

MOST

Media Oriented Systems Transport

Multimedia and Control
Networking Technology

MOST FBlock Enhanced Testability

Rev 3.0.4

02/2017



Legal Notice

COPYRIGHT

© Copyright 1999 - 2017 MOST Cooperation. All rights reserved.

LICENSE DISCLAIMER

Nothing on any MOST Cooperation Web Site, or in any MOST Cooperation document, shall be construed as conferring any license under any of the MOST Cooperation or its members or any third party's intellectual property rights, whether by estoppel, implication, or otherwise.

CONTENT AND LIABILITY DISCLAIMER

MOST Cooperation or its members shall not be responsible for any errors or omissions contained at any MOST Cooperation Web Site, or in any MOST Cooperation document, and reserves the right to make changes without notice. Accordingly, all MOST Cooperation and third party information is provided "AS IS". In addition, MOST Cooperation or its members are not responsible for the content of any other Web Site linked to any MOST Cooperation Web Site. Links are provided as Internet navigation tools only.

MOST COOPERATION AND ITS MEMBERS DISCLAIM ALL WARRANTIES WITH REGARD TO THE INFORMATION (INCLUDING ANY SOFTWARE) PROVIDED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

In no event shall MOST Cooperation or its members be liable for any damages whatsoever, and in particular MOST Cooperation or its members shall not be liable for special, indirect, consequential, or incidental damages, or damages for lost profits, loss of revenue, or loss of use, arising out of or related to any MOST Cooperation Web Site, any MOST Cooperation document, or the information contained in it, whether such damages arise in contract, negligence, tort, under statute, in equity, at law or otherwise.

FEEDBACK INFORMATION

Any information provided to MOST Cooperation in connection with any MOST Cooperation Web Site, or any MOST Cooperation document, shall be provided by the submitter and received by MOST Cooperation on a non-confidential basis. MOST Cooperation shall be free to use such information on an unrestricted basis.

TRADEMARKS

MOST Cooperation and its members prohibit the unauthorized use of any of their trademarks. MOST Cooperation specifically prohibits the use of the MOST Cooperation LOGO unless the use is approved by the Steering Committee of MOST Cooperation.

SUPPORT AND FURTHER INFORMATION

For more information on the MOST technology, please contact:

MOST Cooperation

Administration
Emmy-Noether-Str. 14
76131 Karlsruhe
Germany

Tel: (+49) (0) 721 966 50 00

E-mail: contact@mostcooperation.com
Web: www.mostcooperation.com



© Copyright 1999 - 2017 MOST Cooperation.
All rights reserved.

MOST is a registered trademark

Contents

BIBLIOGRAPHY	5
DOCUMENT HISTORY	6
1 INTRODUCTION	11
2 FUNCTION CATALOG	11
2.1 EnhancedTestability (FBlockID=0x0F)	11
2.1.1 AutoWakeup (0x201)	13
2.1.2 DiagResult (0x203)	16
2.1.3 Shutdown (0x204)	17
2.1.4 ShutdownSuspendMode (0x205)	18
2.1.5 SendMessage (0x207)	19
2.1.6 EchoMessage (0x208)	20
2.1.7 MessageBufSize (0x209)	21
2.1.8 Reset (0x211)	23
2.1.9 CentralRegistrySize (0x212)	24
2.1.10 MOSTRemoteReset (0x217)	25
2.1.11 PhysicalLayerTest (0x218)	26
2.1.12 PhysicalLayerTestResult (0x219)	28
2.1.13 ECLTrigger (0x220)	29
2.1.14 ECLInitiatorState (0x221)	31
2.1.15 Void (0x3C8)	33
2.1.16 DSIDSOCCount (0x3FD)	34
2.1.17 DSO (0x3FE)	35
2.1.18 DSI (0x400)	38

Bibliography

All documents, which this MOST document has references to, are listed here with the actual revision this document is referring to.

Number	Document	Revision
[1]	MOST Specification	3.0

Document History

Changes EnhancedTestability FBlock Rev. 3.0.3 to EnhancedTestability FBlock Rev. 3.0.4

Change Ref.	FktID	Changes
3V04-001	0x218	PhysicalLayerTest: <ul style="list-style-type: none"> Changed condition. Not implemented if FBlock 0x0A present. Removed MOSTPort parameter.
3V04-002	0x219	PhysicalLayerTestResult: <ul style="list-style-type: none"> Changed condition. Not implemented if FBlock 0x0A present. Removed MOSTPort parameter
3V04-003	0x3FE	DSO: Rephrased NextPacketMethod. Terminate description for clarity.

Changes EnhancedTestability FBlock Rev. 3.0.2 to EnhancedTestability FBlock Rev. 3.0.3

Change Ref.	FktID	Changes
3V03-001	General	<ul style="list-style-type: none"> In FBlock description, added remark regarding validity of parameters. (MCTG decision D176-5) Modified FBlock description: No values stored when disconnected from power. (MCTG decision D177-1)
3V03-002	0x044	Removed FBlockInfo from GeneralFBlock references. (MCTG action item D173-4)
3V03-003	0x201	AutoWakeup: <ul style="list-style-type: none"> Added clarification for Duration = 0. (MCTG decision D176-4) Made function conditional. (MCHP proposal) Removed OPTypes SetGet and Status. (MCHP proposal) Deleted "or a timeout of 5 min. after Netinterface Off." (MCTG decision D177-2)
3V03-004	0x203	Made function DiagResult conditional and modified description. (MCHP proposal)
3V03-005	0x204	Shutdown: <ul style="list-style-type: none"> Made function conditional and modified description. (MCHP proposal) Deleted If a "Start shutdown sequence" (Type = 0x00) is sent to a device, which is not PowerMaster, a "parameter not available" error message (code 0x07) shall be returned. (MCTG decision D177-2)
3V03-006	0x205	ShutdownSuspendMode: <ul style="list-style-type: none"> Changed occurrence to Conditional. (MCTG decision D177-2) Provided conditions for implementation. (WGDA decision D128_3) Removed OPTypes SetGet, Get, and Status. (WGDA decision D128_3) Replaced "device" with "node". (Spec. Support review)
3V03-007	0x207	SendMessage: Changed occurrence to Conditional. (MCTG decision D177-2)
3V03-008	0x208	EchoMessage: Deleted superfluous remark that the (mandatory) function has to be implemented in every node. (Spec. Support review)
3V03-009	0x209	MessageBufSize: Deleted superfluous remark that the (mandatory) function has to be implemented in every node. (Spec. Support review)
3V03-010	0x20F	Deleted function CodingErrors. (MCTG decision D175-1)
3V03-011	0x211	Reset: <ul style="list-style-type: none"> Reset is performed on the Network Interface Controller, not the NetInterface. (MCTG D165-1) Made function conditional. (MCHP proposal)
3V03-012	0x212	CentralRegistrySize: <ul style="list-style-type: none"> Made function conditional. (MCHP proposal)

Change Ref.	FktID	Changes
		– Changed to indicate the number of FBlocks that can be stored in the Central Registry. (MCTG decision D177-2)
3V03-013	0x213	Function NotificationMatrixSize deleted. (MCTG decision D177-2)
3V03-014	0x216	Deleted function SystemState . (MCTG decision D175-1)
3V03-015	0x218	Made function PhysicalLayerTest conditional and added MOSTPort parameter. (MCHP proposal)
3V03-016	0x219	Made function PhysicalLayerTestResult conditional and added MOSTPort parameter. (MCHP proposal)
3V03-017	0x220	ECLTrigger: – Determined which condition requires the implementation of this function. (MCTG action item A127-10) – Indicated what value the "not used" bits are set to. Added an example. (MCTG action item A144-15) – Changed description. (MCTG decision D177-2)
3V03-018	0x221	ECLInitiatorStatus: Determined which condition requires the implementation of this function. (MCTG action item A127-10) Modified description of ECLCommunicationStatus. (MCTG decision D164-3)
3V03-019	0x203	DiagResult: Added statement about availability of RBD result. (MCTG action item A133-4)
3V03-020	0x3FE	DSO: – Improved description of NextPacketMethod. (MCTG decision D151-5) – Made function conditional. (MCHP proposal) – Marked SingleFrameAcknowledge as deprecated. (MCTG review)
3V03-021	0x3C8	Void: Deleted misleading remark that the (mandatory) function must not to be implemented in any node. (Spec. Support review)
3V03-022	0x3FD	DSIDSOCCount: Deleted superfluous remark that the (mandatory) function has to be implemented in every node. (Spec. Support review)
3V03-023	0x3FF	Deleted function DSIHold. (MCTG decision D175-1)
3V03-024	0x400	DSI: Added test vectors. (MCTG decision D164-8)
3V03-025	0xF00	Deleted function Version. (MCTG decision D175-1)

Changes EnhancedTestability FBlock Rev. 3.0.1 to EnhancedTestability FBlock Rev. 3.0.2

Change Ref.	FktID	Changes
3V02-001	General	Filled the SymbolicName attribute for all Enums.
3V02-002	0x201	AutoWakeup: Modified description of parameter Duration.
3V02-003	0x217	MOSTRemoteReset: Changed from Mandatory to Optional.
3V02-004	0x203	ECLTrigger: – Added bit 0 (not used) to ECLTestParameters. – Changed description - ErrorInfo 0x02 instead of 0x03 for unsupported combinations.
3V02-005	0x3FE	DSO: Modified description so that "function busy" may be returned.

Changes EnhancedTestability FBlock Rev. 3.0 to EnhancedTestability FBlock Rev. 3.0.1

Change Ref.	FktID	Changes
3V01-001	General	Correction of typos and clerical errors.
3V01-002	0x201	AutoWakeup: – Modified description to reflect removal of PermissionToWake and CapabilityToWake – Renamed parameter Attenuation to Reserved – Modified diagrams to use "MOST Signal" instead of "light; modified signal levels"

Change Ref.	FktID	Changes
3V01-003	0x202	DiagTimeout: Function deleted
3V01-004	0x203	DiagResult: – Improved wording regarding the validity of the value – 0xFC indicates that RBD is inconclusive; removed 0xFD and 0xFE
3V01-005	0x204	Shutdown: – Renamed "Simulation of Shutdown" to "Simulation of over-temperature shutdown"; Renamed "Simulation of Dead" to "Simulation of critical temperature" – Made specific reference to MOST Specification revision and section more generic – Use "Start shutdown sequence (Type = 0x00)" instead of "Normal Shutdown"
3V01-006	0x205	ShutdownSuspendMode: Modified description of code 0x02 to indicate that the Enhanced-Testability FBlock does not affect the behavior of the application
3V01-007	0x207	SendMessage: Use LAMSmax instead of MaxAMSLength.
3V01-008	0x20E	ResetTests: Function deleted
3V01-009	0x20F	CodingErrors: Changed function signature, removed Hysteresis parameter
3V01-010	0x210	VoltageLevels: Function deleted
3V01-011	0x215	ActivateSlaveMode: Function deleted
3V01-012	0x218	PhysicalLayerTest: – Made requirement regarding time before power down more specific – Stated that this function is not relevant for ePhy
3V01-013	0x219	PhysicalLayerTestResult: Stated that this function is not relevant for ePhy
3V01-014	0x3FE	DSO: – Set range of TargetFktID to 0x400 – Added to description of parameter NetPacketMethod that ErrorCode 0x42 is sent if not supported – Added to parameter MHP_ConID that the source address of the request is used as target address
3V01-015	0x220	ECLTrigger: New function
3V0-016	0x221	ECLInitiatorState: New function

Changes EnhancedTestability FBlock Rev. 2.6.1 to EnhancedTestability FBlock Rev. 3.0

Change Ref.	FktID	Changes
3V0-001	General	Minor editorial corrections.
3V0-002	General	Substituted FunctionSection attribute with Occurrence attribute in all functions.
3V0-003	0x200	Removed function Attenuation.
3V0-004	0x201	Function AutoWakeup now has to be implemented by ePhy nodes that support RBD.
3V0-005	0x202	Function DiagTimeout now has to be implemented by ePhy nodes that support RBD.
3V0-006	0x203	Function DiagResult now has to be implemented by ePhy nodes that support RBD.
3V0-007	0x206	Removed function NetInterfaceState.
3V0-008	0x207	SendMessage: – Function becomes mandatory. – In the case of a dynamic buffer, the message must not exceed 4 segments.
3V0-009	0x214	Removed function ManufacturerTimings.
3V0-010	0x217	Added function MOSTRemoteReset.
3V0-011	0x218	Added function PhysicalLayerTest.
3V0-012	0x219	Added function PhysicalLayerTestResult.

Changes EnhancedTestability FBlock Rev. 2.6 to EnhancedTestability FBlock Rev. 2.6.1

Change Ref.	FktID	Changes
2V61-001	General	Minor editorial corrections.
2V61-002	0x000	Function FktIDs no longer referenced from GeneralFBlock.
2V61-003	0x203	Added description for code 0x40 to DiagResult.
2V61-004	0x3FD	Added new function DSIDSOCCount.

Changes EnhancedTestability FBlock 2V5 to EnhancedTestability FBlock 2V6

Change Ref.	FktID	Changes
2V6-001	General	General: Corrections of clerical errors and unification of spelling of MOST terms.
2V6-002	General	Added reference to GeneralFBlock 2.5.1, which covers FktIDs (0x000) and Version (0x010)
2V6-003	0x000	Removed function FktIDs
2V6-004	0x200	Function becomes optional. Not applicable for ePhy.
2V6-005	0x201	AutoWakeup: <ul style="list-style-type: none"> Function not used for ePhy Replaced AbilityToWake by CapabilityToWake resp. PermissionToWake Improved description of parameter "Duration" and "Delaytime": wakeup condition Figures added to explain "wakeup condition" and "timing"
2V6-006	0x202	DiagTimeout: Function not used for ePhy
2V6-007	0x203	DiagResult: Function not used for ePhy
2V6-008	0x204	Shutdown: Improved description in case neither PM nor temp. management supported.
2V6-009	0x206	Function becomes optional.
2V6-010	0x207	Function becomes optional.
2V6-011	0x209	MessageBufSize: <ul style="list-style-type: none"> Changed unit for MessageLengthRx and MessageLengthTx from none to Byte Description improved: consideration of "parallel message buffers"
2V6-012	0x20A	SendViaMHP: Deleted function.
2V6-013	0x20B	EchoViaMHP: Deleted function.
2V6-014	0x20C	MHPData: Deleted function.
2V6-015	0x20D	MamacPing: Deleted function.
2V6-016	0x211	Reset: <ul style="list-style-type: none"> Function not used for ePhy Description improved: Function has to reset NetInterface
2V6-017	0x214	ManufacturerTimings: Modified Enum description.
2V6-018	0x216	SystemState: New function.
2V6-019	0x3FE	DSO: New function.
2V6-020	0x3FF	DSIHold: New function.
2V6-021	0x400	DSI: New function.
2V6-022	0xF00	Marked Version as deprecated.
2V6-023	-	Removed empty "Dynamic Specification" chapter.

Changes EnhancedTestability FBlock 2V4 to EnhancedTestability FBlock 2V5

Change Ref.	Section	Changes
2V5-001	General	Using new MOST_document_template for better FCat consistency.
2V5-002	General	Change of erroneous Section types.
2V5-003	General	Revised descriptions.
2V5-004		Added Introduction regarding ErrorCode and ErrorInfo.
2V5-005		Added parameters Hysteresis and Timeout to function CodingErrors (0x20F). Changed description.
2V5-006		Added function ActivateSlaveMode (0x215).
2V5-007		Added function Void (0x3C8).

Changes EnhancedTestability FBlock 2V3 to EnhancedTestability FBlock 2V4

Change Ref.	Section	Changes
2V4-001	0x200	Inverted parameter Enabled.
2V4-002	0x201	Added parameter Duration for Set.
2V4-003	0x201	Added parameter Duration for SetGet.
2V4-004	0x201	Added parameter Duration for Status.
2V4-005	0x209	Changed description for parameter MessageLengthRx.
2V4-006	0x209	Changed description for parameter MessageLengthTx.
2V4-007	0x20D	Changed unit for parameter Timeout to ms.
2V4-008	0x20F	Added function CodingErrors.
2V4-009	0x210	Added function VoltageLevels.
2V4-010	0x211	Added function Reset.
2V4-011	0x212	Added function CentralRegistrySize.
2V4-012	0x213	Added function NotificationMatrixSize.
2V4-013	0x214	Added function ManufacturerTimings.
2V4-014	0xF00	Added function Version.

1 Introduction

A MOST Function Catalog is a collection of MOST Function Blocks (FBlocks).

This document contains the specification of an FBlock. MOST FBlocks are standardized and maintained by MOST workgroup Device Architecture (WG_DA). In order to speed up the process of making new FBlocks available, every FBlock will be updated individually as required.

2 Function Catalog

2.1 EnhancedTestability (FBlockID=0x0F)

This FBlock is used to trigger sequences which have to be tested in the MOST Compliance Test but which are normally triggered by a project specific, sometimes complicated, mechanism. Due to the nature of this FBlock neither notification nor processing messages will be implemented.

The FBlock should be initialized every time the NetOn state is reached. The FBlock is only available during NetOn. All properties are reset to their default state when entering NetOn, if not mentioned otherwise.

The functions in this FBlock describe a general interface for starting functionality partly implemented in the application, partly in the Network Service.

If an application callback returns wrong or unexpected values the FBlock sends a "device malfunction" error message (code 0x0B). For other defined values it is allowed that FBlock EnhancedTestability functions do not check the validity of parameters unless an error condition is explicitly stated in the function description.

FBlock EnhancedTestability philosophy: This FBlock generates or suppresses signals. It resides between application and the Netinterface. It should have no impact on the devices overall behavior, especially when the device is not under test condition. FBlock EnhancedTestability does not store any values beyond power down.

In addition to the functions contained in this document, the following functions are also part of the EnhancedTestability FBlock. They exist in the GeneralFBlock template and are included here by reference:

FktID	Function name
0x010	Version

Function Overview		
FktID	Name	Occurrence
0x201	AutoWakeup	Conditional
0x203	DiagResult	Conditional
0x204	Shutdown	Conditional
0x205	ShutdownSuspendMode	Conditional
0x207	SendMessage	Conditional
0x208	EchoMessage	Mandatory
0x209	MessageBufSize	Mandatory
0x211	Reset	Conditional
0x212	CentralRegistrySize	Conditional
0x217	MOSTRemoteReset	Optional
0x218	PhysicalLayerTest	Conditional
0x219	PhysicalLayerTestResult	Conditional
0x220	ECLTrigger	Conditional
0x221	ECLInitiatorState	Conditional
0x3C8	Void	Mandatory
0x3FD	DSIDSOCount	Mandatory
0x3FE	DSO	Conditional
0x400	DSI	Conditional

2.1.1 AutoWakeup (0x201)

Occurrence: Conditional

Condition: The AutoWakeup function has to be implemented in every node with the exception of ePhy nodes that do not support RBD.

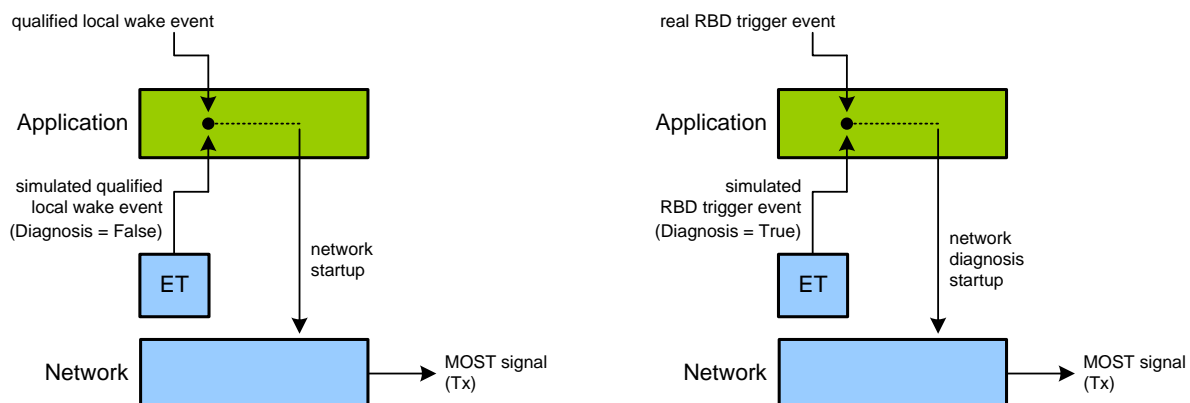
This function is used for triggering the device to wake-up the network via modulated signal. It covers two use cases: wake up with and without performing RBD.

Calling AutoWakeUp.Set configures the wake-up behavior. AutoWakeup gets valid with the most recent set of values on the next transition to NetInterface Off. It masks other wake-up signals of the application that might interfere.

This is valid until the next transition to NetInterface Normal Operation. Then the property is reset to its default state (and AutoWakeup is disabled.)

In case of Diagnosis = True, RBD is started only once, even if NetInterface Normal Operation is not reached. After the MOST network has been shutdown by the tester, the device will wakeup after the DelayTime (optionally using ring break diagnosis) as if it was triggered by an external event. The result of a diagnosis will be stored in the property DiagResult until it is either overwritten by the next result or the device is disconnected from power.

For further clarification, see the following figures:



If the parameter DelayTime equals zero, the AutoWakeup is disabled (other parameters are ignored).

The default state is DelayTime = 0, Diagnosis = False, Duration = 0.

In case of DelayTime > 0 and Diagnosis = False but the device is not designed for waking up the network, the device returns a "parameter not available" error message (code 0x07, 0x02, False). The property shall remain unchanged.

Diagnosis = False: Until next NetInterface NormalOperation state, all RBD triggers are ignored.

Diagnosis = True: Until next transition to NetInterface Normal Operation, RBD must be triggered once.

Default behavior: The application decides whether RBD is performed or not. The default behavior is reached on every transition to NetInterface Normal Operation.

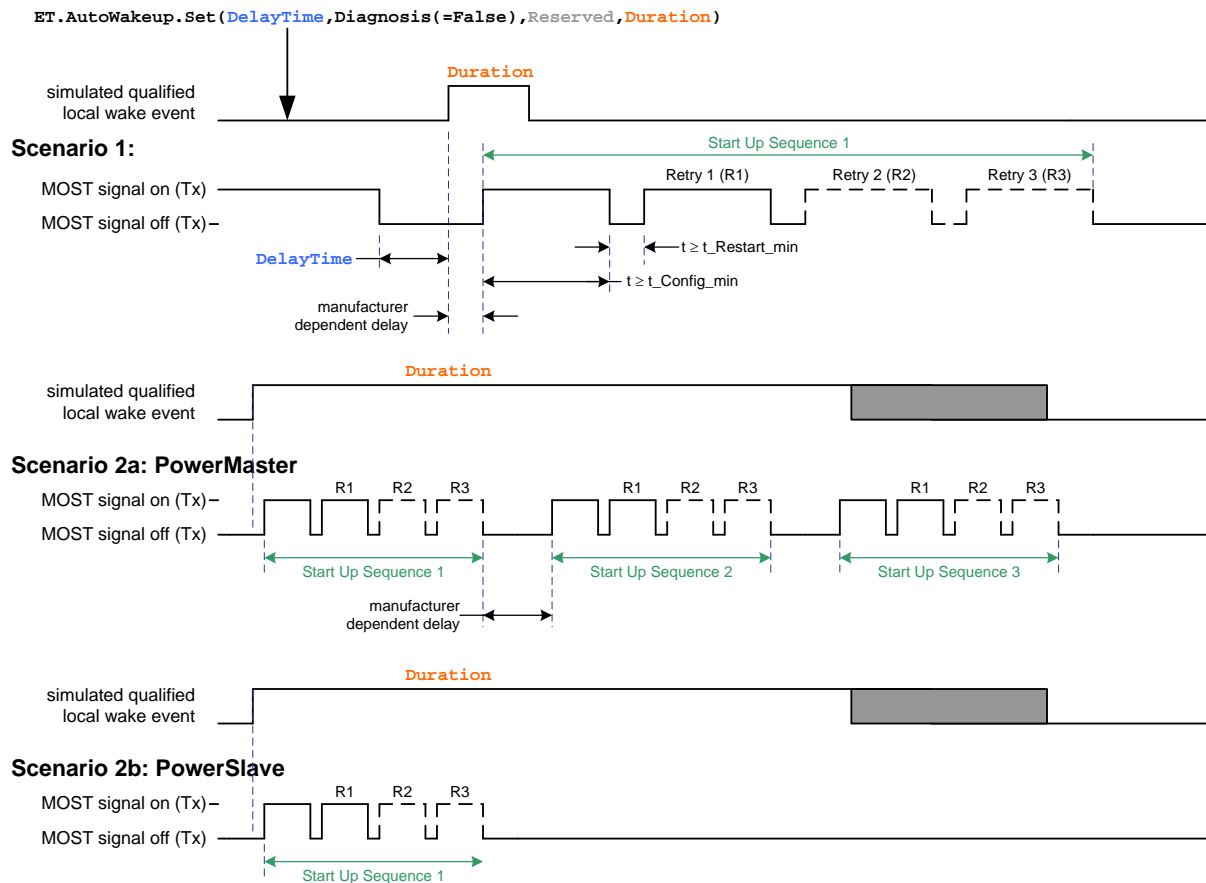
This behavior is valid until EnhancedTestability.AutoWakeUp is called.

The qualified local wakeup event simulation is illustrated in the figure below, where network startup occurs, with optional retries as needed (Scenario 1).

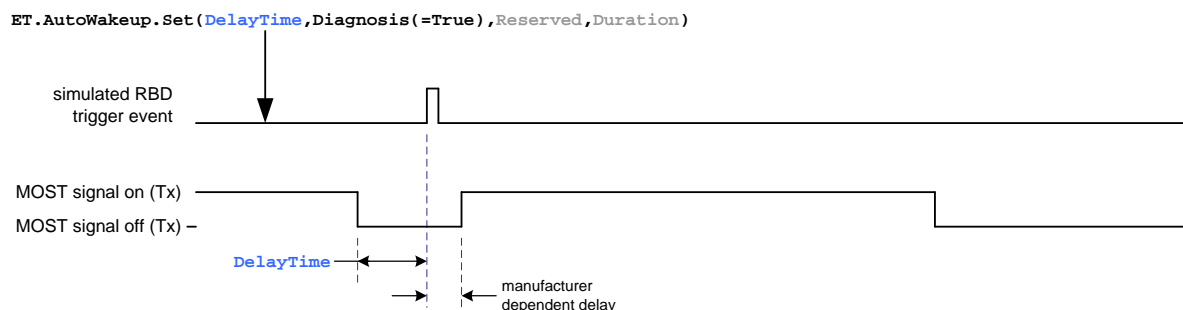
Scenario 2a for PowerMaster illustrates an extended simulated local wakeup event where the Duration

parameter extends beyond all normal retries; therefore, the DUT (PM) will continue to try and startup the network until the simulated local wakeup event ceases (expiration of Duration). In contrast, Scenario 2b for PowerSlave shows that there is only one sequence.

These scenarios illustrated below assume that NetInterface never achieves the NormalOperation state and that the network startup attempts are unsuccessful.



The RBD trigger event simulation is illustrated in the figure below, where the DUT initiates a diagnostic network startup after the DelayTime expires.



2.1.1.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	AutoWakeup (0x201)	Set	DelayTime , Diagnosis , Reserved , Duration
		Error	ErrorCode, ErrorInfo

2.1.1.2 Parameter

DelayTime

Time the device waits after reaching the NetInterface Off state before it will initiate the appropriate network startup event. A value of zero resets AutoWakeup to its default state (DelayTime = 0, Diagnosis = False), even if a prior call has not completed execution.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	s

Diagnosis

Determines the type of network startup, based on a simulated qualified wake up event, or a simulated RBD trigger event.

Basis data type	Bit #	Code	Description
Boolean	Bit 0	True	Simulated RBD trigger event. After DelayTime, assume a RBD trigger event occurred and initiate a network diagnosis startup.
		False	Simulated local wakeup event. After DelayTime, assume a qualified local wakeup event occurred and initiate network startup.

Reserved

Basis data type	Bit #	Code	Description
Boolean	Bit 0	True	-
		False	This parameter is always set to false.

Duration

For simulated local wakeup events (Diagnosis = False), this parameter determines how long the simulated wakeup event (the wakeup condition) is active.
The value of Duration = 0 is mapped to 255 s.

When Diagnosis is set to True, this parameter is ignored.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	s

2.1.2 DiagResult (0x203)

Occurrence: Conditional

Condition: DiagResult has to be implemented in nodes, where the result of RBD Phase 2 is evaluated.

The result from a preceding ring break diagnosis is stored in this property and can be accessed in a following network session.

The content of the property is stored over a NetOff phase as long as power is connected. The content changes with every new result of a running RBD. The content may get lost if the device is disconnected from power. Due to this special need the value of DiagResult cannot be initialized.

In other words: This value is only valid if ring break diagnosis has been run since the device was connected to power.

2.1.2.1 Format of Function

Function classes: Number

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	DiagResult (0x203)	Get	-
		Status	Result
		Error	ErrorCode, ErrorInfo

2.1.2.2 Parameter

Result

Result given by Network Service at the end of ring break diagnosis.

- A value from 0x00 to 0x3F indicates the relative position value in case of Ring-Break.
- A value of 0x40 (only relevant for oPhy) indicates signal at input but stable lock was never gained.
- A value of 0xFC indicates that ring break diagnosis is inconclusive.
- A value of 0xFF indicates a fully operational network.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	none

2.1.3 Shutdown (0x204)

Occurrence: Conditional

Condition: Shutdown has to be implemented in the PM node and nodes that implement temperature management.

Type = 0x00 only has to be supported by nodes containing the PowerMaster (PM).

Type = 0x01,0x02 only have to be supported by nodes which handle the temperature management.

This method is used to trigger different shutdown scenarios:

- * Normal shutdown (PowerMaster only)
- * Simulation of over-temperature shutdown
- * Simulation of critical temperature

Pending or new wake-up conditions have to be ignored until the system is shut down, if the method is called with parameter Type = 0.

All signals from the application within the DUT which prevent PM to shutdown are ignored.

The MOST message NetBlock.Shutdown.Result(Suspend) has to be respected.

2.1.3.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	Shutdown (0x204)	Start	Type
		Error	ErrorCode, ErrorInfo

2.1.3.2 Parameter

Type

If Type is 0x00 and the device is the PowerMaster it will start the shutdown sequence. Codes 0x01 and 0x02 refer to the over-temperature management as specified in the MOST Specification.

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	0x00...0x02	0x00	StartShutdownSequence	Start shutdown sequence if PowerMaster
		0x01	SimulateOverTempShutdown	Simulate over-temperature shutdown
		0x02	SimulateCriticalTemp	Simulate critical temperature

2.1.4 ShutdownSuspendMode (0x205)

Occurrence: Conditional

Condition: ShutdownSuspendMode has to be implemented in every node except PM and nodes that never send NetBlock.ShutDown.Result(Suspend).

This property can be set to On or Off. If set to "On" the node will respond with a report NetBlock.ShutDown.Result(Suspend) to each NetBlock.ShutDown.Start(Query) it receives. If set to "Off" no report NetBlock.ShutDown.Result(Suspend) will be sent. In "Default" mode (which is the default for this function), the application decides whether the sending of Suspend messages is required.

2.1.4.1 Format of Function

Function classes: Enumeration

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	ShutdownSuspendMode (0x205)	Set	Suspend
		Error	ErrorCode, ErrorInfo

2.1.4.2 Parameter

Suspend

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	0x00...0x02	0x00	SuspendOff	Off / Do not send Suspend
		0x01	SuspendOn	On / Always send Suspend
		0x02	SuspendDefault	Default behavior (i.e., FBlock EnhancedTestability does not change the behavior of the application)

2.1.5 SendMessage (0x207)

Occurrence: Conditional

Condition: SendMessage has to be implemented in every node that supports segmented messages.

The device will answer to this function call by sending an application message via Control Channel. The maximum length the device can use for sending should determine the length of the complete message. This should be indicated by the parameter MessageLengthTx of the MessageBufSize property.

In the case of a dynamic buffer (i.e., MessageLengthTx = 0), the message must not exceed 4 segments. The minimum length is LAMSmax + 1.

2.1.5.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	SendMessage (0x207)	StartResult	-
		Result	Data
		Error	ErrorCode, ErrorInfo

2.1.5.2 Parameter

Data

The payload of SendMessage.Result is a well-known pattern. The first data byte shall be 0x00; the second shall be 0x01 etc. The value is increased with every data byte. If 0xFF is reached the next data byte starts with 0x00 again.

Basis data type	Length	Condition	Description
Stream		-	

2.1.6 EchoMessage (0x208)

Occurrence: Mandatory

The device will answer to this request by sending back an application message via Control Channel that contains the same data as the request.

2.1.6.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	EchoMessage (0x208)	StartResult	Data
		Result	Data
		Error	ErrorCode, ErrorInfo

2.1.6.2 Parameter

Data

Arbitrary data that shall be echoed by the device under test. If the device can receive messages that are longer than the maximum length that can be transmitted, the echoed data will be truncated in such way that the first n bytes of the data are discarded. Discarding first n bytes is used for a test scenario where TxBuf < RxBuf.

Basis data type	Length	Condition	Description
Stream		-	

2.1.7 MessageBufSize (0x209)

Occurrence: Mandatory

Read only function to retrieve the number of message buffers available in parallel for message reception and their size in bytes from a device.

2.1.7.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	MessageBufSize (0x209)	Get	-
		Status	BufferCountRx , MessageLengthRx , BufferCountTx , MessageLengthTx , Shared
		Error	ErrorCode, ErrorInfo

2.1.7.2 Parameter

BufferCountRx

Maximum number of message buffers available in parallel for message reception. 0x0000 may be used to indicate that the number is not fixed (e.g., dynamic memory allocation).

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

MessageLengthRx

Maximum size of receivable application messages in Byte. This value counts only the payload bytes. Header information like FBlockID, InstID, Function, OPType, or additional information like MessageHandle, Status, Priority, Target address is not included. 0x0000 may be used to indicate that the length is not fixed (e.g., dynamic memory allocation).

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	Byte

BufferCountTx

Maximum number of message buffers available for sending. 0x0000 may be used to indicate that the number is not fixed (e.g., dynamic memory allocation).

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

MessageLengthTx

Maximum size of transmittable application messages in Byte. This value counts only the payload bytes. Header information like FBlockID, InstID, Function, OPType or additional information like MessageHandle, Status, Priority, Target address is not included. 0x0000 may be used to indicate that the length is not fixed (e.g., dynamic memory allocation).

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	Byte

Shared

Indicates that RX and TX buffers share the same memory pool.

Basis data type	Bit #	Code	Description
Boolean	Bit 0	True	RX and TX buffers share the same buffer pool.
		False	RX and TX own separate buffer pools.

2.1.8 Reset (0x211)

Occurrence: Conditional

Condition: Reset has to be implemented in every node except ePhy nodes.

The Reset function performs a reset of the Network Interface Controller.

2.1.8.1 Format of Function

Function classes: Trigger

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	Reset (0x211)	Start	-
		Error	ErrorCode, ErrorInfo

2.1.9 CentralRegistrySize (0x212)

Occurrence: Conditional

Condition: CentralRegistrySize has to be implemented in the node containing the NetworkMaster (NWM).

This Property indicates the maximum number of FBlocks that can be stored in the Central Registry. This Property is only used for devices with NetworkMaster functionality

2.1.9.1 Format of Function

Function classes: Number

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	CentralRegistrySize (0x212)	Get	-
		Status	Size
		Error	ErrorCode, ErrorInfo

2.1.9.2 Parameter

Size

Parameter Size describes the maximum number of FBlocks the NWM can store in his registry. The value 0 describes a dynamic size.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

2.1.10 MOSTRemoteReset (0x217)

Occurrence: Optional

The function configures the initialization respectively re-initialization of all MOST applications in the system.

Setting this property has its effect at next system start. In case the reset settings are 'Current' all user settings will be held (audio settings, telephone book, ...). In case settings have been set to 'Factory', the user settings are lost and the application resets to its default values. However, MOSTRemoteReset does not cause deletion of diagnostic error memory.

Note: To keep implementations of MOSTRemoteReset simple, the use of MOSTRemoteReset is restricted to situations where the application is able to write its persistent memory before shutdown. If the shutdown is forced by a power failure or due to critical temperature, MOSTRemoteReset may not have an effect.

2.1.10.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	MOSTRemoteReset (0x217)	SetGet	Settings
		Status	Settings
		Error	ErrorCode, ErrorInfo

2.1.10.2 Parameter

Settings

If Settings is 0x00, the application has to start up with all application data initialized to factory defaults. This does not apply to error memory though.

If Settings is 0x01, the application has to restore the current data at next system start up.

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	0x00...0x01	0x00	Factory	Factory
		0x01	Current	Current

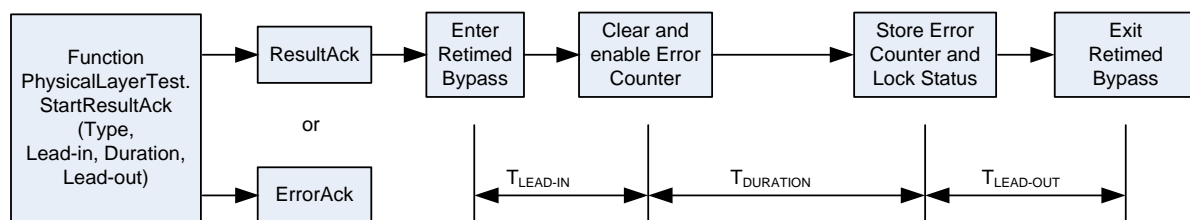
2.1.11 PhysicalLayerTest (0x218)

Occurrence: Conditional

Condition: PhysicalLayerTest has to be implemented in every node, except

- in ePhy nodes.
- if FBlock ExtendedNetworkControl (0x0A) is implemented.

Function is used during testing of the physical layer of a MOST150 device. During the time defined by the parameters passed to the function the network interface controller is in special (retimed) bypass mode and cannot perform MOST communication, although the received signal is still being locked upon (if possible) and frames containing coding errors being counted. The device must stay powered for (at least) tPwrSwitchOffDelay after exiting retimed bypass mode.



2.1.11.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	PhysicalLayerTest (0x218)	StartResultAck	SenderHandle , Type , Lead-in , Duration , Lead-out
		ErrorAck	SenderHandle , ErrorInfo
		ResultAck	SenderHandle

2.1.11.2 Parameter

SenderHandle

Unique identifier of the 'send'-task within the device.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

Type

If Type is 0x00, the device preserves its original device mode (TimingMaster/TimingSlave) during Retimed Bypass phase. Codes 0x01 and 0x02 refer to forced TimingMaster and TimingSlave Retimed Bypass Mode, respectively.

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	0x00...0x02	0x00	Auto	Auto (preserves original timing mode)
		0x01	RetimedBypassTimingMaster	Force Retimed Bypass TimingMaster mode

Basis data type	Range of values	Code	Symbolic Name	Description
		0x02	RetimedBypassTimingSlave	Force Retimed Bypass TimingSlave mode

Lead-in

Time from: Network interface controller entering Retimed Bypass Mode *until:* error counter reset and activated.

This time is needed to mask out any errors detected during the time the network interface controller and external devices need for switching over to Retimed Bypass Mode and ring stabilizes.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	ms

Duration

The value passed by this parameter defines the time during which physical layer tests can be performed. During that time, frames containing coding errors are actively being counted and unlocks detected. The coding error counter saturates to its maximum value if the number of frames containing coding errors is greater than that maximum. At the end of that time, the coding error counter value and the lock status flag are stored in the ErrorCounterValue and the LockStatus properties, respectively, and the coding error counter is disabled.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Long	0	50...4294967295	1	ms

Lead-out

Time from: error counter value being stored, *until:* Network interface controller switched to its original operation mode (TimingMaster/TimingSlave). This time is needed to compensate for time drifts between the network interface controller and external hardware and ensure safe switching back to original operation mode.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	ms

2.1.12 PhysicalLayerTestResult (0x219)

Occurrence: Conditional

Condition: PhysicalLayerTestResult has to be implemented in every node, except

- in ePhy nodes.
- if FBlock ExtendedNetworkControl (0x0A) is implemented.

Read only function to retrieve the result of the test of the physical layer initiated by function PhysicalLayerTest (0x218). The content of this property is stored as long as power is connected or function PhysicalLayerTest (0x218) is initiated anew.

Note: Sending *PhysicalLayerTestResult.Get* without prior *PhysicalLayerTest.StartResultAck* returns *OPType Error*.

2.1.12.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	PhysicalLayerTestResult (0x219)	Get	-
		Status	LockStatus , ErrorCounterValue
		Error	ErrorCode, ErrorInfo

2.1.12.2 Parameter

LockStatus

Value notifying presence of at least one unlock event during Physical Layer Test.

Basis data type	Bit #	Code	Description
Boolean	Bit 0	True	At least one unlock detected during the test
		False	No unlocks detected during the test

ErrorCounterValue

Contains the stored (at the end of the physical layer test) value of the coding error counter.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Long	0		1	none

2.1.13 ECLTrigger (0x220)

Occurrence: Conditional

Condition: The implementation of this function is required in an ECL wakeup initiator and an ECL system test initiator.

ECL (Electrical Control Line) is typically used when the system is not in NetInterface Normal Operation. Nevertheless, during compliance testing, the ECLTrigger function is applied to trigger the use of the ECL. Either an ECL wake-up or ECL system test can be performed. The Result is only returned after the initiated ECL action has been finished.

2.1.13.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	ECLTrigger (0x220)	StartResult	ECLAction , ECLData
		Result	-
		Error	ErrorCode, ErrorInfo

2.1.13.2 Parameter

ECLAction

This parameter identifies the action that is to be performed by the initiator.

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	0x00...0x02	0x00	Undefined	Not defined.
		0x01	WakeUp	Request wake-up from all ECL wake-up initiators.
		0x02	TestSequence	Request a particular test sequence from the ECL Test Initiator.

ECLData

Additional data that is required to perform the requested ECL action.

Basis data type	Length	Condition	Description
Stream		ECLAction = 0x01	
		ECLAction = 0x02	Content: ECLTestParameters

ECLTestParameters

The bits in this parameter correspond to the ECL test parameters P1-P5. In the case that the combination P1-P5 is not supported, Error (0x06, 0x02) is returned.

Example:

The "alive" test is defined as P1-P5 set to 10000, where P1 is the leftmost bit and P5 the rightmost bit.

According to the MOST Specification, Little Endian encoding is used. Thus, the ECLTestParameters byte contains the value 0x02:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0 (P5)	0 (P4)	0 (P3)	0 (P2)	1 (P1)	0

Basis data type	Bit #	Code	Description
Unsigned Byte	Bit 0	False	0
		True	not used
	Bit 1	False	P1 is not set.
		True	P1 is set.
	Bit 2	False	P2 is not set.
		True	P2 is set.
	Bit 3	False	P3 is not set.
		True	P3 is set.
	Bit 4	False	P4 is not set.
		True	P4 is set.
	Bit 5	False	P5 is not set.
		True	P5 is set.
	Bit 6	False	0
		True	not used
	Bit 7	False	0
		True	not used

2.1.14 ECLInitiatorState (0x221)

Occurrence: Conditional

Condition: The implementation of this function is required in an ECL wakeup initiator and an ECL system test initiator.

The last state of the ECL is stored in this property; it must not be initialized on a transition to NetInterface Normal Operation.

This property is used to retrieve the ECL test results that the initiator gathered from the participants.

2.1.14.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	ECLInitiatorState (0x221)	Get	ECLNodeClass
		Status	ECLCommunicationStatus , ECLAliveResult , ECLSignalResult
		Error	ErrorCode, ErrorInfo

2.1.14.2 Parameter

ECLNodeClass

The Electrical Control Line node class of a MOST device.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

ECLCommunicationStatus

ECLCommunicationStatus refers to the ECL System Test sequence.

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	0x00...0x02	0x00	Init	Init (default): reset to Init after power off.
		0x01	OK	SystemTest detected; written only at the end of the whole ECL System Test sequence.
		0x02	Invalid	Invalid, e.g. short circuit; written immediately when an error occurs.

ECLAliveResult

The Electrical Control Line alive result (En).

Basis data type	Bit #	Code	Description
Boolean	Bit 0	True	Device alive
		False	The device is not alive.

ECLSignalResult

The Electrical Control Line signal result (On). The interpretation of the signal result depends on the test parameters P1-P5.

Basis data type	Bit #	Code	Description
Boolean	Bit 0	True	1 reported
		False	0 reported.

2.1.15 Void (0x3C8)

Occurrence: Mandatory

Function 0x3C8 shall be kept empty without implementation as a place holder for test purposes, so that, e.g., a Broadcast Error Test with a message like "EnhancedTestability.InstID.0x3C8.SetGet" might be established.

2.1.15.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	Void (0x3C8)	Error	ErrorCode, ErrorInfo

2.1.16 DSIDSOCount (0x3FD)

Occurrence: Mandatory

Function has to be implemented in every node to indicate the maximum number of simultaneous MOST High Protocol connections (DSI / DSO).

In the case that no DSI is implemented, the parameter DSICount has to return "0x00". In the case that no DSO is implemented, the parameter DSOCCount has to return "0x00".

2.1.16.1 Format of Function

Function classes: Sequence Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	DSIDSOCount (0x3FD)	Get	-
		Status	DSICount , DSOCCount
		Error	ErrorCode, ErrorInfo

2.1.16.2 Parameter

DSICount

Maximum number of simultaneous (external) receiving MOST High Protocol connections (DSI).

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	none

DSOCCount

Maximum number of simultaneous (external) sending MOST High Protocol connections (DSO).

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	none

2.1.17 DSO (0x3FE)

Occurrence: Conditional

Condition: DSO has to be implemented in every node with Data source (DSO) functionality (MOST High sources).

The device will answer to this function call by opening a connection with MHP via the Packet Data Channel (target: TargetFBlockID, TargetInstID, TargetFktID and TargetOPType). The device has to send dummy data via MOSTHigh using suitable PacketSize and a well-known pattern. The first data byte shall be 0x00, the seconds shall be 0x01, etc. The value is increased with every data byte. If 0xFF is reached the next data byte starts with 0x00 again.

If the maximum number of connections is reached or there is already a connection, the method may return a 'function busy' (error code 0x40) error message. If a 'device malfunction' (error code 0x0B) error message is sent, there was not enough memory available to allocate the buffers (the application callback returned no memory). The operations StartAck, and ErrorAck of this method are transmitted via the Control Channel. "Prio" and "RevID" have to be chosen by DUT. The tester cannot give a priority to DUT.

2.1.17.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	DSO (0x3FE)	StartAck	SenderHandle , NumPackets , NextPacketMethod , AckMode , MHP_ConID
		ErrorAck	SenderHandle , ErrorCode, ErrorInfo

2.1.17.2 Parameter

SenderHandle

Unique identifier of the 'send'-task within the device.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

NumPackets

Number of MOSTHigh packets that shall be sent. This is used for long-term and performance tests, as well as for testing the protocol handshaking, like the blocks hand-over and MOSTHigh connection establishing procedure.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

NextPacketMethod

Determine what to do after a TX_SUCCESS event.
In NextPacketMethod "Hold" and "SendNext", every packet is sent in the same connection. In NextPacketMethod "Terminate", every packet is sent in a separate connection.

If NextPacketMethod=0x02 is not supported, ErrorAck is sent, using ErrorCode 0x42.

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	0x00...0x02	0x00	Hold	Normal HOLD phase (will close after timeout).
		0x01	Terminate	Terminate the connection and open a new one if further data needs to be transmitted.
		0x02	SendNext	Try to send next packet without a hold (optional).

AckMode

Determine acknowledge method

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	0x00...0x01	0x00	SingleFrameAcknowledge	use single frame acknowledge (deprecated)
		0x01	BlockAcknowledge	use block acknowledge

MHP_ConID

Determine target FBlockID, InstID, FktID and CmdOPType for MHP connection. The source address of the request is used as target address.

Basis data type	Length	Condition	Description
Stream		-	Content: TargetFBlockID , TargetInstID , TargetFktID , TargetOPType

TargetFBlockID

Target FBlockID for MHP connection

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte	0	0xF...0xF	1	none

TargetInstID

Target InstID for MHP connection

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	none

TargetFktID

Target FktID for MHP connection

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word	0	0x400...0x4FF	1	none

TargetOPType

Target OPType (only commands allowed) for MHP connection.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte	0	0x0...0x8	1	none

2.1.18 DSI (0x400)

Occurrence: Conditional

Condition: DSI has to be implemented in every node with DSI functionality (MOSTHigh sinks).

This function has to be called with MHP via Packet Data Channel. The device will answer to this function call with result, transmitted via Control Channel (no MHP used). With the result, the device has to return the checksum of the data, received during the function call via MHP).

Note: For checksum calculation, use CRC32 algorithm (according to IEEE 802.3).
This is a standard cyclic redundancy check classified by the following properties:

- CRC polynomial is 0x04C11DB7 (bit sequence of coefficients)
- CRC is 32bit wide
- Initial value is 0xFFFFFFFF
- Result must be XOR-ed to 0xFFFFFFFF
- Reversing bit order on incoming values
- Reversing bit order on resulting data

The checksum must be calculated highest byte of Data first.
The following test vectors can be used:

No.	Data	DataLength	crc32
a	{0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39}	9	0xCB43926
b	{0xEF, 0x29, 0x1B, 0x81, 0x56, 0x38, 0xC7, 0xD8}	8	0x48CD00A8
c	{0x00, 0x00, 0x00}	3	0xFF41D912
d	{}	0	0

The FBlock EnhancedTestability has to provide as many 'copies' of function 'DSI' as the DUT supports simultaneous MOST High connections (as stated in the DUT Manufacturer Information List). This is necessary to check whether the DUT is able to handle the maximum amount of connections. For every connection, one FktID is used. The function DSI and its copies have to be implemented in contiguous FktID range starting from 0x400 to (at max.) 0x4FF.

2.1.18.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	DSI (0x400)	StartResultAck	SenderHandle , Data
		ErrorAck	SenderHandle , ErrorCode, ErrorInfo
		ResultAck	SenderHandle , crc32 , DataLength

2.1.18.2 Parameter

SenderHandle

Unique identifier of the 'send'-task within the device.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

Data

Arbitrary data that shall be used by the device to calculate the checksum.

Basis data type	Length	Description
Stream		

crc32

Checksum of received data

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Long	0		1	none

DataLength

Length (in byte) of received data

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Long	0		1	Byte

Notes: