

# Freescall Serial Communication Interface (FSCI) for IEEE 802.15.4 MAC/PHY

## Contents

### 1. Introduction

This reference manual provides a detailed description of the 802.15.4 MAC/PHY serial commands, communication packet structure, available services and usage.

#### 1.1. Scope and objective

This document exposes MAC layer functionality through a serial communication interface to a host system. The typical serial interfaces used by the Freescale MAC/PHY black boxes are UART, SPI and I<sup>2</sup>C. The typical applications employing MAC/PHY FSCI are host systems such as a PC or an embedded system that has an upper layer (i.e. network) implementation which interacts with the 802.15.4 MAC functionality.

#### 1.2. Audience description

This reference manual is intended for application designers and users of the 802.15.4 MAC/PHY FSCI.

1.	Introduction.....	1
1.1.	Scope and objective .....	1
1.2.	Audience description .....	1
1.3.	Definitions, acronyms, and abbreviations .....	2
2.	Overview.....	2
2.1.	MAC/PHY test client architecture .....	3
2.2.	MAC interface set .....	3
2.3.	SAP handler and API .....	4
2.4.	Basic PC requirements .....	4
2.5.	Overview and packet structure.....	5
3.	FSCI messages description .....	6
3.1.	Reserved operation group codes .....	6
3.2.	General FSCI messages .....	7
3.3.	MAC 2006 messages .....	27
3.4.	ASP messages .....	55
4.	References.....	69
5.	Revision history .....	69

## 1.3. Definitions, acronyms, and abbreviations

Table 1. Acronyms

Term/Acronym	Definition
ARM®	Advanced RISC Machine (computer processor)
BE	Backoff Exponent
CCA	Clear Channel Assessment
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
DSN	Data Sequence Number
LQI	Link Quality Indication
MAC	Medium Access Control Layer
MCPS	MAC Common Part Sublayer
MCPS-SAP	MAC Common Part Sublayer Service Access Point
MFR	MAC Footer
MHR	MAC Header
MLME	MAC sublayer Management Entity
MLME-SAP	MAC sublayer Management Entity Service Access Point
MPDU	MAC protocol data unit
MSDU	MAC service data unit
PAN	Personal Area Network
PD	PHY data
PD-SAP	PHY data service access point
PHY	Physical Layer
PIB	PAN information base
PLME	Physical Layer Management Entity
PLME-SAP	Physical Layer Management Entity Service Access Point
PPDU	PHY Protocol Data Unit
PSDU	PHY Service Data Unit
RTOS	Real-Time Operating System
SAP	Service Access Point
SPDU	SSCS protocol data units
SSCS	Service-specific convergence sublayer

## 2. Overview

The Freescale Serial Communication Interface (FSCI) is at the same time a software module and a protocol that allows monitoring and extensive testing of the protocol layers interfaces. It also allows separation of the protocol stack between two protocol layers in a two processing entities setup: the host

processor (typically running the upper layers of a protocol stack) and the Black Box application (typically containing the lower layers of the stack, serving as a modem).

The Freescale Test Tool software is an example of a host processor which can interact with the FSCI Black Boxes at various layers. In this setup a user can run numerous commands to test the Black Box application services and interfaces.

The FSCI enables common service features for each device and allows monitoring of specific interfaces and API calls. Additionally, the FSCI injects or calls specific events and commands into the interfaces between layers. The layers communicate by sending primitives via service access points (SAP). The FSCI permits the developer to test specific SAP handlers and SAPs.

The architecture builds on the OSI Seven-Layer model, ensuring inter-operability between network devices. In the MAC\_FSCI\_App implementation, the IEEE 802.15.4 stack provides the physical (PHY) and media access control (MAC) layers.

An entity which needs to be interfaced to the FSCI module can use the API to register op codes to specific interfaces. After doing so, any packet coming from that interface with the same op code will trigger a callback execution. Two or more entities cannot register the same op code on the same interface, but they can do so for different interfaces. For example two MAC instances can register the same op codes, one over UARTA and the other over UARTB. This way we can use Test Tool to communicate to each MAC layer over two UART interfaces.

## NOTE

The FSCI module executes in the context of the Serial Manager task.

## 2.1. MAC/PHY test client architecture

The FSCI is a small application running separate of each layer in the stack, whether that node is a Coordinator or End Device. The host PC or host processor connects to the device under test (DUT) via a USB, UART, RS-232 cable SPI, or I<sup>2</sup>C (depending on the board type) in serial mode. The device can then be controlled by API calls generated by the host to test the interfaces between layers or implement an application on the host CPU using the FSCI protocol.

The FSCI enables common service features for each device and allows monitoring of specific interfaces and API calls. Additionally, the FSCI injects or calls specific events and commands into the interfaces between layers.

## 2.2. MAC interface set

The protocol suite provides the building blocks, or layers, that drive the functionality of an 802.15.4 network. Designed to support a wireless sensor network, the network exposes each layer to the next through a service access point (SAP).

Data entities provide data transport services between layers, while the management entities handle management services. For example, the MAC Common Part Sublayer (MCPS) provides data services through the MCPS-SAP. The MAC Sublayer Management Entity (MLME) maintains a database of managed objects and provides management services through the MLME-SAP.

Each SAP provides primitives that an upper layer uses to access services provided by the lower layers. Through these primitives, one layer can request information from another layer, and that layer can confirm in response, returning an indication when required.

A SAP or SAP handler, depending on the direction, passes data or manages data passed to it. SAPs are implemented as functions in an application, although only the request can be called directly.

This manual includes a detailed but partial list of the primitives used in the testing procedures.

## 2.3. SAP handler and API

The FSCI utilizes a series of commands enabling the host PC or processor to make API calls to SAP handlers and SAPs. Those control modes, selected as parameters in the software, include:

<b>Disable Mode</b>	Ignores the layer when running the tests.
<b>Hook Mode</b>	Allows the FSCI to exclusively control a specific layer. The FSCI hooks only SAPs that receive messages from a lower layer. In this mode, the FSCI replaces the layer whose SAP is hooked, and returns, but does not process, messages sent to a specific SAP. This mode should not be used with the Black Box Application and can in general cause unexpected behavior as the upper layer can no longer process indications from lower layer.
<b>Monitor Mode</b>	Allows the FSCI to capture all messages received by a SAP. This feature does not impact the flow of message between layers. When in monitor mode, the FSCI receives all messages except those disabled at compile time and those filtered out.

### NOTE

Monitoring a large number of SAPs can cause serial overflow and potentially disable test network devices. Monitor only the SAP calls required, usually only one or two at a time.

Depending on the compile-time options used, some SAPs or commands may not be available.

## 2.4. Basic PC requirements

- IAR Embedded Workbench® for ARM
- Freescale Test Tool software installed on host PC
- Freescale development boards for testing
- Optional wireless sniffer for testing

A complete environment requires the following

- MAC FSCI application loaded on to the appropriate development board or module using Freescale 802.15.4 capable chip sets
- Freescale test tool or another host processor capable of communicating through UART, USB, SPI or I<sup>2</sup>C

For more information about Test Tool, see the *Freescale Test Tool User's Guide*.

## 2.4.1. Interface description

The FSCI uses the SerialManager, and use one of the following interfaces for communications:

- A 3-wire UART connection
- A 2-wire I<sup>2</sup>C connection
- A 4-wire SPI connection
- A USB connection

## 2.5. Overview and packet structure

The FSCI module sends and receives messages as shown in the figure below. This structure is not specific to a serial interface and is designed to offer the best communication reliability. The Black Box device is expecting messages in little-endian format and responds with messages in little-endian format.

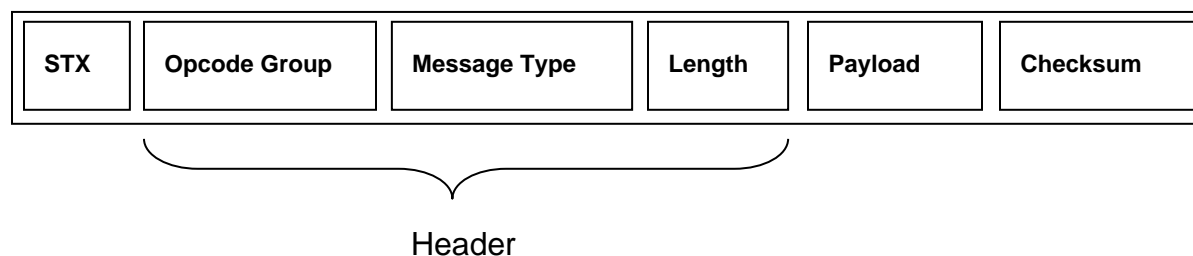


Figure 1. Packet structure

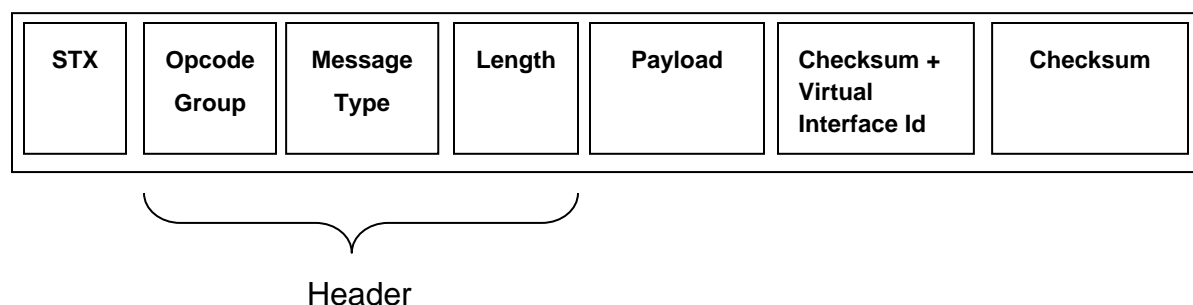


Figure 2. Packet structure when virtual interfaces are used

**Table 2. Packet field description**

Field Name	Length (bytes)	Description
STX	1	Used for synchronization over the serial interface. The value is always 0x02.
Opcode Group	1	Distinguishes between different Service Access Primitives (e.g. MLME or MCPS)
Message Type	1	Specifies the exact message opcode that is contained in the packet
Length	1/2	The length of the packet payload, excluding the header and FCS. The length field content shall be provided in little endian format.
Payload	variable	Payload of the actual message.
Checksum	1	Checksum field used to check the data integrity of the packet.
Checksum2	0/1	The second CRC field appears only for virtual interfaces

**NOTE**

When virtual interfaces are used, the first checksum is incremented with the Id of the interface, and the second checksum is used for error detection.

### 3. FSCI messages description

This chapter describes all MAC/PHY commands (requests) and events (confirmations, indications and monitored requests) in detail. The `MAC2006.XML` and `MAC2006_g.XML` documents found in the Freescale Test Tool directory (in the XML folder) provide an XML representation of all these commands.

**NOTE**

The commands listed in the `MAC2006.XML` uses 1 byte to represent the length of the payload by default, while the commands listed in `MAC2006_g.XML` uses 2 bytes for the length field by default.

#### 3.1. Reserved operation group codes

The following table illustrates the Operation Group codes (OG) used by the MAC FSCI.

**Table 3. Reserved OG codes**

Operation Group	Description
0xA3	General FSCI Request
0xA4	General FSCI Confirm/Indication
0x85	802.15.4 MLME Request
0x84	802.15.4 MLME Confirm/Indication
0x87	802.15.4 MCPS Request
0x86	802.15.4 MCPS Confirm/Indication
0x95	ASP Request
0x94	ASP Confirm/Indication

## 3.2. General FSCI messages

### 3.2.1. FSCI-AllowDeviceToSleep.Request

#### Description

Allow the device to enter deep sleep modes.

The possible wake-up sources are: UART module interrupt or radio timer timeout interrupt. The radio timer timeout time (deep sleep duration) is configurable.

#### Parameters

**Table 4. FSCI-AllowDeviceToSleep.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x40
Length	1/2	Length in bytes of the following parameters
Signal Host When Wake Up	1	Enable or disable of sending a wake-up indication to the host after wake-up from low-power mode (after the deep sleep duration expired or after any byte received over the UART from host).
Deep Sleep Duration	4	Deep sleep duration in 802.15.4 PHY symbols (16us). Default value is 5 s
Deep Sleep Mode	1	Platform dependent! Only for platforms where the Deep Sleep Mode is configurable at runtime

### 3.2.2. FSCI-AllowDeviceToSleep.Confirm

#### Description

Allow device to sleep confirmation.

#### Parameters

**Table 5. FSCI-AllowDeviceToSleep.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x40
Length	1/2	Length in bytes of the following parameters
Status	1	Possible values: 0x00: gSuccess (Success) 0xFF: gError (Error)

### 3.2.3. FSCI-CPUReset.Request

#### Description

Reset CPU. No confirm is received.

#### Parameters

**Table 6. FSCI-CPUReset.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x08
Length	1/2	0x00 - This message does not have any parameters

### 3.2.4. FSCI-Error.Event

#### Description

FSCI is reporting an error condition.

#### Parameters

**Table 7. FSCI-Error.Event parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0xFE
Length	1/2	Length in bytes of the following parameters
Status	1	Which error occurred. Possible values: 0x00: gSuccess_c (Should not be seen in this event.) 0xF0: gFSCISAPDisabled_c (Self-explanatory.) 0xF1: gFSCISAPInfoNotFound_c (Self-explanatory.) 0xF2: gFSCIUnknownPIB_c (Self-explanatory.) 0xF3: gFSCIAppMsgTooBig_c (A data item to be set or retrieved is too big for the buffer available to hold it) 0xF4: gFSCIOutOfMessages_c (FSCI tried to allocate a message, but the allocation failed.) 0xF5: gFSCIEndPointTablesFull_c (Self-explanatory.) 0xF6: gFSCIEndPointNotFound_c (Self-explanatory.) 0xF7: gFSCIUnknownOpcodeGroup_c (FSCI does not recognize the opcode group, and there is no application hook.) 0xF8: gFSCIOpcodeGroupsDisabled_c (FSCI support for an opcode group is turned off by a compile option.) 0xF9: gFSCIDebugPrintFailed_c (An attempt to print a debug message ran out of buffer space.) 0xFA: gFSCIReadOnly_c (Attempt to set read-only data.) 0xFB: gFSCIUnknownIdentifier_c (Self-explanatory.) 0xFC: gFSCIRequestIsDisabled_c (FSCI support for an opcode is turned off by a compile option.) 0xFD: gFSCIUnknownOpcode_c (Self-explanatory.) 0xFE: gFSCITooBig_c (A data item to be set or retrieved is too big for the buffer available to hold it.) 0xFF: gFSCIError_c (Non-specific, catchall error code.)



### 3.2.5. FSCI-GetLastReceivedPacketLqiValue.Request

#### Description

Get the LQI value of the last received packet.

#### Parameters

**Table 8. FSCI-GetLastReceivedPacketLqiValue.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x2D
Length	1/2	0x00 - This message does not have any parameters

### 3.2.6. FSCI-GetLastReceivedPacketLqiValue.Confirm

#### Description

Returns the LQI value of the last received packet.

#### Parameters

**Table 9. FSCI-GetLastReceivedPacketLqiValue.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x2D
Length	1/2	Length in bytes of the following parameters
LQI	1	The LQI value

### 3.2.7. FSCI-GetMode.Request

#### Description

Get FSCI mode select settings.

#### Parameters

**Table 10. FSCI-GetMode.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x02
Length	1/2	0x00 - This message does not have any parameters

### 3.2.8. FSCI-GetMode.Confirm

#### Description

Returns mode select settings currently used.

#### Parameters

**Table 11. FSCI-GetMode.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x02
Length	1/2	Length in bytes of the following parameters
Status	1	Possible values: 0x00: gSuccess (Success) 0xFF: gInvalidRequest_c (Internal error; should never happen.)
UART Tx Blocking	1	Enable UART driver blocking on Tx
SAP Handler Modes	Variable	Disable, Monitor, or Hook for each SAP handler group Structure type parameter. See detailed table below for parameter structure.

**Table 12. SAP handler modes parameter structure**

Parameter	Size (bytes)	Comments
MCPS	1	MCPS Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode) 0xFF: NotSupported (FSCI support for these SAP Handlers has been turned off by a compile switch)
MLME	1	MLME Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode) 0xFF: NotSupported (FSCI support for these SAP Handlers has been turned off by a compile switch)
ASP	1	ASP Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode) 0xFF: NotSupported (FSCI support for these SAP Handlers has been turned off by a compile switch)

Table 12. SAP handler modes parameter structure

Parameter	Size (bytes)	Comments
NLDE	1	NLDE Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode) 0xFF: NotSupported (FSCI support for these SAP Handlers has been turned off by a compile switch)
NLME	1	NLME Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode) 0xFF: NotSupported (FSCI support for these SAP Handlers has been turned off by a compile switch)
APSD	1	APSD Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode) 0xFF: NotSupported (FSCI support for these SAP Handlers has been turned off by a compile switch)
APSME	1	APSME Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode) 0xFF: NotSupported (FSCI support for these SAP Handlers has been turned off by a compile switch)
AFDE	1	AFDE Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode) 0xFF: NotSupported (FSCI support for these SAP Handlers has been turned off by a compile switch)
ZDP	1	ZDP Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode) 0xFF: NotSupported (FSCI support for these SAP Handlers has been turned off by a compile switch)
InterPan	1	InterPan Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode) 0xFF: NotSupported (FSCI support for these SAP Handlers has been turned off by a compile switch)

### 3.2.9. FSCI-GetWakeUpReason.Request

#### Description

Get the Wake Up reason.

#### Parameters

**Table 13. FSCI-GetWakeUpReason.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x72
Length	1/2	0x00 - This message does not have any parameters

### 3.2.10. FSCI-GetWakeUpReason.Confirm

#### Description

Get the wake up reason.

#### Parameters

**Table 14. FSCI-GetWakeUpReason.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x72
Length	1/2	0x00 - This message does not have any parameters
Wake Up Reason	1	Bitfield. Platform specific!

### 3.2.11. FSCI-ModeSelect.Request

#### Description

Sets FSCI Sap mode select configuration for each SAP.

#### Parameters

**Table 15. FSCI-ModeSelect.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x00
Length	1/2	Length in bytes of the following parameters
UART TX Blocking	1	Enable UART driver blocking on TX
MCPS	1	MCPS Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode)

Table 15. FSCI-ModeSelect.Request parameters

Parameter	Size (bytes)	Comments
MLME	1	MLME Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode)
ASP	1	ASP Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode)
NLDE	1	NLDE Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode)
NLME	1	NLME Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode)
APSD	1	APSD Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode)
APSME	1	APSME Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode)
AFDE	1	AFDE Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode)
ZDP	1	ZDP Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode)
InterPan	1	InterPan Mode Possible values: 0x00: DisableMode (Disable Mode) 0x01: HookMode (Hook Mode) 0x02: MonitorMode (monitor Mode)

### 3.2.12. FSCI-ModeSelect.Confirm

#### Description

FSCI-ModeSelect request Confirmation.

#### Parameters

**Table 16. FSCI-ModeSelect.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x00
Length	1/2	Length in bytes of the following parameters
Status	1	Possible values: 0x00: gSuccess (Success) 0xFF: gInvalidRequest_c (Wrong number of mode settings, or invalid mode value.)

### 3.2.13. FSCI-OTASupportCancelImage.Request

#### Description

Terminate the upgrade process (error occurred or download process was stopped)

#### Parameters

**Table 17. FSCI-OTASupportCancelImage.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x2C
Length	1/2	0x00 - This message does not have any parameters

### 3.2.14. FSCI-OTASupportCancelImage.Confirm

#### Description

Confirmation of the FSCI OTA cancel image request.

#### Parameters

**Table 18. FSCI-OTASupportCancelImage.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x2C
Length	1/2	Length in bytes of the following parameters
Status	1	Possible values: 0x00: gSuccess (Success)

### 3.2.15. FSCI-OTASupportCommitImage.Request

#### Description

Mark the download process as completed, and set a bitmap of flash sectors to be updated. A "0" bit means that the sector corresponding to the position of the bit will not be altered. A "1" bit means that the sector corresponding to the position of the bit will be erased, and possibly written by the new image.

#### Parameters

**Table 19. FSCI-OTASupportCommitImage.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x2B
Length	1/2	Length in bytes of the following parameters
Bitmap	32	Bitmap

### 3.2.16. FSCI-OTASupportCommitImage.Confirm

#### Description

Confirmation of the FSCI OTA commit image request.

#### Parameters

**Table 20. FSCI-OTASupportCommitImage.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x2B
Length	1/2	Length in bytes of the following parameters
Status	1	Possible values: 0x00: gSuccess (Success) 0x05: gOtaInvalidParam_c (Invalid Parameters) 0x06: gOtaInvalidOperation_c (Invalid Operation) 0x07: gOtaExternalFlashError_c (External Memory Error)

### 3.2.17. FSCI-OTASupportImageChunk.Request

#### Description

Request a block of OTA Image from the PC Application

#### Parameters

**Table 21. FSCI-OTASupportImageChunk.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x2F
Length	1/2	Length in bytes of the following parameters
Deviceld	2	Device ID (short address - in almost all cases)

ImageOffset	4	Image Offset
ImageBlockLength	1	Block Length

### 3.2.18. FSCI-OTASupportImageNotify.Request

#### Description

Test if the image is already in the external memory and if the new image can be used to upgrade the device (store image in the external memory or/and notify over the air - dongle solution-).

#### Parameters

**Table 22. FSCI-OTASupportImageNotify.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0xC4
Length	1/2	Length in bytes of the following parameters
DeviceId	2	Device ID (short address - in almost all cases)
ManufacturerCode	2	Manufacturer Code
ImageType	2	Image Type
FileVersion	4	File Version

### 3.2.19. FSCI-OTASupportImageNotify.Confirm

#### Description

Confirmation of the FSCI OTA image notify request.

#### Parameters

**Table 23. FSCI-OTASupportImageNotify.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0xC4
Length	1/2	Length in bytes of the following parameters
Status	1	Possible values: 0x00: gSuccess (Success) 0x01: gZclFailure_c (Failure)

### 3.2.20. FSCI-OTASupportPushImageChunk.Request

#### Description

Receive a block of Image from PC Application

#### Parameters

**Table 24. FSCI-OTASupportPushImageChunk.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3



OpCode	1	0x2A
Length	1/2	Length in bytes of the following parameters
DataImageBlock	1 x Length	Data image block

### 3.2.21. FSCI-OTASupportPushImageChunk.Confirm

#### Description

Confirmation of the FSCI OTA push image chunk request.

#### Parameters

**Table 25. FSCI-OTASupportPushImageChunk.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x2A
Length	1/2	Length in bytes of the following parameters
Status	1	Possible values: 0x00: gSuccess (Success) 0x05: gOtaInvalidParam_c (Invalid Parameters) 0x06: gOtaInvalidOperation_c (Invalid Operation) 0x07: gOtaExternalFlashError_c (External Memory Error)

### 3.2.22. FSCI-OTASupportQueryImage.Request

#### Description

Request a minimal set of information about the image. A value of 0xFFs has a special meaning of a wild card. The value has a 'match all' effect on the PC App.

#### Parameters

**Table 26. FSCI-OTASupportQueryImage.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0xC2
Length	1/2	Length in bytes of the following parameters
DeviceId	2	Device ID (short address - in almost all cases)
ManufacturerCode	2	Manufacturer Code
ImageType	2	Image Type
FileVersion	4	File Version

### 3.2.23. FSCI-OTASupportQueryImageRsp.Request

#### Description

This message is sent by the host application (PC application) to give details about the new image to the device.

#### Parameters

**Table 27. FSCI-OTASupportQueryImageRsp.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0xC3
Length	1/2	Length in bytes of the following parameters
OtaImageStatus	1	OTA image Status Possible values: 0x00: gOtaSucess_c (Success) 0x01: gOtaNoImage_c (No Image Available)
DeviceId	2	Device ID (short address - in almost all cases)
ManufacturerCode	2	Manufacturer Code
ImageType	2	Image Type
FileVersion	4	File Version
TotalImageSize	4	Total Image Size

### 3.2.24. FSCI-OTASupportQueryImageRsp.Confirm

#### Description

Returns the status of the upgrade image query.

#### Parameters

**Table 28. FSCI-OTASupportQueryImageRsp.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0xC3
Length	1/2	Length in bytes of the following parameters
Status	1	Possible values: 0x00: gOtaSucess_c (Success) 0x01: gOtaNoImage_c (No Image Available)

### 3.2.25. FSCI-OTASupportSetMode.Request

#### Description

Set the operation mode

#### Parameters

**Table 29. FSCI-OTASupportSetMode.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x28
Length	1/2	Length in bytes of the following parameters
OperationMode	1	Operation Mode Possible values: 0x00: gUpgradeImageOnCurrentDevice_c (Upgrade Image On Current Device) 0x01: gUseExternalMemoryForOtaUpdate_c (Use External Memory for OTA Upgrade Process) 0x02: gDoNotUseExternalMemoryForOtaUpdate_c (Do not use External Memory for OTA Upgrade Process)

### 3.2.26. FSCI-OTASupportSetMode.Confirm

#### Description

Returns the status of the Set operation

#### Parameters

**Table 30. FSCI-OTASupportSetMode.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x28
Length	1/2	Length in bytes of the following parameters
Status	1	Possible values: 0x00: gSuccess (Success) 0x05: gOtaInvalidParam_c (Invalid Parameter)

### 3.2.27. FSCI-OTASupportStartImage.Request

#### Description

Start downloading image process

#### Parameters

**Table 31. FSCI-OTASupportStartImage.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x29
Length	1/2	Length in bytes of the following parameters
ImageLength	4	Total image size to download

### 3.2.28. FSCI-OTASupportStartImage.Confirm

#### Description

Returns the status of the OTA Start request.

#### Parameters

**Table 32. FSCI-OTASupportStartImage.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x29
Length	1/2	Length in bytes of the following parameters
Status	1	Status Possible values: 0x00: gSuccess_c (Success)
Version	1	Protocol version Possible values: 0x00: Old Protocol Version - Codebase 3.10, 3.11 0x01: New Protocol Version
ExternalMemorySupported	1	External memory Supported Possible values: 0x00: External Memory Not Supported 0x01: External Memory Supported

### 3.2.29. FSCI-Ping.Request

#### Description

FSCI-Ping.Request simply echoes back the payload.

#### Parameters

**Table 33. FSCI-Ping.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x38
Length	1/2	Length in bytes of the following parameters
EchoLength	1	The number of payload echo bytes.
EchoData	EchoLength	The echo payload (raw bytes)

### 3.2.30. FSCI-Ping.Confirm

#### Description

FSCI-Ping.Confirm.

#### Parameters

**Table 34. FSCI-Ping.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x38
Length	1/2	Length in bytes of the following parameters
EchoLength	1	The number of payload echo bytes.
EchoData	EchoLength	The echo payload (raw bytes)

### 3.2.31. FSCI-ReadExtAddr.Request

#### Description

Read the 802.15.4 Extended Address.

#### Parameters

**Table 35. FSCI-ReadExtAddr.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0xD2
Length	1/2	0x00 - This message does not have any parameters

### 3.2.32. FSCI-ReadExtAddr.Confirm

#### Description

Extended address read.

#### Parameters

**Table 36. FSCI-ReadExtAddr.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0xD2
Length	1/2	Length in bytes of the following parameters
Status	1	Status Value
DeviceAddr	8	Devices extended address

### 3.2.33. FSCI-ReadMemoryBlock.Request

#### Description

Reads NumberOfBytes bytes from memory, starting from StartAddress.

#### Parameters

**Table 37. FSCI-ReadMemoryBlock.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x31
Length	1/2	Length in bytes of the following parameters
StartAddress	2	Start address for read operation
NumberOfBytes	1	Number of bytes to be read from memory

### 3.2.34. FSCI-ReadMemoryBlock.Confirm

#### Description

Returns the requested block of memory.

#### Parameters

**Table 38. FSCI-ReadMemoryBlock.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x31
Length	1/2	Length in bytes of the following parameters
ReadData	Length	Data read memory

### 3.2.35. FSCI-ReadUniqueld.Request

#### Description

Read the MCU Unique Id

#### Parameters

**Table 39. FSCI-ReadUniqueld.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0xB0
Length	1/2	0x00

### 3.2.36. FSCI-ReadUniqueld.Confirm

#### Description

Read the MCU Unique Id

#### Parameters

**Table 40. FSCI-ReadUniqueld.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0xB0
Length	1/2	16
Unique Id	16	

### 3.2.37. FSCI-ReadMCUID.Request

#### Description

Read the MCU Id

#### Parameters

**Table 41. FSCI-ReadMCUID.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x81
Length	1/2	0x00

### 3.2.38. FSCI-ReadMCUIId.Confirm

#### Description

Returns the MCU Id

#### Parameters

**Table 42. FSCI-ReadMCUIId.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x81
Length	1/2	4
Unique Id	4	

### 3.2.39. FSCI-ReadSoftwareVersionsId.Request

#### Description

Read the versions of the SW modules

#### Parameters

**Table 43. FSCI- ReadSoftwareVersionsId.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x82
Length	1/2	0x00

### 3.2.40. FSCI-ReadSoftwareVersionsId.Confirm

#### Description

Returns a list of null terminated strings, containing the SW versions.

#### Parameters

**Table 44. FSCI-ReadSoftwareVersionsId.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x82
Length	1/2	Depends on the number of registered modules
Versions string	variable	



### 3.2.41. FSCI-WriteExtAddr.Request

#### Description

Write the 802.15.4 Extended Address.

#### Parameters

**Table 45. FSCI-WriteExtAddr.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0xDB
Length	1/2	Length in bytes of the following parameters
Address	8	The address to write

### 3.2.42. FSCI-WriteExtAddr.Confirm

#### Description

Extended address written.

#### Parameters

**Table 46. FSCI-WriteExtAddr.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0xDB
Length	1/2	Length in bytes of the following parameters
Status	1	Result of write command

### 3.2.43. FSCI-WriteMemoryBlock.Request

#### Description

Writes NumberOfBytes bytes from Data array, starting from StartAddress. In case Start Address is in flash, 0 bytes are written.

#### Parameters

**Table 47. FSCI-WriteMemoryBlock.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA3
OpCode	1	0x30
Length	1/2	Length in bytes of the following parameters
StartAddress	4	Start address for write operation.
NumberOfBytes	1	Number of bytes to be written in memory
Data	NumberOfBytes	Data to be written in memory

### 3.2.44. FSCI-WriteMemoryBlock.Confirm

#### Description

Returns the status of the FSCI-WriteMemorBlock.Request

#### Parameters

**Table 48. FSCI-WriteMemoryBlock.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x30
Length	1/2	Length in bytes of the following parameters
NumberOfBytesWritten	1	In case the Start Address is in flash, 0 bytes are written. In case the Start Address is in RAM, NumberOfBytes are written.

### 3.2.45. FSCI-WakeUp.Indication

#### Description

Indication message sent by the device after waking up from deep sleep.

#### Parameters

**Table 49. FSCI-WakeUp.Indication parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0xA4
OpCode	1	0x41
Length	1/2	Length in bytes of the following parameters

### 3.3. MAC 2006 messages

#### 3.3.1. MacAssociate.Request

##### Description

Request to associate a device to an 802.15.4 PAN.

##### Parameters

**Table 50. MacAssociate.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x00
Length	1/2	Length in bytes of the following parameters
CoordAddress	8	The coordinator address
CoordPANId	2	The PAN coordinator ID
		The coordinator addressing mode
		Possible values:
CoordAddrMode	1	0x02: Value16bitAddr (16 bit short addresses are used) 0x03: Value64bitAddr (64 bit extended addresses are used)
LogicalChannel	1	The current logical channel occupied by the network
securityLevel	1	The security level for the packet
keyIdMode	1	The key identifier mode for the packet
keySource	8	The key source for the packet
keyIndex	1	The key index for the packet
CapabilityInformation	1	The operational capabilities of the associating device

#### 3.3.2. MacAssociate.Confirm

##### Description

Return the status of the Association request.

##### Parameters

**Table 51. MacAssociate.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x01
Length	1/2	Length in bytes of the following parameters
AssocShortAddress	2	The short device address allocated by the coordinator

Table 51. MacAssociate.Confirm parameters

Parameter	Size (bytes)	Comments
Status	1	The status of the association attempt Possible values: 0x00: gSuccess_c (Association successful) 0x01: gPanAtCapacity_c (PAN at capacity) 0x02: gPanAccessDenied_c (PAN access denied) 0xDF: gUnsupportedSecurity_c (Unsupported Security) 0xE1: gChannelAccessFailure_c (Transmission failed due to activity on the channel) 0xE4: gFailedSecurityCheck_c (The received frame failed security check) 0xE8: gInvalidParameter_c (A parameter in the primitive is out of the valid range) 0xE9: gNoAck_c (No acknowledgement was received) 0xEB: gNoData_c (No response data was available following a request) 0xF3: gUnavailableKey_c (The appropriate key is not available in the PIB security tables)
securityLevel	1	The security level for the packet
keyIdMode	1	The key identifier mode for the packet
keySource	8	The key source for the packet
keyIndex	1	The key index for the packet

### 3.3.3. MacAssociate.Indication

#### Description

This message appears on a Coordinator to inform the upper layer that a device has requested to associate to the PAN.

#### Parameters

Table 52. MacAssociate.Indication parameters

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x00
Length	1/2	Length in bytes of the following parameters
DeviceAddress	8	The address of the device requesting association
securityLevel	1	The security level used
keyIdMode	1	The mode used to identify the key used
keySource	8	The originator of the key used
keyIndex	1	The index of the key used
CapabilityInformation	1	The operational capabilities of the device requesting association

### 3.3.4. MacAssociate.Response

#### Description

This message is sent by the Coordinator, in response to a MacAssociate.Request

#### Parameters

**Table 53. MacAssociate.Response parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x01
Length	1/2	Length in bytes of the following parameters
DeviceAddress	8	The address of the device requesting association
AssocShortAddress	2	The short device address allocated by the coordinator
securityLevel	1	The security level for the packet
keyIdMode	1	The key identifier mode for the packet
keySource	8	The key source for the packet
keyIndex	1	The key index for the packet
Status	1	The status of the association attempt Possible values: 0x00: gSuccess_c (Association successful) 0x01: gPanAtCapacity_c (PAN at capacity) 0x02: gPanAccessDenied_c (PAN access denied)

### 3.3.5. MacBeaconNotify.Indication

#### Description

Notify the upper layer that a Beacon frame has been received.

#### Parameters

**Table 54. MacBeaconNotify.Indication parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x04
Length	1/2	Length in bytes of the following parameters
BSN	1	The beacon sequence number
PendAddrSpec	1	The beacon pending address specification
sduLength	1	The beacon frame payload length (in bytes)
AddrList16bit	2 x (PendAddrSpec & (bitwise AND) 0x07)	The list of 16-bit short addresses of the devices for which the beacon source has data
AddrList64bit	8 x ((PendAddrSpec & (bitwise AND) 0x70) >> 4)	The list of 16-bit short addresses of the devices for which the beacon source has data
PanDescriptor	Variable	The PAN descriptor for the received beacon frame Structure type parameter. See detailed table below for parameter structure.
Sdu	sduLength	The beacon frame payload (raw bytes)
EBSN <sup>1</sup>	1	Beacon sequence number used for enhanced beacon frames
beaconType <sup>1</sup>	1	Indicates a beacon (0x00) or enhanced beacon (0x01) was received

1. These fields are present only in the commands from the MAC2006\_g.XML file.

**Table 55. PanDescriptor parameter structure**

Structure Parameter	Size (bytes)	Comments
CoordAddress	8	The coordinator address
CoordPANId	2	The PAN coordinator ID
CoordAddrMode	1	The coordinator addressing mode Possible values: 0x02: Value16bitAddr (16 bit short addresses are used) 0x03: Value64bitAddr (64 bit extended addresses are used)
LogicalChannel	1	The current logical channel occupied by the network
SecurityFailure	1	The parameter is set to TRUE if there was an error in the security processing of the frame
SuperFrameSpec	2	The superframe specification
GTSPermit	1	TRUE if the beacon is from a PAN coordinator which is accepting GTS requests
LinkQuality	1	The link quality at which the network beacon was received
TimeStamp	3	The time stamp (in symbols)
securityLevel	1	The security level for the packet
keyIdMode	1	The key identifier mode for the packet
keySource	8	The key source for the packet
keyIndex	1	The key index for the packet

### 3.3.6. MacBeaconStart.Indication

#### Description

Legacy

#### Parameters

**Table 56. MacBeaconStart.Indication parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x12
Length	1/2	Length in bytes of the following parameters
source	1	

### 3.3.7. MacCommStatus.Indication

#### Description

Informs the upper layer about a communication status

#### Parameters

**Table 57. MacCommStatus.Indication parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x0C
Length	1/2	Length in bytes of the following parameters
SrcAddr	8	Source address
PANId	2	The PAN ID
SrcAddrMode	1	The source address mode Possible values: 0x00: None (No source address supplied) 0x01: Reserved (This address mode is reserved) 0x02: Value16bitAddr (16 bit address supplied) 0x03: Value64bitAddr (64 bit address supplied)
DestAddr	8	Destination address
DestAddrMode	1	The destination address mode Possible values: 0x00: None (No destination address supplied) 0x01: Reserved (This address mode is reserved) 0x02: Value16bitAddr (16 bit address supplied) 0x03: Value64bitAddr (64 bit address supplied)
Status	1	Possible values: 0x00: gSuccess_c (COMM status OK) 0xE1: gChannelAccessFailure_c (Transmission failed due to activity on the channel) 0xE4: gFailedSecurityCheck_c (The received frame failed security check) 0xE5: gFrameTooLong_c (The frame was too long to send after security processing) 0xE8: gInvalidParameter_c (A parameter in the primitive is out of the valid range) 0xE9: gNoAck_c (No acknowledgement was received) 0xF0: gTransactionExpired_c (The transaction has expired and its information discarded) 0xF1: gTransactionOverflow_c (There is no capacity to store the transaction)

**Table 57. MacCommStatus.Indication parameters**

Parameter	Size (bytes)	Comments
		0xF3: gUnavailableKey_c (The appropriate key is not available in the ACL)
securityLevel	1	The security level for the packet
keyIdMode	1	The key identifier mode for the packet
keySource	8	The key source for the packet
keyIndex	1	The key index for the packet

### 3.3.8. MacData.Request

#### Description

Request to send a data packet over the air

#### Parameters

**Table 58. MacData.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x87
OpCode	1	0x00
Length	1/2	Length in bytes of the following parameters
DestAddr	8	The destination address
DestPanId	2	The destination PAN identifier
		The destination address mode to use Possible values: 0x00: None (No source address supplied) 0x01: Reserved (This address mode is reserved) 0x02: Value16bitAddr (16 bit address supplied) 0x03: Value64bitAddr (64 bit address supplied)
DestAddrMode	1	
SrcAddr	8	The source address
SrcPanId	2	The source PAN identifier
		The source address mode to use Possible values: 0x00: None (No source address supplied) 0x01: Reserved (This address mode is reserved) 0x02: Value16bitAddr (16 bit address supplied) 0x03: Value64bitAddr (64 bit address supplied)
SrcAddrMode	1	
msduLength	1	The number of payload bytes
msduHandle	1	The msdu handle
TxOptions	1	Bitmask of valid tx options
securityLevel	1	The security level for the packet
keyIdMode	1	The key identifier mode for the packet
keySource	8	The key source for the packet
keyIndex	1	The key index for the packet
msdu	msduLength	The msdu payload



### 3.3.9. MacData.Confirm

#### Description

Returns the status of the data request.

#### Parameters

**Table 59. MacData.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x86
OpCode	1	0x00
Length	1/2	Length in bytes of the following parameters
msduHandle	1	Handle of packet to be confirmed
Status	1	<p>The status of the data request</p> <p>Possible values:</p> <p>0x00: gSuccess_c</p> <p>0xE1: gChannelAccessFailure_c (Transmission failed due to activity on the channel)</p> <p>0xE4: gFailedSecurityCheck_c (The received frame failed security check)</p> <p>0xE5: gFrameTooLong_c (The frame was too long to send after security processing)</p> <p>0xE6: gInvalidGts_c (The requested GTS is invalid)</p> <p>0xE8: gInvalidParameter_c (A parameter in the primitive is out of the valid range)</p> <p>0xE9: gNoAck_c (No acknowledgement was received)</p> <p>0xF0: gTransactionExpired_c (The transaction has expired and its information discarded)</p> <p>0xF1: gTransactionOverflow_c (There is no capacity to store the transaction)</p> <p>0xF3: gUnavailableKey_c (The appropriate key is not available in the ACL)</p>

### 3.3.10. MacData.Indication

#### Description

A data packet was receiver over the air.

#### Parameters

**Table 60. MacData.Indication parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x86
OpCode	1	0x01
Length	1/2	Length in bytes of the following parameters
DestAddr	8	The destination address
DestPanId	2	The destination PAN identifier
DestAddrMode	1	<p>The destination address mode to use</p> <p>Possible values:</p> <p>0x00: None (No source address supplied)</p> <p>0x01: Reserved (This address mode is reserved)</p> <p>0x02: Value16bitAddr (16 bit address supplied)</p> <p>0x03: Value64bitAddr (64 bit address supplied)</p>

Table 60. MacData.Indication parameters

Parameter	Size (bytes)	Comments
SrcAddr	8	The source address
SrcPanId	2	The source PAN identifier
SrcAddrMode	1	The source address mode to use Possible values: 0x00: None (No source address supplied) 0x01: Reserved (This address mode is reserved) 0x02: Value16bitAddr (16 bit address supplied) 0x03: Value64bitAddr (64 bit address supplied)
msduLength	1	The number of payload bytes
mpduLinkQuality	1	The link quality measured
dsn	1	The dsn for the received data frame
timeStamp	4	The timestamp for the received data frame
securityLevel	1	The security level for the packet
keyIdMode	1	The key identifier mode for the packet
keySource	8	The key source for the packet
keyIndex	1	The key index for the packet
msdu	msduLength	The msdu payload

### 3.3.11. MacDisassociate.Request

#### Description

Request to disassociate from a PAN

#### Parameters

Table 61. MacDisassociate.Request parameters

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x02
Length	1/2	Length in bytes of the following parameters
DeviceAddress	8	The address of the device to disassociate [from]
DevicePanId	2	The PAN Id of the device to disassociate [from]
DeviceAddrMode	1	The device addressing mode Possible values: 0x02: Value16bitAddr (16 bit short addresses are used) 0x03: Value64bitAddr (64 bit extended addresses are used)
DisassociateReason	1	The reason for the disassociation Possible values: 0x01: gCoordLeave_c (The coordinator wishes the device to leave the PAN) 0x02: gDeviceLeave_c (The device wishes to leave the PAN)
txIndirect	1	Indirect or direct transmission of the Disassociate Request command
securityLevel	1	The security level for the packet
keyIdMode	1	The key identifier mode for the packet
keySource	8	The key source for the packet
keyIndex	1	The key index for the packet

### 3.3.12. MacDisassociate.Confirm

#### Description

Returns the status of the disassociation request

#### Parameters

**Table 62. MacDisassociate.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x03
Length	1/2	Length in bytes of the following parameters
DeviceAddress	8	The address of the device that has either requested disassociation or been instructed to disassociate by its coordinator.
DevicePanID	2	The PanID of the device that has either requested disassociation or been instructed to disassociate by its coordinator.
DeviceAddrMode	1	The addressing mode of the device that has either requested disassociation or been instructed to disassociate by its coordinator.
Status	1	The status of the disassociation attempt Possible values: 0x00: gSuccess_c (Disassociation successful) 0xE1: gChannelAccessFailure_c (Transmission failed due to activity on the channel) 0xE4: gFailedSecurityCheck_c (The received frame failed security check) 0xE8: gInvalidParameter_c (A parameter in the primitive is out of the valid range) 0xE9: gNoAck_c (No acknowledgement was received) 0xF0: gTransactionExpired_c (The transaction has expired and its information discarded) 0xF1: gTransactionOverflow_c (There is no capacity to store the transaction) 0xF3: gUnavailableKey_c (The appropriate key is not available in the ACL)

### 3.3.13. MacDisassociate.Indication

#### Description

Informs the upper layer that a disassociation request have been received

#### Parameters

**Table 63. MacDisassociate.Indication parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x02
Length	1/2	Length in bytes of the following parameters
DeviceAddress	8	The address of the device
DisassociateReason	1	The reason for the disassociation Possible values: 0x01: gCoordLeave_c (The coordinator wishes the device to leave the PAN) 0x02: gDeviceLeave_c (The device wishes to leave the PAN)
securityLevel	1	The security level for the packet
keyIdMode	1	The key identifier mode for the packet
keySource	8	The key source for the packet
keyIndex	1	The key index for the packet

### 3.3.14. MacGetPIBAttribute.Request

#### Description

Get the value of a MAC PIB

#### Parameters

**Table 64. MacGetPIBAttribute.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x03
Length	1/2	Length in bytes of the following parameters
PIBAttribute	1	<p>The MAC PIB attribute identifier</p> <p>Possible values:</p> <p>0x40: macAckWaitDuration (The maximum number of symbols to wait for an acknowledgement)</p> <p>0x41: macAssociationPermit (The coordinator is allowing association or not)</p> <p>0x42: macAutoRequest (The device automatically sends data requests (if listed) or not)</p> <p>0x43: macBattLifeExt (Extended battery life enabled or not)</p> <p>0x44: macBattLifeExtPeriods (The number of backoff periods where the receiver is enabled)</p> <p>0x45: macBeaconPayload (The contents of the beacon payload)</p> <p>0x46: macBeaconPayloadLength (The length of the beacon payload in bytes)</p> <p>0x47: macBeaconOrder (Specifies the beacon interval)</p> <p>0x48: macBeaconTxTime (Time (in symbols) since last transmitted beacon frame)</p> <p>0x49: macBSN (The beacon sequence number)</p> <p>0x4A: macCoordExtendedAddress (The 64-bit address of the coordinator)</p> <p>0x4B: macCoordShortAddress (The 16-bit address of the coordinator)</p> <p>0x4C: macDSN (The data sequence number)</p> <p>0x4D: macGTSPermit (The coordinator is allowing GTS requests or not)</p> <p>0x21: macLogicalChannel (The channel used)</p> <p>0x4E: macMaxCSMABackoffs (The maximum number of backoff slots used)</p> <p>0x4F: macMinBE (The minimum value of the backoff slot exponent)</p> <p>0x50: macPANId (The id of the PAN on which the device is operating)</p> <p>0x51: macPromiscuousMode (Promiscuous (receive all) mode enabled or not)</p> <p>0x20: macRole (The current role of the device)</p> <p>0x24: macBeaconResponseDenied (Beacon response denied - if set to true, no beacon will be issued for an active scan request)</p> <p>0x52: macRxOnWhenIdle (The receiver is enabled during IDLE periods)</p> <p>0x53: macShortAddress (The 16-bit short address of the device)</p> <p>0x54: macSuperframeOrder (The length of the active portion of the superframe)</p> <p>0x55: macTransactionPersistenceTime (The maximum time (in superframe periods) that data is indicated in the beacon)</p> <p>0x56: macAssociatedPANCoord (Indicates if the device is currently associated to the PAN coordinator)</p> <p>0x57: macMaxBe (MAC Maximum Backoff Exponent (BE))</p> <p>0x58: macMaxFrameTotalWaitTime (MAC Frame Total Wait time)</p> <p>0x59: macMaxFrameRetries (MAC Max Frame Retries allowed for a transmission failure)</p> <p>0x5A: macResponseWaitTime (MAC Response Wait time)</p> <p>0x5B: macSyncSymbolOffset (MAC Sync Symbol Offset)</p> <p>0x5C: macTimestampSupported (MAC Timestamp Supported)</p> <p>0x5D: macSecurityEnabled (MAC Security Enabled)</p> <p>0x63: macCslPeriod (MAC Csl Period)</p> <p>0x64: macCslMaxPeriod (MAC Csl Max Period)</p>

**Table 64. MacGetPIBAttribute.Request parameters**

Parameter	Size (bytes)	Comments
		0x65: macCslFramePendingWait (MAC Csl Frame Pending Wait) 0x66: macEnhAckWaitDuration (MAC Enhanced Ack Wait Duration) 0x67: macRitPeriod (MAC Rit Period) 0x68: macRitDataWaitDuration (MAC Rit Data Wait Duration) 0x69: macRitTxWaitDuration (MAC Rit Tx Wait Duration) 0x6A: macRittle (MAC Rit Information Element) 0x6B: macPhyMode (MAC Phy Mode) 0x6C: macPhyCcaDuration (MAC Phy Cca Duration) 0x6D: macPhyFskScramblePsdu (MAC Phy Fsk Scramble Psdu) 0x6E: macPhyFrequencyBand (MAC Phy Frequency Band) 0x71: macKeyTable (MAC Key Table) 0x72: macKeyTableEntries (MAC Key Table entries) 0x73: macDeviceTable (MAC Device Table) 0x74: macDeviceTableEntries (MAC Device Table entries) 0x75: macSecurityLevelTable (MAC Security Level Table) 0x76: macSecurityLevelTableEntries (MAC Security Level Table entries) 0x77: macFrameCounter (MAC Frame Counter) 0x78: macAutoRequestSecurityLevel (MAC Auto Request Security Level) 0x79: macAutoRequestKeyIdMode (MAC Auto Request Key ID Mode) 0x7A: macAutoRequestKeySource (MAC Auto Request Key Source) 0x7B: macAutoRequestKeyIndex (MAC Auto Request Key Index) 0x7C: macDefaultKeySource (MAC Default Key Source) 0x7D: macPANCoordExtendedAddress (MAC PAN Coordinator Extended Address) 0x7E: macPANCoordShortAddress (MAC PAN Coordinator Short Address) 0x7F: macKeyIdLookupDescriptor (MAC Key Id Lookup Descriptor) 0x80: macKeyIdLookupEntries (MAC Key Id Lookup Entries) 0x81: macKeyDeviceList (MAC Key Device List) 0x82: macKeyDeviceListEntries (MAC Key Device List Entries) 0x83: macKeyUsageList (MAC Key Usage List) 0x84: macKeyUsageListEntries (MAC Key Usage List Entries) 0x85: macKey (MAC Key) 0x86: macKeyUsageFrameType (MAC Key Usage Frame Type) 0x87: macKeyUsageCmdFrameId (MAC Key Usage Cmd Frame Id) 0x88: macKeyDeviceDescriptorHandle (Handle to the DeviceDescriptor corresponding to the device) 0x89: macUniqueDevice (MAC Unique Device) 0x8A: macBlackListed (MAC Blacklisted) 0x8B: macSecLevFrameType (MAC Security Level Frame Type) 0x8C: macSecLevCommandFrameIdentifier (MAC Security Level Command Frame Identifier) 0x8D: macSecLevSecurityMinimum (MAC - Minimum Security Level) 0x8E: macSecLevDeviceOverrideSecurityMinimum (MAC - Minimum Security Level) 0x8F: macDeviceDescriptorPanId (MAC Device Descriptor PAN id) 0x90: macDeviceDescriptorShortAddress (MAC Device Descriptor Short Address) 0x91: macDeviceDescriptorExtAddress (MAC Device Descriptor Extended Address) 0x92: macDeviceDescriptorFrameCounter (MAC Device Descriptor Frame Counter) 0x93: macDeviceDescriptorExempt (MAC Device Descriptor Exempt) 0x94: macKeyIdLookupData (MAC Key Id Lookup Data) 0x95: macKeyIdLookupDataSize (MAC Key Id Lookup Data Size) 0x96: macKeyTableCrtEntry (MAC Key Table - Current Entry) 0x97: macDeviceTableCrtEntry (MAC Device Table - Current Entry) 0x98: macSecurityLevelTableCrtEntry (MAC Security Level Table - Current Entry) 0x99: macKeyIdLookupListCrtEntry (MAC Key Id Lookup List - Current Entry) 0x9A: macKeyUsageListCrtEntry (MAC Key Usage List - Current Entry) 0x9B: macKeyDeviceListCrtEntry (MAC Key Device List - Current Entry)

**Table 64. MacGetPIBAttribute.Request parameters**

Parameter	Size (bytes)	Comments
		0x25: macNBSuperFrameInterval (Non-beacon mode superframe interval) 0x22: macTreemodeStartTime (Treemode Start Time) 0x23: macPanIdConflictDetection (PanId Conflict Detection) 0x27: macBeaconResponseLQIThreshold (Beacon response LQI Threshold) 0x28: BeaconEDCfpStartTrimVaue (Beacon Mode ED CFP Start Trim Value)
Index	1	Index PIB Table entries (not used)

### 3.3.15. MacGetPIBAttribute.Confirm

#### Description

Return the value of the requested MAC PIB, or an error status.

#### Parameters

**Table 65. MacGetPIBAttribute.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x05
Length	1/2	Length in bytes of the following parameters
Status	1	The result of the PhyGetPIBAttribute.Request Possible values: 0x00: gSuccess_c (The MacGetPIBAttribute.Request operation was successful) 0xF4: UNSUPPORTED_ATTRIBUTE (Unknown PIB attribute)
PIBAttribute	1	The MAC PIB attribute identifier (same as the request)
Index	1	
DataLength	2	Length of the attribute data
PIBAttributeValue	DataLength	The MAC PIB attribute value

### 3.3.16. MacGTS.Request

#### Description

Request a GTS in the superframe

#### Parameters

**Table 66. MacGTS.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x04
Length	1/2	Length in bytes of the following parameters
GTSCharacteristics	1	The characteristics of the GTS request
securityLevel	1	The security level to be used
keyIdMode	1	The mode used to identify the key to be used
keySource	8	The originator of the key to be used
keyIndex	1	The index of the key to be used

### 3.3.17. MacGTS.Confirm

#### Description

Returns the status of the GTS Request.

#### Parameters

**Table 67. MacGTS.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x07
Length	1/2	Length in bytes of the following parameters
Status	1	<p>The status of the GTS request</p> <p>Possible values:</p> <p>0x00: gSuccess_c (Disassociation successful)</p> <p>0xE1: gChannelAccessFailure_c (Transmission failed due to activity on the channel)</p> <p>0xE2: gDenied_c (The GTS request has been denied by the PAN coordinator)</p> <p>0xE4: gFailedSecurityCheck_c (The received frame failed security check)</p> <p>0xE8: gInvalidParameter_c (A parameter in the primitive is out of the valid range)</p> <p>0xE9: gNoAck_c (No acknowledgement was received)</p> <p>0xEB: gNoData_c (No response data was available following a request)</p> <p>0xEC: gNoShortAddress_c (The operation failed due to a short address not being allocated)</p> <p>0xF3: gUnavailableKey_c (The appropriate key is not available in the PIB security tables)</p>
GTSCharacteristics	1	The characteristics of the GTS request

### 3.3.18. MacGTS.Indication

#### Description

Informs the Coordinator that a GTS was requested.

#### Parameters

**Table 68. MacGTS.Indication parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x06
Length	1/2	Length in bytes of the following parameters
DeviceAddress	2	The short address of the device that has been allocated or deallocated a GTS
GTSCharacteristics	1	The characteristics of the GTS request
securityLevel	1	The security level for the packet
keyIdMode	1	The key identifier mode for the packet
keySource	8	The key source for the packet
keyIndex	1	The key index for the packet

### 3.3.19. MacKeepAlive.Request

#### Description

The MLME-KEEP-ALIVE.request primitive is generated by the next higher layer, and issued to the MLME to request monitor of frames exchanged with a TSCH neighbor used as clock source, and configure the keep-alive period in timeslots for that clock source.

#### Parameters

**Table 69. MacKeepAlive.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x10
Length	1/2	Length in bytes of the following parameters
DstAddr	2	Address of the neighbor device with which to maintain timing. Keep-alives with dstAddr of 0xffff do not expect to be acknowledged and cannot be used for timekeeping
KeepAlivePeriod	2	Period in timeslots after which a frame is sent if no frames have been sent to dstAddr

### 3.3.20. MacKeepAlive.Confirm

#### Description

The MLME-KEEP-ALIVE.confirm primitive is generated by the MAC layer, and issued to the next higher layer to confirm the request to configure the keep-alive frames for a TSCH clock source.

#### Parameters

**Table 70. MacKeepAlive.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x18
Length	1/2	Length in bytes of the following parameters
Status	1	Indicates results of the MLME-KEEP-ALIVE.request



### 3.3.21. MacOrphan.Indication

#### Description

Informs the upper layer that an “orphan” device is scanning for its parent.

#### Parameters

**Table 71. MacOrphan.Indication parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x08
Length	1/2	Length in bytes of the following parameters
OrphanAddress	8	Extended address of orphaned device
securityLevel	1	The security level for the packet
keyIdMode	1	The key identifier mode for the packet
keySource	8	The key source for the packet
keyIndex	1	The key index for the packet

### 3.3.22. MacOrphan.Response

#### Description

Respond to the orphan device.

#### Parameters

**Table 72. MacOrphan.Response parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x05
Length	1/2	Length in bytes of the following parameters
OrphanAddress	8	Address of orphaned device
ShortAddress	2	Orphans association address, if applicable
AssociatedMember	1	Indicates if orphaned device was associated to this coordinator
securityLevel	1	The security level to be used
keyIdMode	1	The mode used to identify the key to be used
keySource	8	The key source for the packet
keyIndex	1	The key index for the packet

### 3.3.23. MacPoll.Request

#### Description

Check if the coordinator has data pending for this device.

#### Parameters

**Table 73. MacPoll.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x0C
Length	1/2	Length in bytes of the following parameters
CoordAddress	8	The coordinator address
CoordPANId	2	The PAN coordinator ID
CoordAddrMode	1	The coordinator addressing mode Possible values: 0x02: Value16bitAddr (16 bit short addresses are used) 0x03: Value64bitAddr (64 bit extended addresses are used)
securityLevel	1	The security level to be used
keyIdMode	1	The mode used to identify the key to be used
keySource	8	The originator of the key to be used
keyIndex	1	The index of the key to be used

### 3.3.24. MacPoll.Confirm

#### Description

Returns the status of the poll request.

#### Parameters

**Table 74. MacPoll.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x10
Length	1/2	Length in bytes of the following parameters
Status	1	The status of the poll request Possible values: 0x00: gSuccess_c (Association successful) 0xE1: gChannelAccessFailure_c (Transmission failed due to activity on the channel) 0xE4: gFailedSecurityCheck_c (The received frame failed security check) 0xE8: gInvalidParameter_c (A parameter in the primitive is out of the valid range) 0xE9: gNoAck_c (No acknowledgement was received) 0xEB: gNoData_c (No response data was available following a request) 0xF3: gUnavailableKey_c (The appropriate key is not available in the ACL)

### 3.3.25. MacPollNotify.Indication

#### Description

Notify the Coordinator's upper layer that a device has issued a MLME-POLL.Request

#### Parameters

**Table 75. MacPollNotify.Indication parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x14
Length	1/2	Length in bytes of the following parameters
SrcAddrMode	1	The source address mode to use Possible values: 0x00: None (No source address supplied) 0x01: Reserved (This address mode is reserved) 0x02: Value16bitAddr (16 bit address supplied) 0x03: Value64bitAddr (64 bit address supplied)
SrcAddr	8	The source address
SrcPanId	2	The source PAN identifier

### 3.3.26. MacPurge.Request

#### Description

Purge an indirect data transmission

#### Parameters

**Table 76. MacPurge.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x87
OpCode	1	0x01
Length	1/2	Length in bytes of the following parameters
msduHandle	1	Handle of packet to be purged

### 3.3.27. MacPurge.Confirm

#### Description

Returns the status of the purge request.

#### Parameters

**Table 77. MacPurge.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x86
OpCode	1	0x02
Length	1/2	Length in bytes of the following parameters
msduHandle	1	Handle of packet to be purged
status	1	The status of the purge request Possible values: 0x00: gSuccess_c (Purge successful) 0xE7: gInvalidHandle_c (Invalid handle)

### 3.3.28. MacReset.Request

#### Description

Reset the 802.15.4 MAC to the initial state. The MAC PIBs can be preserved or not, depending on the “SetDefaultPib” field of the request

#### Parameters

**Table 78. MacReset.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x06
Length	1/2	Length in bytes of the following parameters
SetDefaultPib	1	Reset MAC PIB attributes to default values

### 3.3.29. MacReset.Confirm

#### Description

Return the status of the MLME-RESET.Request

#### Parameters

**Table 79. MacReset.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x09
Length	1/2	Length in bytes of the following parameters
Status	1	The status of the reset attempt Possible values: 0x00: gSuccess_c (Reset successful) 0xE3: gDisableTrxFailure_c (Failed to disable the transceiver)

### 3.3.30. MacRxEnable.Request

#### Description

Request to start an Rx sequence for the specified duration.

If the “RxOnWhenIdle” MAC PIB is set to TRUE, this request does not do anything.

#### Parameters

**Table 80. MacRxEnable.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x07
Length	1/2	Length in bytes of the following parameters
DeferPermit	1	Defer receiver enable
RxOnTime	3	Number of symbols from the start of the superframe before receiver is to be enabled
RxOnDuration	3	The number of symbols the receiver is turned on

### 3.3.31. MacRxEnable.Confirm

#### Description

Return the status of the MLME-RX-ENABLE.Request

#### Parameters

**Table 81. MacRxEnable.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x0A
Length	1/2	Length in bytes of the following parameters
Status	1	Status field Possible values: 0x00: SUCCESS (Scan completed with requested info found) 0xD0: gRxEnableDone_c (Rx Enable Done) 0xE8: gInvalidParameter_c (A parameter in the primitive is out of the valid range) 0xF2: gTxActive_c (The transceiver was in the transmitter enabled state when the receiver was requested to be enabled.) 0xED: gOutOfCap_c (A receiver enable request was unsuccessful because it could not be completed within the CAP.)

### 3.3.32. MacScan.Request

#### Description

Start a scan operation: energy detect, active/passive scan, orphan scan.

#### Parameters

**Table 82. MacScan.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x08
Length	1/2	Length in bytes of the following parameters
ScanType	1	Type of scan to perform Possible values: 0x00: EnergyDetection (Request energy detection (ED) scan) 0x01: ActiveScan (Request active scan, transmitting Beacons (FFD only)) 0x02: PassiveScan (Request passive scan, not transmitting Beacons) 0x03: OrphanScan (Request scanning for orphans)
ScanChannels	4	List of bits, indicating (=1) channels to scan
ScanDuration	1	Duration of scan (order of 2), for each channel
securityLevel	1	The security level to be used
keyIdMode	1	The mode used to identify the key to be used
keySource	8	The originator of the key to be used
keyIndex	1	The index of the key to be used

### 3.3.33. MacScan.Confirm

#### Description

Return the status of the MLME-SCAN.Request

#### Parameters

**Table 83. MacScan.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x0B
Length	1/2	Length in bytes of the following parameters
Status	1	Status field Possible values: 0x00: SUCCESS (Scan completed with requested info found) 0xEA: NO_BEACON (No coordinator responded with requested info) 0xE8: gInvalidParameter_c (A parameter in the primitive is out of the valid range)
ScanType	1	Type of scan to perform Possible values: 0x00: EnergyDetection (Request energy detection (ED) scan) 0x01: ActiveScan (Request active scan, transmitting Beacons (FFD only)) 0x02: PassiveScan (Request passive scan, not transmitting Beacons) 0x03: OrphanScan (Request scanning for orphans)
ResultListSize	1	Number of elements in result list
UnscannedChannels (only PS/AS)	4	Bit mask indicating unscanned channels
ResultList	Variable	List of results "Union" type parameter. Its structure is based on the value of parameter ScanType. See detailed table below for parameter structure.

**Table 84. ResultList parameter structure**

ScanType	Structure parameter	Size (bytes)	Comments
0x00	EnergyDetection	ResultListSize	List of energy detect results
0x01	Active Scan	31 x ResultListSize	List of PAN Descriptors
0x02	Passive Scan	31 x ResultListSize	List of PAN Descriptors
0x03	Orphan Scan	1	No result list for orphan scan

### 3.3.34. MacSetLink.Request

#### Description

The MLME-SET-LINK.request primitive is generated by the next higher layer, and issued to the MLME to request a set of a TSCH link to the MAC sublayer.

#### Parameters

**Table 85. MacSetLink.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x0E
Length	1/2	Length in bytes of the following parameters
Operation	1	Type of link management operation to be performed. ADD_LINK = 0, DELETE_LINK = 1, MODIFY_LINK = 2
LinkHandle	2	Unique identifier (local to specified slotframe) for the link
SlotframeHandle	1	The slotframeHandle of the slotframe to which the link is associated
Timeslot	2	Timeslot of the link to be added
ChannelOffset	2	The Channel offset of the link
LinkOptions	1	b0 = Transmit, b1 = Receive, b2 = Shared, b3= Timekeeping, b4–b7 reserved
LinkType	1	Type of link. NORMAL = 0. ADVERTISING = 1, and indicates the link may be used to send an Enhanced beacon
NodeAddr	2	Address of neighbor device connected by the link. 0xffff indicates the link may be used for frames destined for the broadcast address

### 3.3.35. MacSetLink.Confirm

#### Description

The MLME-SET-LINK.confirm primitive is generated by the MAC layer, and issued to the next higher layer to confirm the set of a TSCH link to the MAC sublayer.

#### Parameters

**Table 86. MacSetLink.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x16
Length	1/2	Length in bytes of the following parameters
Status	1	Result of the request operation
LinkHandle	2	Unique (local to specified slotframe) identifier for the link
SlotframeHandle	1	The slotframeHandle of the slotframe to which the link is associated



### 3.3.36. MacSetPIBAtribute.Request

#### Description

Change the value of a MAC PIB

#### Parameters

**Table 87. MacSetPIBAtribute.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x09
Length	1/2	Length in bytes of the following parameters
PIBAtribute	1	<p>The MAC PIB attribute identifier</p> <p>Possible values:</p> <p>0x40: macAckWaitDuration (The maximum number of symbols to wait for an acknowledgement)</p> <p>0x41: macAssociationPermit (The coordinator is allowing association or not)</p> <p>0x42: macAutoRequest (The device automatically sends data requests (if listed) or not)</p> <p>0x43: macBattLifeExt (Extended battery life enabled or not)</p> <p>0x44: macBattLifeExtPeriods (The number of backoff periods where the receiver is enabled)</p> <p>0x45: macBeaconPayload (The contents of the beacon payload)</p> <p>0x46: macBeaconPayloadLength (The length of the beacon payload in bytes)</p> <p>0x47: macBeaconOrder (Specifies the beacon interval)</p> <p>0x48: macBeaconTxTime (Time (in symbols) since last transmitted beacon frame)</p> <p>0x49: macBSN (The beacon sequence number)</p> <p>0x4A: macCoordExtendedAddress (The 64-bit address of the coordinator)</p> <p>0x4B: macCoordShortAddress (The 16-bit address of the coordinator)</p> <p>0x4C: macDSN (The data sequence number)</p> <p>0x4D: macGTSPermit (The coordinator is allowing GTS requests or not)</p> <p>0x21: macLogicalChannel (The channel to use)</p> <p>0x4E: macMaxCSMABackoffs (The maximum number of backoff slots used)</p> <p>0x4F: macMinBE (The minimum value of the backoff slot exponent)</p> <p>0x50: macPANId (The id of the PAN on which the device is operating)</p> <p>0x51: macPromiscuousMode (Promiscuous (receive all) mode enabled or not)</p> <p>0x20: macRole (The current role of the device)</p> <p>0x24: macBeaconResponseDenied (Beacon response denied - if set to true, no beacon will be issued for an active scan request)</p> <p>0x52: macRxOnWhenIdle (The receiver is enabled during IDLE periods)</p> <p>0x53: macShortAddress (The 16-bit short address of the device)</p> <p>0x54: macSuperframeOrder (The length of the active portion of the superframe)</p> <p>0x55: macTransactionPersistenceTime (The maximum time (in superframe periods) that data is indicated in the beacon)</p> <p>0x56: macAssociatedPANCoord (Indicates if the device is currently associated to the PAN coordinator)</p> <p>0x57: macMaxBe (MAC Maximum Backoff Exponent (BE))</p> <p>0x58: macMaxFrameTotalWaitTime (MAC Frame Total Wait time)</p> <p>0x59: macMaxFrameRetries (MAC Max Frame Retries allowed for a transmission failure)</p> <p>0x5A: macResponseWaitTime (MAC Response Wait time)</p> <p>0x5B: macSyncSymbolOffset (MAC Sync Symbol Offset)</p> <p>0x5C: macTimestampSupported (MAC Timestamp Supported)</p> <p>0x5D: macSecurityEnabled (MAC Security Enabled)</p> <p>0x63: macCslPeriod (MAC Csl Period)</p> <p>0x64: macCslMaxPeriod (MAC Csl Max Period)</p> <p>0x65: macCslFramePendingWait (MAC Csl Frame Pending Wait)</p> <p>0x66: macEnhAckWaitDuration (MAC Enhanced Ack Wait Duration)</p> <p>0x67: macRitPeriod (MAC Rit Period)</p> <p>0x68: macRitDataWaitDuration (MAC Rit Data Wait Duration)</p>

**Table 87. MacSetPIBAttribute.Request parameters**

Parameter	Size (bytes)	Comments
		0x69: macRitTxWaitDuration (MAC Rit Tx Wait Duration) 0x6A: macRitle (MAC Rit Information Element) 0x6B: macPhyMode (MAC Phy Mode) 0x6C: macPhyCcaDuration (MAC Phy Cca Duration) 0x6D: macPhyFskScramblePsdu (MAC Phy Fsk Scramble Psdu) 0x6E: macPhyFrequencyBand (MAC Phy Frequency Band) 0x71: macKeyTable (MAC Key Table) 0x72: macKeyTableEntries (MAC Key Table entries) 0x73: macDeviceTable (MAC Device Table) 0x74: macDeviceTableEntries (MAC Device Table entries) 0x75: macSecurityLevelTable (MAC Security Level Table) 0x76: macSecurityLevelTableEntries (MAC Security Level Table entries) 0x77: macFrameCounter (MAC Frame Counter) 0x78: macAutoRequestSecurityLevel (MAC Auto Request Security Level) 0x79: macAutoRequestKeyIdMode (MAC Auto Request Key ID Mode) 0x7A: macAutoRequestKeySource (MAC Auto Request Key Source) 0x7B: macAutoRequestKeyIndex (MAC Auto Request Key Index) 0x7C: macDefaultKeySource (MAC Default Key Source) 0x7D: macPANCoordExtendedAddress (MAC PAN Coordinator Extended Address) 0x7E: macPANCoordShortAddress (MAC PAN Coordinator Short Address) 0x7F: macKeyIdLookupDescriptor (MAC Key Id Lookup Descriptor) 0x80: macKeyIdLookupEntries (MAC Key Id Lookup Entries) 0x81: macKeyDeviceList (MAC Key Device List) 0x82: macKeyDeviceListEntries (MAC Key Device List Entries) 0x83: macKeyUsageList (MAC Key Usage List) 0x84: macKeyUsageListEntries (MAC Key Usage List Entries) 0x85: macKey (MAC Key) 0x86: macKeyUsageFrameType (MAC Key Usage Frame Type) 0x87: macKeyUsageCmdFrameId (MAC Key Usage Cmd Frame Id) 0x88: macKeyDeviceDescriptorHandle (Handle to the DeviceDescriptor corresponding to the device) 0x89: macUniqueDevice (MAC Unique Device) 0x8A: macBlackListed (MAC Blacklisted) 0x8B: macSecLevFrameType (MAC Security Level Frame Type) 0x8C: macSecLevCommandFrameIdentifier (MAC Security Level Command Frame Identifier) 0x8D: macSecLevSecurityMinimum (MAC - Minimum Security Level) 0x8E: macSecLevDeviceOverrideSecurityMinimum (MAC - Minimum Security Level) 0x8F: macDeviceDescriptorPanId (MAC Device Descriptor PAN id) 0x90: macDeviceDescriptorShortAddress (MAC Device Descriptor Short Address) 0x91: macDeviceDescriptorExtAddress (MAC Device Descriptor Extended Address) 0x92: macDeviceDescriptorFrameCounter (MAC Device Descriptor Frame Counter) 0x93: macDeviceDescriptorExempt (MAC Device Descriptor Exempt) 0x94: macKeyIdLookupData (MAC Key Id Lookup Data) 0x95: macKeyIdLookupDataSize (MAC Key Id Lookup Data Size) 0x96: macKeyTableCrtEntry (MAC Key Table - Current Entry) 0x97: macDeviceTableCrtEntry (MAC Device Table - Current Entry) 0x98: macSecurityLevelTableCrtEntry (MAC Security Level Table - Current Entry) 0x99: macKeyIdLookupListCrtEntry (MAC Key Id Lookup List - Current Entry) 0x9A: macKeyUsageListCrtEntry (MAC Key Usage List - Current Entry) 0x9B: macKeyDeviceListCrtEntry (MAC Key Device List - Current Entry) 0x25: macNBSuperFrameInterval (Non-beacon mode superframe interval) 0x22: macTreemodeStartTime (Treemode Start Time) 0x23: macPanIdConflictDetection (PanId Conflict Detection) 0x27: macBeaconResponseLQIThreshold (Beacon response LQI Threshold) 0x28: BeaconEDCfpStartTrimVaue (Beacon Mode ED CFP Start Trim Value)

**Table 87. MacSetPIBAtribute.Request parameters**

Parameter	Size (bytes)	Comments
Index	1	Index PIB Table entries (not used)
Value	16	The value to set the attribute to

### 3.3.37. MacSetPIBAtribute.Confirm

#### Description

Returns the status of the MLME-SET.Request

#### Parameters

**Table 88. MacSetPIBAtribute.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x0D
Length	1/2	Length in bytes of the following parameters
Status	1	The result of the MacSetPIBAtribute.Request Possible values: 0x00: gSuccess_c (The PhySetPIBAtribute.Request operation was successful) 0xF4: UNSUPPORTED_ATTRIBUTE (Unknown PIB attribute) 0xE8: gInvalidParameter_c (Unknown PIB attribute) 0xFB: gReadOnly_c (Read Only Attribute)
PIBAtribute	1	The MAC PIB attribute identifier (see the parameter table of the MacSetPIBAtribute.Request)

### 3.3.38. MacSetSlotframe.Request

#### Description

The MLME-SET-SLOTFRAME.request primitive is generated by the next higher layer, and issued to the MLME to request a set of a TSCH slotframe to the MAC sublayer.

#### Parameters

**Table 89. MacSetSlotframe.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x0D
Length	1/2	Length in bytes of the following parameters
SlotframeHandle	1	Unique identifier of the slotframe
Operation	1	Operation to perform on the slotframe. ADD = 0, DELETE = 2, MODIFY = 3
Size	2	Number of timeslots in the new slotframe

### 3.3.39. MacSetSlotframe.Confirm

#### Description

The MLME-SET-SLOTFRAME.confirm primitive is generated by the MAC layer and issued to the next higher layer to confirm the set of a TSCH slotframe to the MAC sublayer.

#### Parameters

**Table 90. MacSetSlotframe.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x15
Length	1/2	Length in bytes of the following parameters
SlotframeHandle	1	Unique identifier of the slotframe to be added, deleted, or modified
Status	1	Indicates result of the MLME-SET-SLOTFRAME.request

### 3.3.40. MacStart.Request

#### Description

Start an 802.15.4 PAN

#### Parameters

**Table 91. MacStart.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x0A
Length	1/2	Length in bytes of the following parameters
PANId	2	The PAN ID to be used by the beacon
LogicalChannel	1	The logical channel on which to start transmitting beacons
StartTime	4	The time at which to begin transmitting beacons
BeaconOrder	1	The beacon order
SuperframeOrder	1	The superframe order
PANCoordinator	1	Device will be PAN coordinator if true
BatteryLifeExtension	1	Battery life extension enabled
CoordRealign	1	Transmit coordinator realignment command
CoordRealignSecurityLevel	1	The security level for the realign
CoordRealignKeyIdMode	1	The key identifier mode for the realign
CoordRealignKeySource	8	The key source for the realign
CoordRealignKeyIndex	1	The key index for the realign
BeaconSecurityLevel	1	The security level for the beacons
BeaconKeyIdMode	1	The key identifier mode for the beacons
BeaconKeySource	8	The key source for the beacons
BeaconKeyIndex	1	The key index for the beacons

### 3.3.41. MacStart.Confirm

#### Description

Return the status of the MLME\_START.Request

#### Parameters

**Table 92. MacStart.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x0E
Length	1/2	Length in bytes of the following parameters
Status	1	The status of the start attempt Possible values: 0x00: gSuccess_c (Success) 0xE4: gFailedSecurityCheck_c (The received frame failed security check) 0xE5: gFrameTooLong_c (The frame was too long to send after security processing) 0xE8: gInvalidParameter_c (A parameter in the primitive is out of the valid range) 0xEC: gNoShortAddress_c (No short address was allocated) 0xF3: gUnavailableKey_c (The appropriate key is not available in the ACL)

### 3.3.42. MacSync.Request

#### Description

Synchronize with the Coordinator's Beacon

#### Parameters

**Table 93. MacSync.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x0B
Length	1/2	Length in bytes of the following parameters
LogicalChannel	1	The logical channel on which to attempt synchronization
TrackBeacon	1	Track the beacon

### 3.3.43. MacSyncLoss.Indication

#### Description

Inform the upper layer that the device is not synchronized with the Coordinator's Beacon.

#### Parameters

**Table 94. MacSyncLoss.Indication parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x0F
Length	1/2	Length in bytes of the following parameters
LossReason	1	Possible values: 0xEE: gPanIdConflict_c (PAN ID conflict ) 0xEF: gRealignment_c (Coordinator realignment) 0xE0: gBeaconLoss_c (Beacon lost)
PanID	2	The PanID with which the device lost synch
LogicalChannel	1	The logical channel on which the device lost synch
securityLevel	1	The security level for the packet
keyIdMode	1	The key identifier mode for the packet
keySource	8	The key source for the packet
keyIndex	1	The key index for the packet

### 3.3.44. MacPromiscuous.Indication

#### Description

A data packet was received over the air when the MAC is in promiscuous mode.

#### Parameters

**Table 95. MacPromiscuous.Indication parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x86
OpCode	1	0x03
Length	1/2	Length in bytes of the following parameters
LinkQuality	1	Link Quality
TimeStamp	4	Time Stamp
msduLength	1	Length of the frame
msdu	1 x msduLength	The actual frame received

### 3.3.45. MacTschMode.Request

#### Description

The MLME-TSCH-MODE.request primitive is generated by the next higher layer, and issued to the MLME to request the TSCH enable or disable procedure.

#### Parameters

**Table 96. MacTschMode.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x85
OpCode	1	0x0F
Length	1/2	Length in bytes of the following parameters
TSCHMode	1	Target mode ON = 1: TSCH mode is to be started, OFF = 0: TSCH mode is to be stopped

### 3.3.46. MacTschMode.Confirm

#### Description

The MLME-TSCH-MODE.confirm primitive is generated by the MAC layer, and issued to the next higher layer to confirm the enable or disable of the TSCH module.

#### Parameters

**Table 97. MacTschMode.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x84
OpCode	1	0x17
Length	1/2	Length in bytes of the following parameters
TSCHMode	1	Target mode ON = 1: TSCH mode is to be started, OFF = 0: TSCH mode is to be stopped
Status	1	Indicates results of the MLME-TSCH-MODE.request

## 3.4. ASP messages

### 3.4.1. ASP-GetTime.Request

#### Description

Read the 802.15.4 free running timer.

#### Parameters

**Table 98. ASP- GetTime.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x00
Length	1/2	0x00

### 3.4.2. ASP- GetTime.Confirm

#### Description

Returns 802.15.4 timestamp in symbols.

#### Parameters

**Table 99. ASP- GetTime.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x00
Length	1/2	Length in bytes of the following parameters
Timestamp	4/8	Timestamp in symbols.

### 3.4.3. ASP-SetPowerLevel.Request

#### Description

Set the TX power level.

#### Parameters

**Table 100. ASP- SetPowerLevel.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x0F
Length	1/2	0x01
powerLevel	1	

### 3.4.4. ASP- SetPowerLevel.Confirm

#### Description

Confirm that the requested power level have been set correctly or not.

#### Parameters

**Table 101. ASP- SetPowerLevel.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x0F
Length	1/2	Length in bytes of the following parameters
Status	1	The possible values are: 0x00: SUCCESS (Request successfully performed) 0xE8: INVALID_PARAMETER (a parameter is invalid or the request is not allowed at the moment).



### 3.4.5. ASP- GetPowerLevel.Request

#### Description

Get the current TX power level.

#### Parameters

**Table 102. ASP- GetPowerLevel.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x1F
Length	1/2	0x00

### 3.4.6. ASP- GetPowerLevel.Confirm

#### Description

Returns the value of the current TX power level.

#### Parameters

**Table 103. ASP- GetPowerLevel.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x1F
Length	1/2	Length in bytes of the following parameters
powerLevel	1	

### 3.4.7. ASP-GetRSSILevel.Request

#### Description

Read the current RSSI value.

#### Parameters

**Table 104. Asp-GetRSSILevel.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x23
Length	1/2	0x00 - This message does not have any parameters

### 3.4.8. ASP-GetRSSILevel.Confirm

#### Description

Return the RSSI value

#### Parameters

**Table 105. Asp-GetRSSILevel.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x23
Length	1/2	Length in bytes of the following parameters
Value	1	The Value of the RSSI Level

### 3.4.9. ASP-SetLQIMode.Request

#### Description

Change the LQI mode

#### Parameters

**Table 106. Asp-SetLQIMode.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x22
Length	1/2	Length in bytes of the following parameters
LQImode	1	1 - LQI based on RSSI/ 0 - LQI based on Correlation peaks 0x00: LQI based on Correlation peaks 0x01: LQI based on RSSI

### 3.4.10. ASP-SetLQIMode.Confirm

#### Description

Returns the status of the request

#### Parameters

**Table 107. Asp-SetLQIMode.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x22
Length	1/2	Length in bytes of the following parameters
Status	1	The status of ANTState request Possible values: 0x00: SUCCESS (Request successfully performed) 0xE2: INVALID_PARAMETER (A parameter is invalid or the primitive is not allowed at the moment)

### 3.4.11. ASP-GetMpmConfig.Request

#### Description

Get the state of the Multi PAN Manager (MPM)

#### Parameters

**Table 108. Asp-GetMpmConfig.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x25
Length	1/2	0x00 - This message does not have any parameters

### 3.4.12. ASP-GetMpmConfig.Confirm

#### Description

Returns the current state of the MPM, or an error code.

#### Parameters

**Table 109. Asp-GetMpmConfig.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x25
Length	1/2	Length in bytes of the following parameters
Status	1	Error code
AutoMode	1	The state of DualPan Auto mode
dwellTime	1	DualPan Auto mode PAN switch time
activeMAC	1	The instance of the Active MAC

### 3.4.13. ASP-SetMPMConfig.Request

#### Description

Configure the Multi PAN Manager (MPM).

One should first use ASP-GetMpmConfig, make the desired changes, then call ASP-SetMpmConfig.

#### Parameters

**Table 110. ASP- SetMPMConfig.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x24
Length	1/2	Length in bytes of the following parameters
AutoMode	1	The state of DualPan Auto mode
dwellTime	1	DualPan Auto mode PAN switch time
activeMAC	1	The instance of the Active MAC

### 3.4.14. ASP- SetMPMConfig.Confirm

#### Description

Confirm that the requested have been processed successfully.

#### Parameters

**Table 111. ASP- SetMPMConfig.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x24
Length	1/2	Length in bytes of the following parameters
Status	1	The possible values are: 0x00: SUCCESS (Request successfully performed) or an error code

### 3.4.15. ASP-GetANTXState.Request

#### Description

Get the state of the ANT-X feature.

#### Parameters

**Table 112. ASP- GetANTXState.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x21
Length	1/2	0x00

### 3.4.16. ASP- GetANTXState.Confirm

#### Description

Returns the selected antenna.

#### Parameters

**Table 113. ASP- GetANTXState.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x21
Length	1/2	Length in bytes of the following parameters
value	1	Chosen antenna by the FAD (when FAD_EN=1) or copy of ANT_X_IN. Possible values: 0x00: ANT0 (Disabled) 0x01: ANT1 (Enabled)

### 3.4.17. ASP-SetANTXState.Request

#### Description

Select the working antenna.

#### Parameters

**Table 114. ASP- SetANTXState.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x20
Length	1/2	Length in bytes of the following parameters
ANTXstate	1	Starting antenna (FAD_EN=1) /Used antenna (FAD_EN=0)

### 3.4.18. ASP- SetANTXState.Confirm

#### Description

Returns the status of the request.

#### Parameters

**Table 115. ASP- SetANTXState.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x20
Length	1/2	Length in bytes of the following parameters
status	1	Error code

### 3.4.19. ASP-GetFADState.Request

#### Description

Request the state of the Fast Antenna Diversity (FAD) feature.

#### Parameters

**Table 116. ASP- GetFADState.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x1A
Length	1/2	0x00

### 3.4.20. ASP- GetFADState.Confirm

#### Description

Returns the state of the Fast Antenna Diversity (FAD) feature.

#### Parameters

**Table 117. ASP- GetFADState.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x1A
Length	1/2	Length in bytes of the following parameters
state	1	Possible values for FAD feature's state are: 0x00: FAD is disabled 0x01: FAD is enabled

### 3.4.21. ASP-GetFADThreshold.Request

#### Description

Request the correlation factor used for selecting an antenna during the preamble phase, when Fast Antenna Diversity (FAD) feature is enabled.

#### Parameters

**Table 118. ASP- GetFADThreshold.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x19
Length	1/2	0x00

### 3.4.22. ASP- GetFADThreshold.Confirm

#### Description

Get the correlation factor used for selecting an antenna during the preamble phase, when Fast Antenna Diversity feature is enabled.

#### Parameters

**Table 119. ASP- GetFADThreshold.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x19
Length	1/2	Length in bytes of the following parameters
value	1	The value of the correlation factor used for selecting an antenna during the preamble phase (FAD Threshold).

### 3.4.23. ASP-SetFADState.Request

#### Description

Enable or disable the Fast Antenna Diversity (FAD) feature

#### Parameters

**Table 120. ASP- SetFADState.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x17
Length	1/2	0x00
FADEnable	1	The possible values are: 0x00: request to disable FAD feature 0x01: request to enable FAD feature

### 3.4.24. ASP- SetFADState.Confirm

#### Description

Confirm that the current state of the Fast Antenna Diversity is the requested one.

#### Parameters

**Table 121. ASP- SetFADState.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x17
Length	1/2	Length in bytes of the following parameters
Status	1	The possible values are: 0x00: SUCCESS (Request successfully performed) 0xE8: INVALID_PARAMETER (a parameter is invalid or the request is not allowed at the moment).

### 3.4.25. ASP-SetFADThreshold.Request

#### Description

Request to set the correlation factor used for selecting an antenna during the preamble phase, when Fast Antenna Diversity (FAD) feature is enabled.

#### Parameters

**Table 122. ASP- SetFADThreshold.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x18
Length	1/2	0x00
FADThreshold	1	The correlation factor necessary for an antenna to be selected during the preamble phase. This antenna is also used for the next transmission.



### 3.4.26. ASP- SetFADThreshold.Confirm

#### Description

Confirm that the correlation factor used for selecting an antenna during the preamble phase was set according to the request.

#### Parameters

**Table 123. ASP- SetFADThreshold.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x18
Length	1/2	Length in bytes of the following parameters
Status	1	The possible values are: 0x00: SUCCESS (Request successfully performed) 0xE8: INVALID_PARAMETER (a parameter is invalid or the request is not allowed at the moment).

### 3.4.27. ASP-TelecSetFreq.Request

#### Description

Set the 802.15.4 channel for the Telec test mode.

#### Parameters

**Table 124. Asp-TelecSetFreq.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x11
Length	1/2	Length in bytes of the following parameters
channel	1	Channel to set the device on

### 3.4.28. ASP-TelecSetFreq.Confirm

#### Description

Return the status of the request.

#### Parameters

**Table 125. Asp-TelecSetFreq.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x11
Length	1/2	Length in bytes of the following parameters
Status	1	The status of the Telec Set Frequency Possible values: 0x00: SUCCESS (Request successfully performed) 0xE8: INVALID_PARAMETER (A parameter is invalid or the primitive is not allowed at the moment)

### 3.4.29. ASP-TelecSendRawData.Request

#### Description

Send a raw data packet over the air using the Telec test mode.

#### Parameters

**Table 126. Asp-TelecSendRawData.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x16
Length	1/2	Length in bytes of the following parameters
DataLength	1	Max length is 125, min is 3
Data	DataLength	PSDU Data

### 3.4.30. ASP-TelecSendRawData.Confirm

#### Description

Return the status of the request

#### Parameters

**Table 127. Asp-TelecSendRawData.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x16
Length	1/2	Length in bytes of the following parameters
Status	1	Possible values: 0x00: SUCCESS (Request successfully performed) 0xC3: INVALID_REQUEST (Request not valid)

### 3.4.31. ASP-TelecTest.Request

#### Description

Set the XCVR test mode.

#### Parameters

**Table 128. Asp-TelecTest.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x10
Length	1/2	Length in bytes of the following parameters
mode	1	Possible values: 0x00: TestForceIdle (Force Radio Idle State) 0x01: TestPulseTxPrbs9 (Continuously transmit a PRBS9 pattern) 0x02: TestContinuousRx (Sets the device into continuous RX mode) 0x03: TestContinuousTxMod (Sets the device to continuously transmit a 10101010 pattern) 0x04: TestContinuousTxNoMod (Sets the device to continuously transmit an unmodulated CW)

### 3.4.32. ASP-TelecTest.Confirm

#### Description

Return the status of the request

#### Parameters

**Table 129. Asp-TelecTest.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x10
Length	1/2	Length in bytes of the following parameters
Status	1	Possible values: 0x00: SUCCESS (Request successfully performed) 0xE8: INVALID_PARAMETER (A parameter is invalid or the primitive is not allowed at the moment)

### 3.4.33. ASP-XcvrRead.Request

#### Description

Read XCVR registers.

#### Parameters

**Table 130. ASP- XcvrRead.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x1D
Length	1/2	0x00
Mode	1	Read mode
Address	2	Memory location from where the data will be read
DataLength	1	Number of bytes to be read

### 3.4.34. ASP- XcvrRead.Confirm

#### Description

Returns the value of the requested XCVR register(s).

#### Parameters

**Table 131. ASP- XcvrRead.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x1D
Length	1/2	Length in bytes of the following parameters
Status	1	Status of the command
DataLength	1	Length of the data
Data	1 x DataLength	Data

### 3.4.35. ASP-XcvrWrite.Request

#### Description

Write XCVR registers.

#### Parameters

**Table 132. ASP- XcvrWrite.Request parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x95
OpCode	1	0x1C
Length	1/2	0x00
Mode	1	Read mode
Address	2	Memory location from where the data will be read
DataLength	1	Number of bytes to be read
Data	1 x Length	Data

### 3.4.36. ASP- XcvrWrite.Confirm

#### Description

Returns the status of the XCVR write operation.

#### Parameters

**Table 133. ASP- XcvrRead.Confirm parameters**

Parameter	Size (bytes)	Comments
OpGroup	1	0x94
OpCode	1	0x1C
Length	1/2	Length in bytes of the following parameters
Status	1	Status of the command

## 4. References

- *IEEE Std 802.15.4-2006*—IEEE Standard for Information technology – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs) (available at <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=11161>)
- *802.15.4 MAC PHY Software Reference Manual* (document [802154MPSRM](#))

## 5. Revision history

**Table 134. Revision history**

Revision number	Date	Substantive changes
0	08/2015	Initial release

---

**How to Reach Us:**

**Home Page:**

[freescale.com](http://freescale.com)

**Web Support:**

[freescale.com/support](http://freescale.com/support)

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [freescale.com/SalesTermsandConditions](http://freescale.com/SalesTermsandConditions).

Registered trademarks: Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off.

ARM and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. IAR Embedded Workbench is a trademark of IAR Systems (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2015 Freescale Semiconductor, Inc.

Document Number: 802154FSCIRM  
Rev. 0  
08/2015

