAccess request/response details.

ECU unlock is based on two successive SecurityAccess requests: SA1 and SA2.

The following requirements apply to SecurityAccess management:

Id:	IREQ_OEMIND_028	
Version:	1	
Description:	On "SA1" the ECU shall generate a 4 bytes random Seed	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_029	
Version:	1	
Description:	On "SA1" the ECU shall calculate the ECU key from the 4 bytes Seed and then respond to the request with the 4 bytes generated Seed.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_030	
-----	-----------------	--

Version:	1
Description:	The Security access algorithm used to calculate the ECU key is open. It shall be implemented by Tier1 into a dedicated callback. The algorithm shall be shared by the ECU and the downloading tool.
Provides coverage to: (Id-Version   Variants)	FBL-069-1
Needs coverage of:	dsn, tcs_sw

Id:	IREQ_OEMIND_031
Version:	1
Description:	On "SA2" reception, the ECU shall check if the SA1 has been correctly received. If the SA1 request was not received, a NRC_24 shall be returned.
Provides coverage to: (Id-Version   Variants)	FBL-069-1
Needs coverage of:	dsn, tcs_sw

Id:	IREQ_OEMIND_032
Version:	1
Description:	On "SA2" reception, the ECU shall compare the two keys: the generated one with the received one. If they are equal, a positive response shall be sent and the ECU shall be unlocked. If they are different, a NRC_35 shall be returned and the complete SA1 SA2 sequence shall be done again.
Provides coverage to: (Id-Version   Variants)	FBL-069-1
Needs coverage of:	dsn, tcs_sw

### 1.5.3. Erasing Flash

See chapter "Request/Response format" for RoutineControl request/response details.

The following requirements apply to Erasing Flash management:

Id:	IREQ_OEMIND_034
Version:	1
Description:	On « flash erasing » request, the ECU checks if it is unlocked. If locked, the ECU returns NRC 33.

Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	man, tcs_sw	
Id:	IREQ_OEMIND_035	
Version:	1	
Description:	On « flash erasing » request, The ECU shall evaluate the received parameters. If the address and length received are out of range of the configured segment, a NRC_31 shall be returned.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	
Id:	IREQ_OEMIND_036	
Version:	1	
Description:	Once the erasing of the memory segment is complete: the ECU shall return a positive response if the erasing is completed without error.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	
Id:	IREQ_OEMIND_037	
Version:	1	
Description:	Once the erasing of memory segment is complete: the ECU shall return a NRC_72 if the erasing failed (or completed with error)	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

### 1.5.4. Write segment identifier (optional)

See chapter "Request/Response format" for WriteDataByIdentifier request/response details.

Note: This step can be skipped from the reprogramming sequence. This is done with an EB tresos Studio configuration parameter and is statically defined.

The following requirements apply to Write segment identifier management:

Id:	IREQ_OEMIND_038	
Version:	1	
Description:	The WDBI shall be called just after the erasing request. If it is not called at this time, a NRC_24 is returned.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

### 1.5.5. Check reprogramming information

See chapter "Request/Response format" for RequestDownload request/response details.

The following requirements apply to Check reprogramming information management:

Id:	IREQ_OEMIND_040	
Version:	1	
Description:	On valid "RequestDownload" request, the ECU shall check if it is unlocked. If it is not, the ECU shall return NRC_33.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	man, tcs_sw	

Id:	IREQ_OEMIND_041	
Version:	1	
Description:	On valid "RequestDownload" request, the ECU shall check if the received DataFormatIdentifier is coherent (see [REF1]). If it is not, the ECU shall return NRC_31.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_042	
Version:	1	
Description:	On valid "RequestDownload" request, ECU shall check if the received addressAndLengthFormatIdentifier is coherent (see [REF1]). If it is not, the ECU shall return NRC_31.	

Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_044	
Version:	1	
Description:	On valid "RequestDownload" request, if all the parameters are correct, the ECU shall respond positively to the request.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

### 1.5.6. Program FLASH memory

See chapter "Request/Response format" for TransferData request/response details.

The following requirements apply to Program FLASH memory management:

Id:	IREQ_OEMIND_045	
Version:	1	
Description:	On valid "TransferData" request, the ECU shall check that the RequestDownload has been correctly completed. If not, the ECU shall return NRC 24.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_046	
Version:	1	
Description:	On valid "TransferData" request, the ECU shall check the « BlockSequenceCounter». If not correct, the ECU shall return NRC 73.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_047	
Version:	2	



Description:	On valid "TransferData" request, the size of the data received shall have at least 1 data. If not correct, the ECU shall return NRC 13.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	guide, tcs_sw	

Id:	IREQ_OEMIND_048	
Version:	2	
Description:	On valid "TransferData" request, the ECU shall check that the received data doesn't exceed the total number of data to be received. If the data number is exceeded, the ECU shall return NRC_71.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_050	
Version:	1	
Description:	On valid "TransferData" request, if all parameters are correct the ECU shall write the received data. If the data writing returns an error, NRC_72 shall be returned otherwise a positive response is send.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

### 1.5.7. Close programming

See chapter "Request/Response format" for RequestTransferExit request/response details.

The following requirements apply to Close programming management:

Id:	IREQ_OEMIND_051	
Version:	1	
Description:	On valid "RequestTransferExit" request, the ECU shall check the TransferData has been correctly completed. If not, ECU shall return NRC 24. .	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	

Needs coverage of:	dsn, tcs_sw
--------------------	-------------

Id:	IREQ_OEMIND_052	
Version:	1	
Description:	On valid "RequestTransferExit" request, the ECU shall check that the correct number of data has been received. If not, the ECU shall return NRC 24.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

### 1.5.8. Check application validity Start routine

This routine is used to validate the data previously downloaded into a configured segment.

It calculates a checksum of the "useful data" written in the segment without taking into account the Segment \_Valid and the FingerPrint.

See chapter "Request/Response format" for RoutineControl request/response details.

The following requirements apply to Check application validity management:

Id:	IREQ_OEMIND_054	
Version:	1	
Description:	On valid "programing checking" request, the ECU shall check if it is unlocked. If it is not, the ECU shall return NRC_33. .	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	man, tcs_sw	

Id:	IREQ_OEMIND_056	
Version:	1	
Description:	On valid « programing checking » request, if all the parameters are correct, the ECU shall start the requested segment checksum calculation.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_057	
Version:	1	
Description:	When checksum calculation is completed, the ECU shall compare the calculated Checksum to the one which has been received. Then, the ECU shall prepare response request. If checksums match, a positive response shall be send. If checksums don't match, NRC_72 shall be returned	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_058	
Version:	1	
Description:	If the checksum of the segment is consistent with the one received in parameter, the Segment_Valid flag shall be set to Valid.	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

### 1.5.9. Close programming session

See chapter "Request/Response format" for DiagnosticSessionControl request/response details.

The following requirements apply to Close programming session management:

Id:	IREQ_OEMIND_059	
Version:	1	
Description:	On programming session exit (via explicit DSC request), the ECU shall return a positive response and then reset. .	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

Id:	IREQ_OEMIND_060	
Version:	1	
Description:	On programming session exit (via explicit ER request or any other condition such as session timeout), the ECU shall reset.	

Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

## 1.5.10. Define DID

### 1.5.10.1. Read all segment identifier

A RDBI request is defined to read the application Software Identification (EB Fingerprint). This request returns the identification code of all the segment flashed into the ECU.

Id:	IREQ_OEMIND_061	
Version:	1	
Description:	The bootloader shall be able to return the segment ID of all the configured segment into a RDBI request	
Provides coverage to: (Id-Version   Variants)	FBL-069-1	
Needs coverage of:	dsn, tcs_sw	

See chapter "Request/Response format" for RDBI request/response details.

## 2. Glossary

### 2.1. EB tresos Bootloader

You can find in the following table a definition of the different naming used with the Bootloader documentation

Term	Definition
Application	Runnable code of ECU software that can be updated by a Bootloader
AUTOSAR	Automotive Open System Architecture  A consortium of OEMs, Tier1's and semiconductor vendors that work on standardization of an automotive software architecture.
Bootloader	Permanent software located in Flash memory allowing updating application or calibration software of an ECU. It provides communication with an external Tester  <b>Also called:</b> Flashloader.
Calibration	Configuration code of an ECU software that can be updated by a Bootloader
CRY	Cryptographic primitives implementation (used by the CSM)  This is an AUTOSAR module
CSM	Cryptographic Service Manager  This is an AUTOSAR module
Download verification	Operation realized after a download to verify that the data present in memory match the expected downloaded data. This is usually done using a CRC, Checksum or Hash.  <b>Also called:</b> Check programming dependencies, check memory, message digest
Flash sector	Smallest amount of flash memory that can be erased in one pass. Size depends on flash technology used.
Flash page	Smallest amount of flash memory that can be programmed in one pass. Size depends on flash technology used.
Segment	Continuous address range within a logical block.  Static segment: static address and size present in configuration defining the memory section that can be erased or programmed.

Term	Definition
	Dynamic segment: dynamic address and size present in RAM defining the memory section that have been programmed.
HSM	Hardware Security Module
Logical block	Smallest amount of flash memory that can be individually reprogrammed. Size depends on technology (flash sector, flash page) and user settings.
Memory Erased check	Verification if memory is already erased, before trying to erase it. It allow sparing time, if the memory is already erased.  <b>Also called:</b> blank check.
PDU	Protocol Data Unit  Any piece of information exchanged between two or more communicating entities
PduR	PDU Router  This is an AUTOSAR module
SecOC	Secure OnBoard Communication  This is an AUTOSAR module
SHE	Secure Hardware Extension  A hardware security extension specified by the HIS consortium.
Software integrity	Feature ensuring the software integrity before executing it.  <b>Also called:</b> Software authentication.
Software acceptance check	Feature verifying the validity of the received application/calibration software to be updated  <b>Also called:</b> Application/Calibration signature check.
Software coherency check	Feature verifying the coherency of the application/calibrations  <b>Also called:</b> Application coherency check, consistency check, application validity, Check Programming dependencies.
Streaming	Allow writting data in memory on reception of every consecutive frame of a TransferData request (No wait of the reception of the full request before starting the write). This improve the download performance.

Table 2.1. Bootloader definitions