IEEE 802.15.4 MAC

User Guide

Section 1- Introduction

Section 2- Device Overview

Section 3- Using MAC Services

Appendix A - MAC Request Functions

Appendix B - MAC Confirm Callback Functions

Appendix C - MAC Indication Callback Functions

Section 1 INTRODUCTION

The following paragraphs introduce this User Guide for the Atmel designed Medium Access Control (MAC) layer utilized in an IEEE 802.15.4 (hereinafter referred to as the IEEE standard). The subject, purpose, scope, and plan of development of this user guide is provided as well as a brief background of the IEEE standard and the ZigBee™ Alliance. A list of supporting documentation is also provided.

1.1 Subject

This user guide describes the interface between the application/network and the MAC layer as implemented by Atmel's IEEE 802.15.4 compliant firmware.

1.2 Purpose

This guide provides the application programmer with a technical description of the services provided by the MAC. MAC and callback functions are described in detail so that the programmer can design a program that can fully utilize the services of the MAC.

1.3 Scope

This document is not intended to replace the IEEE standard. The focus of this material contained within is on the interface between the Atmel® MAC and the user's application. For additional details regarding the IEEE standard, the user is encouraged to reference that document.

1.4 Plan of Development

This user guide approaches the interface to the MAC by presenting an overview of the MAC and its interfacing layers. A section on using the MAC services presents an introduction to the function calls used to invoke MAC services and the callbacks that the MAC returns to the application/network layer. The necessary source and include files are listed as well as a presentation of the PIB. The main function is discussed and how to reset the device is presented. Appendixes A through D contain lists of function call semantics and parameters. Appendix E presents a list of all data elements and their detailed parameters.

1.5 Background

Wireless personal area networks (WPAN) are used to convey information over relatively short distances. Unlike wireless local area networks (WLAN), connections effected by way of WPANs involve little or no infrastructure. This feature allows small, power-efficient, inexpensive solutions to be implemented for a wide range of devices. IEEE 802.15.4 defines a standard for a low-rate WPAN (LR-WPAN).

The ZigBee Alliance is developing standards for the application and network layers that will use the LR-WPAN defined by the IEEE standard. OEMs are designing these layers to accommodate their needs and their customer’s needs. The application and network layers can theoretically be anything the user desires. From point-to-point (star) networks that report temperatures to self healing multi-hop mesh networks that not only

report, but also manage useful devices. Radio control toys that not only interface with the human controller, but also interact with other networked toys are a possibility. The number of applications are limitless.

1.6 Supporting Documentation

The programmer should consult the following documents when more detailed information is required.

- IEEE 802.15.4
- MAC stack library and header files*
- MAC API Reference Document*

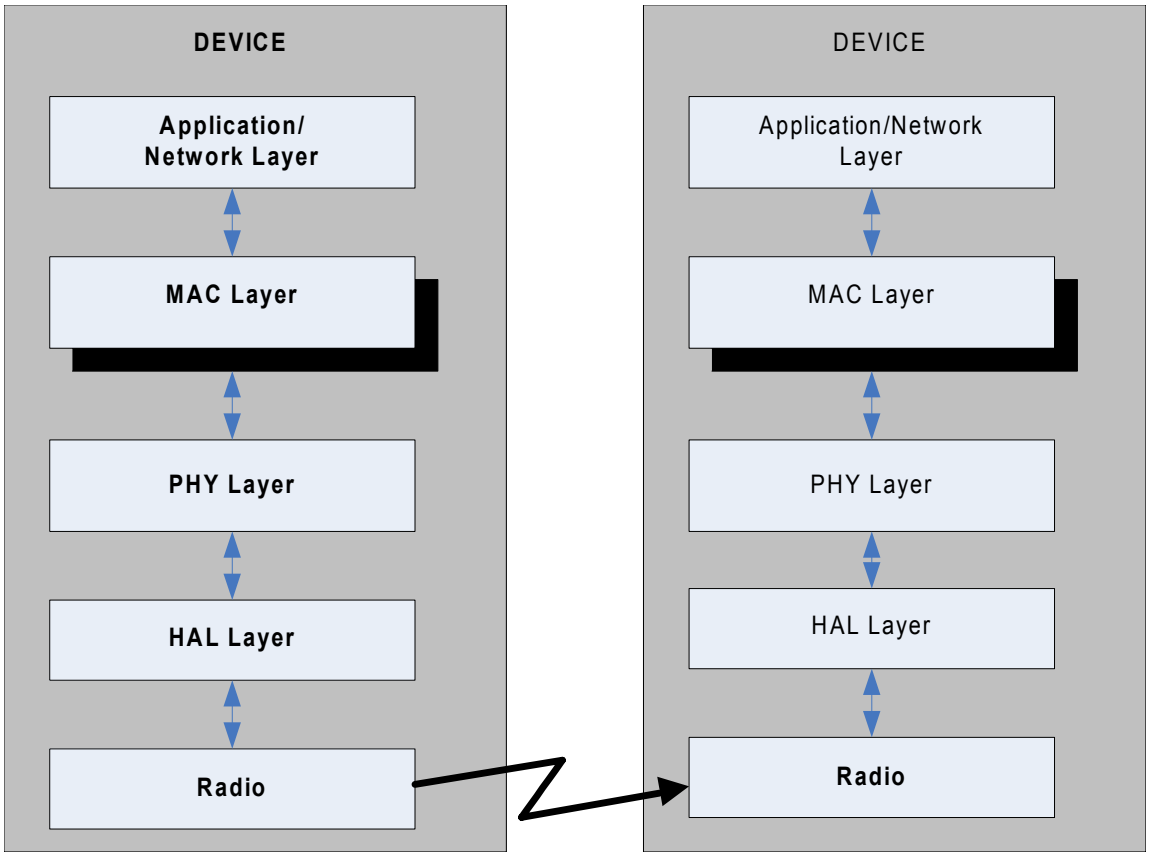
(*These documents are provided as part of the AT86RF230 documentation. See http://www.atmel.com/products/avr/z-link/default.asp)

Section 2 DEVICE OVERVIEW

IEEE 802.15.4 describes two device types: a Full Function Device (FFD) and a Restricted Function Device (RFD). The Atmel MAC stack was designed for use in both FFDs and RFDs. Differentiation between device types occurs at the user's application and network software layers as well as at the user's hardware implementation layer.

As the International Standards Organization (ISO) has created a communications network model, the IEEE standard has modified that model to suit its needs. This guide uses a simplified and modified model to describe the communications layers in a device. This simplified model is depicted in Figure 2-1.

Figure 2-1. Simplified Device Model



2.1 The Application/ Network Layer

The application and network layers of the software solution are necessary components of the final software solution. They comprise the highest levels of the software control hierarchy and direct the Atmel MAC to perform lower level functions. Proper definition and implementation of these layers ultimately allow the user to realize the final application.

2.2 The MAC Layer

The MAC layer handles all access to the physical radio channel and is responsible for the following tasks:

- Generating network beacons if the device is a coordinator
- Synchronizing the beacons
- Supporting PAN association and disassociation
- Supporting device security
- Employing the CSMA-CA mechanism for channel access
- Handling and maintaining the GTS mechanism
- Providing a reliable link between two peer MAC entities

2.3 The Physical Layer

IEEE 802.15.4 defines the Physical Layer (PHY) as responsible for the following tasks:

- Activation and deactivation of the radio transceiver
- ED within the current channel
- LQI for received packets
- CCA for CSMA-CA
- Channel frequency selection
- Data transmission and reception

2.4 The Hardware Abstraction Layer

The Hardware Abstraction Layer (HAL) is specific to a particular radio and interfaces the common PHY layer to the physical radio. Due to the number of radios that are available in the marketplace, Atmel engineers have designed and added the HAL as a module that is changeable to meet the differing requirements of the various radios.

Section 3 USING MAC SERVICES

MAC services and their usage are described in the following paragraphs. The services are overviewed and then a detailed description of the services is presented. Header and source files are listed and the MAC main task is described. The programming concepts for accessing the PIB and resetting the device are discussed.

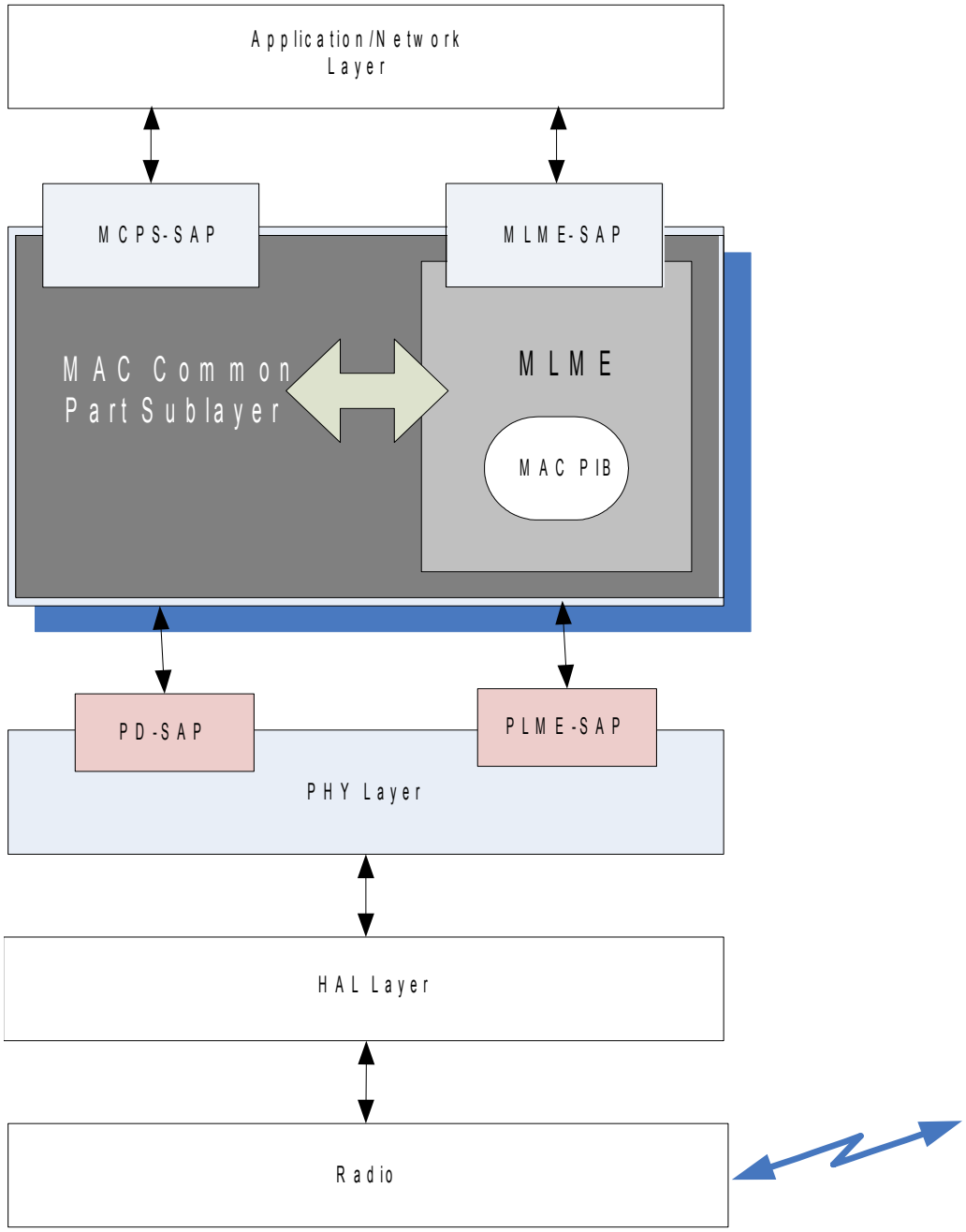
3.1 Overview

The MAC layer provides an interface between the application layer and the PHY layer. The MAC layer provides services to the application layer through two groups: the MAC Management Service (called the MAC Layer Management Entity, or MLME) and the MAC Data Service (called the MAC Common Part Layer, or MCPS).

The MCPS provides data transport services between peer MACs. The MLME provides the service interfaces through which layer management functions may be invoked. The MLME is also responsible for maintaining a database of managed objects pertaining to the MAC layer. This database is referred to as the MAC layer PAN information base (PIB). The MLME also has access to MCPS services for data transport.

A more graphic representation of the MAC layer reference model is depicted in Figure 3-1.

Figure 3-1. MAC layer reference model

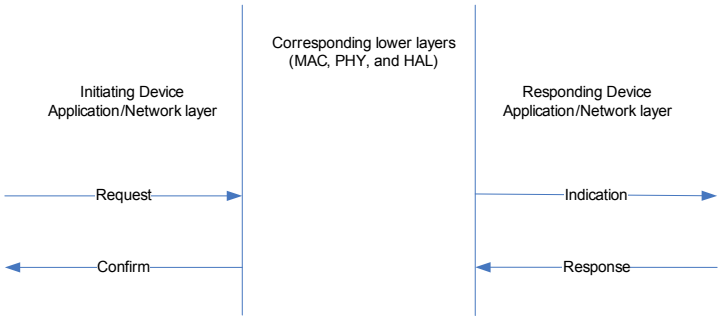


MAC services are invoked by preparing a formatted message and calling the appropriate MCPS or MLME function to process the contents of that message. The appropriate function will respond by way of formatted callback messages that indicate various statuses and completion of the call.

3.2 MAC function calls

The services provided by the MAC are the capabilities it offers to the Application/Network layers. This concept is illustrated in Figure 3-2, showing the relationship of the Application/Network layer communicating through the corresponding MAC and lower layers.

Figure 3-2. Layer Services



The MAC function calls are specified by describing the information flow between the Application/Network layer and the MAC. This information flow is modeled by discrete, instantaneous events, that characterize the provision of a function. Each event consists of passing a function call between the layers. Function calls convey the required information by providing a particular service. These function calls are an abstraction because they specify only the provided service rather than the means by which it is provided. This definition is independent of any other interface implementation. Services are specified by describing the function call and parameters that characterize it. A function may have one or more related calls that constitute the activity that is related to that particular function. Each function call may have zero or more parameters that convey the information required to provide the service.

A function call can be one of four generic types:

- Request: The request primitive is passed down from the Application/Network layer to request that a service is initiated by the MAC.
- Indication: The indication primitive is passed up from the MAC. This event may be logically related to a remote service request, or it may be caused by an internal MAC event.
- Response: The response primitive is passed from the Application/Network layer to the MAC to complete a procedure previously invoked by an indication primitive.
- Confirm: The confirm primitive is passed from the MAC to the Application/Network layer to convey the results of one or more associated previous service requests.

MAC functions are prefixed by wpan_ and callback functions are prefixed by usr_. These prefixes are listed in Figure 3-1, further described in the following paragraphs, and in Appendix A, B, C, and D. A detailed list of the function parameters (Data Dictionary) is shown in Appendix E.

Table 3-1. Function prefixes

	Function and Message Suffixes			
Name	**Request**	**Confirm**	**Response**	**Indication**
MCPS_DATA_	wpan_	usr_		usr_
MCPS_PURGE_	wpan_	usr_		
MLME_ASSOCIATE_	wpan_	usr_	wpan_	usr_
MLME_DISASSOCIATE_	wpan_	usr_		usr_
MLME_BEACON-NOTIFY_		usr_		
MLME_GET_	wpan_			usr_
MLME_GTS_	wpan_	usr_		usr_
MLME_ORPHAN_		usr_	wpan	
MLME_RESET_	wpan_			usr_
MLME_RX-ENABLE_	wpan_			usr_
MLME_SCAN_	wpan_			usr_
MLME_COMM-STATUS_		usr_		
MLME_SET_	wpan_			usr_
MLME_START_	wpan_			usr_
MLME_SYNC_	wpan_			
MLME_SYNC-LOSS_		usr_		
MLME_POLL_	wpan_			usr_

3.2.1 Function calls

MAC functions have to be called from the application in order to initiate an action in the communication stack at the MAC level. MAC functions are prefixed with wpan_ and suffixed with either _request or _response. A sample construct of a call to a MAC function is: wpan_mlme_associate_request.

3.2.2 Callback functions

When the MAC needs to invoke a function in the application, it calls a callback function. If the callback function is not implemented by the application, it will be replaced by an empty function from the library. Callback functions are prefixed with usr_ and suffixed with either _confirm or _indication. A sample construct of a callback to an application function is: usr_mlme_associate_indication.

3.3 Include files

The following include files should be refered to as applicable. They are contained in the accompanying CDROM.

- ieee_const.h – This header holds all IEEE 802.15.5 attributes.
- ieee_types.h - This file declares IEEE data structures.
- wpan.h – Function definitions for API into Atmel's 802.15.4 implementation.
- wpan_defines.h – High level API definitions.

3.4 Source files

The stack library and header files for the MAC are contained in the accompanying CDROM.

3.5 MAC main task

An example main routine for an application using the MAC library is shown below.

```
#include "wpan_defines.h"
#include "wpan.h"
#include "ieee_const.h"
#include "ieee_types.h"
int main(void)
{
    wpan_init();
    wpan_mlme_reset_request( true );
    while(1)
    {
        while(wpan_task())
        {
            /* only call functions with short runtime here */
            get_keys();
        }
        /* longer running tasks are called here */
        do_nwk_fsm();
        process_keys();
    }
    /* this return is (hopefully) never executed */
    return 0xff;
}
```

3.6 PIB constants

For more details, refer to the "Atmel MAC Reference Manual", available online and supplied with Z-Link Application Tool Kits.

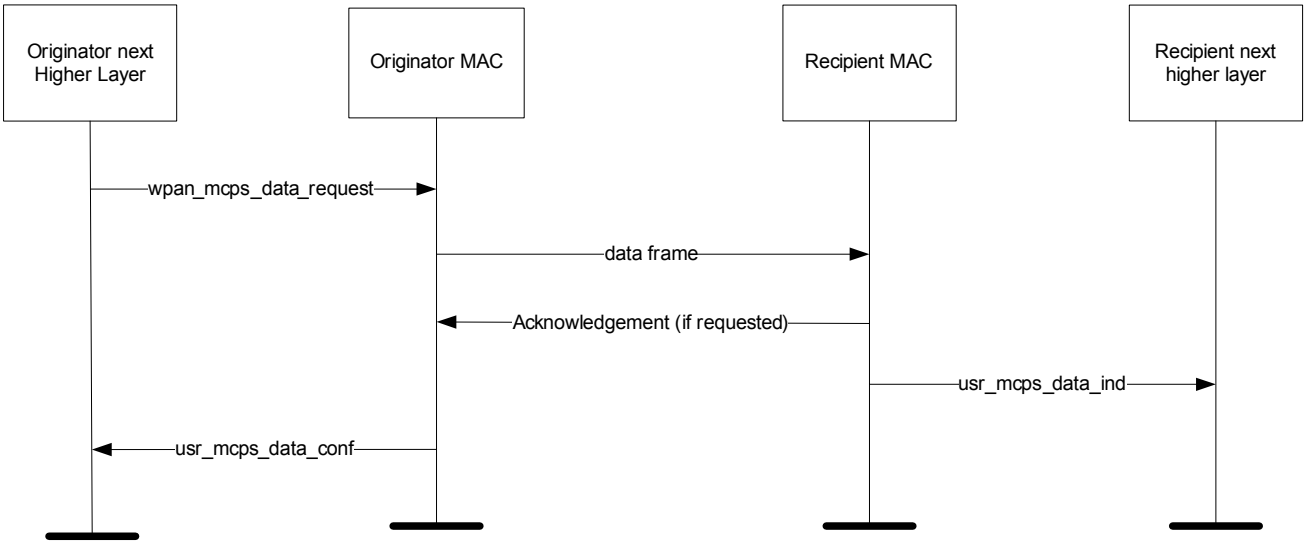
3.7 Message sequence charts

Typical message flow in support of a specific task is shown in the following paragraphs.

3.7.1 Data service message sequence charts

Figure 3-3 illustrates the sequence of messages necessary for a successful data transfer between two devices.

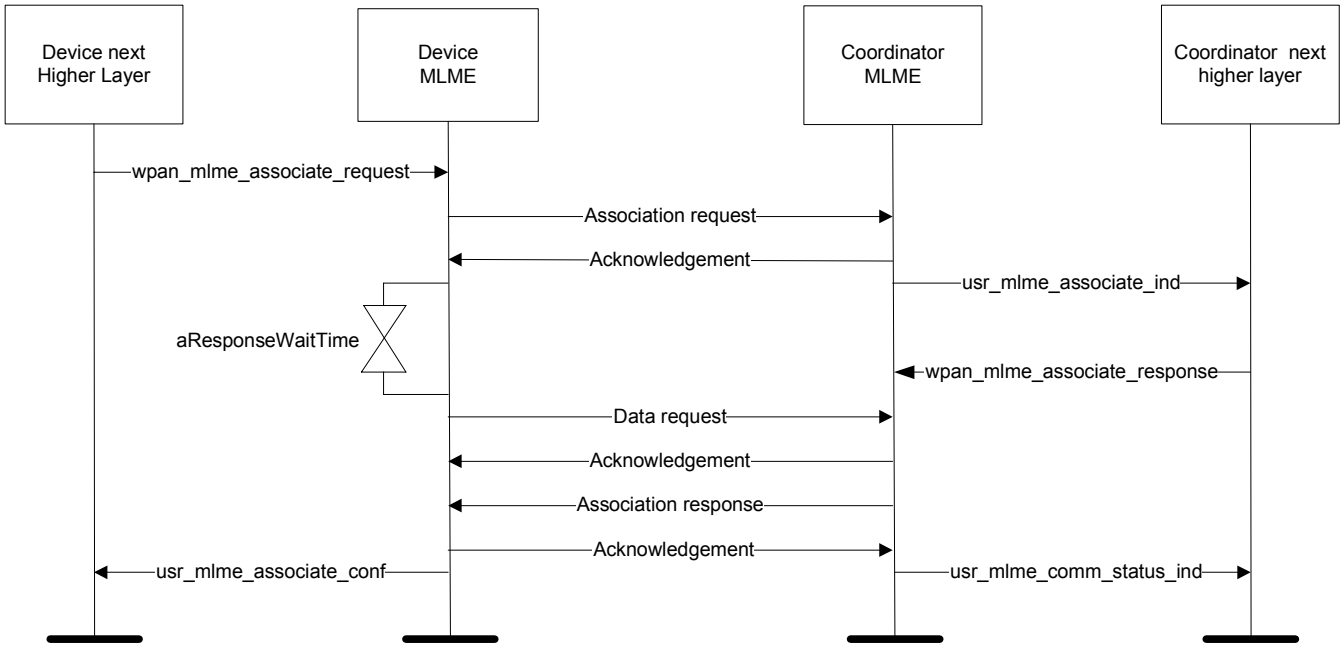
Figure 3-3. Message sequence chart describing the MAC data service



3.7.2 Association message sequence charts

Figure 3-4 illustrates the sequence of messages necessary for a device to successfully associate with a PAN.

Figure 3-4. Message sequence chart for association

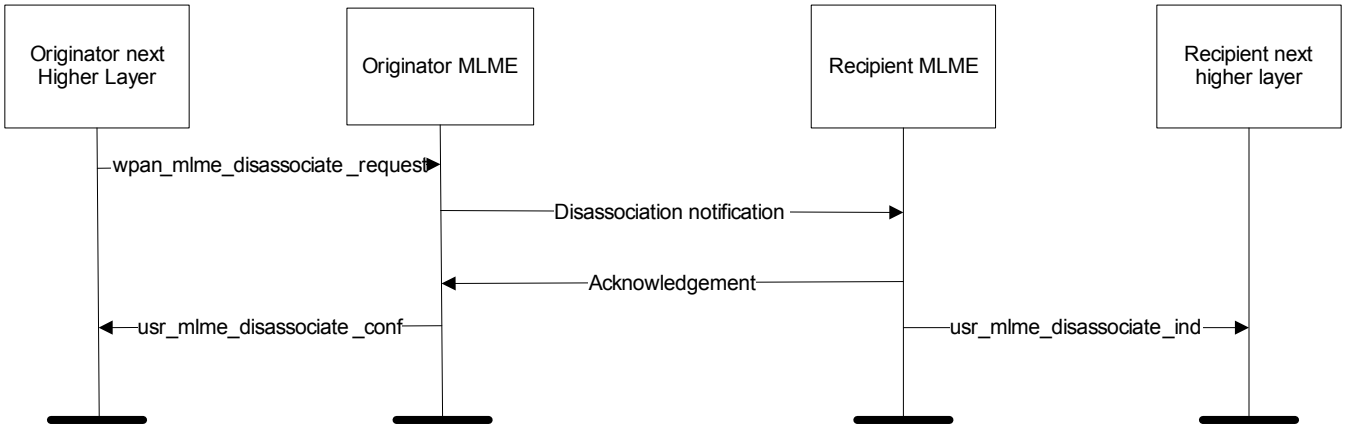


3.7.3 Disassociation message sequence charts

Figure 3-5 illustrates the sequence of messages necessary for successful disassociation from a PAN. The originating

device may be either a device or the coordinator to which the device has associated.

Figure 3-5. Message sequence chart for disassociation



3.7.4 GTS management message sequence charts

Figure 3-6 and Figure 3-7 illustrate the sequence of messages necessary for successful GTS management. The first depicts the message flow for the case in which the device initiates the GTS allocation. The second depicts the message flow for the two cases for which a GTS deallocation occurs, first, by a device (a) and, second, by the PAN coordinator (b).

Figure 3-6. Message sequence chart for GTS allocation initiated by a device

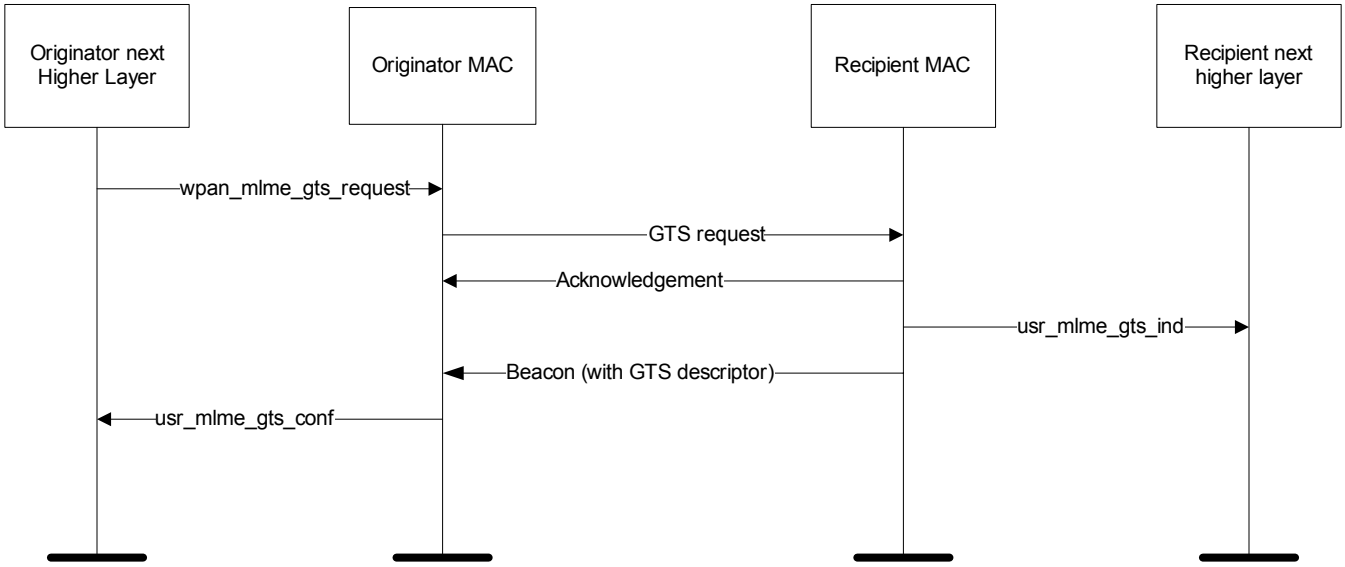
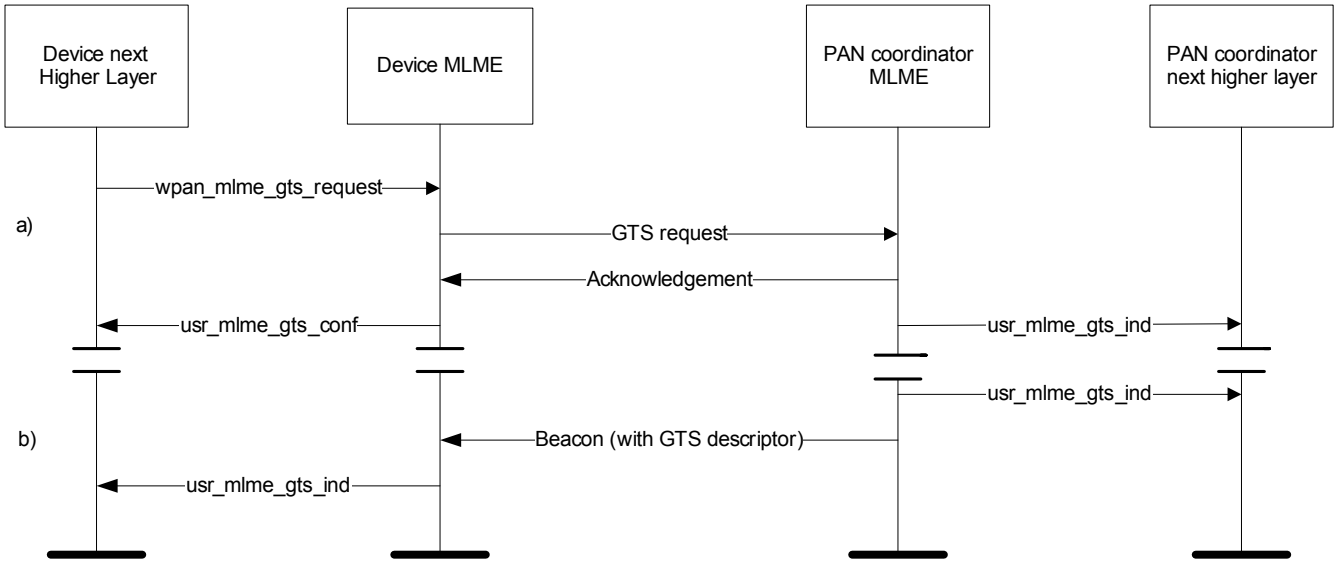


Figure 3-7. Message sequence chart for GTS deallocation initiated by a device (a) and the PAN coordinator (b)



Appendix A MAC Request Functions

The following functions are called from the application/network layer and implemented in the MAC. These request functions are listed in alphabetical order.

A.1 MLME ASSOCIATE Request

Allows a device to request an association with a coordinator.

A.1.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mlme_associate_request(
                                    LogicalChannel,
                                    CoordAddressMode,
                                    CoordPANId,
                                    CoordAddress,
                                    CapabilityInfo,
                                    SecurityEnable
                                    );
```

A.1.2 Expected results of the call

MCPS ASSOCIATE Confirm message with a status of SUCCESS.

A.2 MCPS DATA Request

This request causes an MSDU to be transmitted.

A.2.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mcps_data_request (
                                addrInfo,
                                msduHandle,
                                tx_options
                                data,
                                data_length
                                );
```

A.2.2 Expected results of the call

MCPS DATA Confirm message with a status of SUCCESS.

A.3 MLME DISASSOCIATE Request

Used by an associated device to notify the coordinator of its intent to leave the PAN. It is also used by the coordinator to instruct an associated device to leave the PAN

A.3.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mlme_disassociate_request(
                                    DeviceAddress,
                                    DisassociationReason,
                                    SecurityEnable
                                    );
```

A.3.2 Expected results of the call

MCPS DISASSOCIATE Confirm with a status of SUCCESS.

A.4 MLME GET Request

The MLME GET Request function call requests information about a given PIB attribute.

A.4.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mlme_get_request(
                        PIBAttribute
                        );
```

A.4.2 Expected results of the call

If the requested MAC PIB attribute is successfully retrieved, an MLME GET Confirm message is issued with a status of SUCCESS.

A.5 MLME GTS Request

Allows a device to send a request to the PAN coordinator to allocate a new GTS or to deallocate an existing GTS.

A.5.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mlme_gts_request(
                        GTSCharacteristics,
                        SecurityEnable
                        );
```

A.5.2 Expected results of the call

MCPS GTS Confirm message with a status of SUCCESS.

A.6 MLME POLL Request

Prompts the device to request data from the coordinator.

A.6.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mlme_poll_request(
                        CoordAddrMode,
                        CoordPANId,
                        CoordAddress,
                        SecurityEnable
                        );
```

A.6.2 Expected results of the call

MCPS POLL Confirm message with a status of SUCCESS.

A.7 MCPS PURGE Request

The handle of the MSDU is purged from the transaction queue.

A.7.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mcps_purge_request(
                        msduHandle
                        );
```

A.7.2 Expected results of the call

MCPS PURGE Confirm message with a status of SUCCESS.

A.8 MLME RESET Request

Used to set all PIB values to a default value.

A.8.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mlme_reset_request(
                        SetDefaultPib
                        );
```

A.8.2 Expected results of the call

MCPS RESET Confirm message with a status of SUCCESS.

A.9 MLME RX ENABLE Request

Allows the application to request that the receiver is enabled for a finite period of time.

A.9.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mlme_rx_enable_request(
                            DeferPermit
                            RxOnTime
                            RxOnDuration
                            );
```

A.9.2 Expected results of the call

MCPS RX-ENABLE Confirm message with a status of SUCCESS.

A.10 MLME SCAN Request

Used to initiate a channel scan over a given list of channels. A device can use a channel scan to measure the energy on the channel, search for the coordinator with which it associated, or search for all coordinators transmitting beacon frames within the POS of the scanning device.

A.10.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mlme_scan_request(
                       ScanType,
                       ScanChannels,
                       ScanDuration
                       );
```

A.10.2 Expected results of the call

MCPS SCAN Confirm message with a status of SUCCESS.

A.11 MLME SET Request

Attempts to write the given value to the indicated MAC PIB attribute.

A.11.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mlme_set_request(
                        PIBAttribute,
                        PIBAttributeValue,
                        PIBAttributeValueSize
                        );
```

A.11.2 Expected results of the call

MCPS SET Confirm message with a status of SUCCESS.

A.12 MLME START Request

Makes a request for the device to start using a new superframe configuration.

A.12.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mlme_start_request(
                         PANId,
                         LogicalChannel,
                         BeaconOrder,
                         SuperframeOrder,
                         PANCoordinator,
                         BatteryLifeExtension,
                         CoordRealignment,
                         SecurityEnable
                         );
```

A.12.2 Expected results of the call

MCPS START Confirm message with a status of SUCCESS.

A.13 MLME SYNC Request

Requests to synchronize with the coordinator by acquiring and, if specified, tracking its beacons.

A.13.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mlme_sync_request(
                        LogicalChannel,
                        TrackBeacon
                        );
```

A.13.2 Expected results of the call

Searches for the current network beacon. If the TrackBeacon parameter is equal to TRUE, the MLME will track the beacon, i.e., enable its receiver just before the expected time of each beacon so that the beacon frame can be processed. If the TrackBeacon parameter is equal to FALSE, the MLME will locate the beacon, but not continue to track it. This function call is not followed by a confirm call.

Appendix B MAC Confirm Callback Functions

The following callback functions are generated in the MAC layer and implemented by the application/network layer. If the callback is not implemented in the application, a null function from the library will be used. These confirm functions are listed in alphabetical order.

B.1 MLME ASSOCIATE Confirm

The MLME ASSOCIATE Confirm message is used to indicate the results of one or more associated MLME ASSOCIATE Requests.

B.1.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_associate_conf(
                        AssocShortAddress,
                        status
                        );
```

B.1.2 Trigger Event

If the service request was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

B.2 MCPS DATA Confirm

The MCPS DATA Confirm message is used to indicate the results of one or more associated previous service requests.

B.2.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mcps_data_conf(
                   msduHandle
                   status
                   );
```

B.2.2 Trigger Event

If the service request was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

B.3 MLME DISASSOCIATE Confirm

The MLME DISASSOCIATE Confirm message s used to indicate the results of one or more associated previous service requests.

B.3.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_disassociate_conf(
                           status
                           );
```

B.3.2 Trigger Event

If the service request was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

B.4 MLME GET Confirm

The MLME GET Confirm message s used to indicate the results of one or more associated previous service requests.

B.4.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_get_conf(
                    status,
                    PIBAttribute,
                    PIBAttributeValue,
                    PIBAttributeValueSize
                    );
```

B.4.2 Trigger Event

If the service request was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

B.5 MLME GTS Confirm

The MLME GTS Confirm message is used to indicate the results of one or more associated previous service requests.

B.5.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_gts_conf(
                    GTSCharacteristics,
                    status
                    );
```

B.5.2 Trigger Event

If the service request was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

B.6 MLME POLL Confirm

The MLME POLL Confirm message is used to indicate the results of one or more associated previous service requests..

B.6.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_poll_conf(
                     status
                     );
```

B.6.2 Trigger Event

If the service request was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

B.7 MCPS PURGE Confirm

The MCPS PURGE Confirm message is used to indicate the results of one or more associated previous service requests.

B.7.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mcps_purge_confirm(
                          msduHandle,
                          status
                          );
```

B.7.2 Trigger Event

If the service request was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

B.8 MLME RESET Confirm

The MLME RESET Confirm message is used to indicate the results of one or more associated previous service requests.

B.8.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_reset_conf(
                    status
                    );
```

B.8.2 Trigger Event

If the service request was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

B.9 MLME RX ENABLE Confirm

The MLME RX ENABLE Confirm message is used to indicate the results of one or more associated previous service requests.

B.9.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_rx_enable_conf(
                        status
                        );
```

B.9.2 Trigger Event

If the service request was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

B.10 MLME SCAN Confirm

The MLME SCAN Confirm message is used to indicate the results of one or more associated previous service requests..

B.10.1 Semantics of the service call

The Semantics of the service call are:

```
Usr_mlme_scan_conf(
                   status,
                   ScanType,
                   UnscannedChannels,
                   ResultListSize,
                   data,
                   data_length
                   );
```

B.10.2 Trigger Event

If the service request was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

B.11 MLME SET Confirm

The MLME SET Confirm message is used to indicate the results of one or more associated previous service requests.

B.11.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_set_conf(
                  status,
                  PIBAttribute
                  );
```

B.11.2 Trigger Event

If the service request was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

B.12 MLME START Confirm

The MLME START Confirm message is used to indicate the results of one or more associated previous service requests.

SeThe MLME-POLL.request primitive prompts the device to request data from the coordinator.

The MLME-POLL.request primitive prompts the device to request data from the coordinator.

B.12.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_start_conf(
                    status
                    );
```

B.12.2 Trigger Event

If the service request was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

Appendix C MAC Indication Callback Functions

The following callback functions are called from the MAC layer and implemented in the application/network layer. If the application/network layer does not implement the callback, a null function from the library will be used. These indication functions are listed in alphabetical order.

C.1 MLME ASSOCIATE Indication

The indication is passed from the MAC to indicate an event that is significant to the application. This event may be logically related to a remote service request, or it may be caused by an internal MAC event.

C.1.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_associate_ind(
                        DeviceAddress,
                        CapabilityInformation,
                        SecurityUse,
                        ACLEntry
                        );
```

C.1.2 Trigger Event

Arrival of data at the device.

C.2 MLME BEACON-NOTIFY Indication

The indication is passed from the MAC to indicate an event that is significant to the application. This event may be logically related to a remote service request, or it may be caused by an internal MAC event.

C.2.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_beacon_notify_ind(
                            BSN,
                            PANDescriptor,
                            PendAddrSpec,
                            data,
                            data_length
                            );
```

C.2.2 Trigger Event

Arrival of data at the device.

C.3 MLME COMM STATUS Indication

The indication is passed from the MAC to indicate an event that is significant to the application. This event may be logically related to a remote service request, or it may be caused by an internal MAC event.

C.3.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_comm_status_ind(
                              PANId,
                              SrcAddrMode,
                              SrcAddr,
                              DstAddrMode,
                              DstAddr,
                              status
                              );
```

C.3.2 Trigger Event

Arrival of data at the device.

C.4 MCPS DATA Indication

The indication is passed from the MAC to indicate an event that is significant to the application. This event may be logically related to a remote service request, or it may be caused by an internal MAC event.

C.4.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mcps_data_ind(
                        pAddrMode,
                        mpduLinkQuality,
                        SecurityUse,
                        ACLEntry,
                        msduLength,
                        msdu
                        );
```

C.4.2 Trigger Event

Arrival of data at the device.

C.5 MLME DISASSOCIATE Indication

The indication is passed from the MAC to indicate an event that is significant to the application. This event may be logically related to a remote service request, or it may be caused by an internal MAC event.

C.5.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_disassociate_ind(
                              DeviceAddress,
                              DisassociateReason,
                              SecurityUse,
                              ACLEntry
                              );
```

C.5.2 Trigger Event

Arrival of data at the device.

C.6 MLME GTS Indication

The indication is passed from the MAC to indicate an event that is significant to the application. This event may be logically related to a remote service request, or it may be caused by an internal MAC event.

C.6.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_gts_ind(
                    DevAddress,
                    GTSCharacteristics,
                    SecurityUse,
                    ACLEntry
                    );
```

C.6.2 Trigger Event

Arrival of data at the device.

C.7 MLME ORPHAN Indication

The indication is passed from the MAC to indicate an event that is significant to the application. This event may be logically related to a remote service request, or it may be caused by an internal MAC event.

C.7.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_orphan_ind(
                        OrphanAddress,
                        SecurityUse,
                        ACLEntry
                        );
```

C.7.2 Trigger Event

Arrival of data at the device.

C.8 MLME SYNC-LOSS Indication

The indication is passed from the MAC to indicate an event that is significant to the application. This event may be logically related to a remote service request, or it may be caused by an internal MAC event.

C.8.1 Semantics of the service call

The Semantics of the service call are:

```
usr_mlme_sync_loss_ind(
                            LossReason
                            );
```

C.8.2 Trigger Event

Arrival of data at the device.

Appendix D MAC Response Functions

The following functions are implemented in the MAC. The response functions are called from the application in order to respond to an indication function. These response functions are listed in alphabetical order.

D.1 MLME ASSOCIATE Response

Used to initiate a response to an MLME ASSOCIATE Indication message.

D.1.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mlme_associate_response(
                                DeviceAddress,
                                assoc_ShortAddress,
                                Status,
                                SecurityEnable
                                );
```

D.1.2 Expected results of the call

If requested, the MAC will issue the MLME COMM-STATUS Indication with a status of SUCCESS.

D.2 MLME ORPHAN Response

Allows the next higher layer of a coordinator to respond to the MLME ORPHAN Indication message

D.2.1 Semantics of the service call

The Semantics of the service call are:

```
wpan_mlme_orphan_response (
                               OrphanAdress,
                               ShortAddress,
                               AssociatedMember,
                               SecurityEnable
                               );
```

D.2.2 Expected results of the call

If the MPDU was successfully transmitted and an acknowledgment was received, if requested, the MAC sub layer will issue the MLME COMM-STATUS Indication message with a status of SUCCESS.

Appendix E Parameter Library

Parameters used in the MCPS and MCSE function calls are listed alphabetically in the following table.

Table E--1. Parameter Library

Parameter	Type	Valid Range	
ACLEntry	Integer	0 x 00–0 x 08	
addrInfo	wpan_commstatus_addr_t; wpan_mcpsdata_addr_t	Structure	Pointer to either structure that holds the frame address information.
assocShortAddress	uint16_t	0 x 0000–0 x ffff	The short device address allocated by the coordinator. This parameter is set to 0xffff if the association was unsuccessful
AssociatedMember	bool	TRUE or FALSE	Boolean true if the orphaned device is associated with this coordinator
BatteryLifeExtension	bool	TRUE or FALSE	Boolean true will disable the receiver of the beaconing device for a period of time after the interframe spacing period of the beacon frame
BeaconOrder	uint8_t	0–15	How often the beacon is to be transmitted
BSN	Integer	0 x 00–0 x ff	The beacon sequence number.
CapabilityInformation	Bitmap	Refer to IEEE 802.15.4 paragraph 7.3.1.1.2.	The operational capabilities of the device requesting association.
CoordAddrMode	uint8_t	0 x 02–0 x 03	The addressing mode of the coordinator. This parameter can take one of the following values: 2 = 16 bit short address, 3 = 64 bit extended address.
CoordAddress	uint64_t	As specified by CoordAddrMode parameter.	Address of the coordinator
CoordPANId	uint16_t	0 x 0000–0 x ffff	PAN ID of the coordinator
Coord_Realignment	bool	TRUE or FALSE	Boolean to transmit a coordinator realignment command prior to changing the superframe configuration
data	Array of bytes		
data_length	size_t	aMaxMACFrameSize	

Table E--1. Parameter Library

Parameter	Type	Valid Range	
DeferPermit	bool	TRUE or FALSE	Boolean true if the receiver enable can be deferred until during the next superframe if the requested time ha already passed
DevAddress	Device address	0 x 0000–0 x fffd	The short address of the device that has been allocated or deallocated a GTS.
DeviceAddress	uint64_t	An extended 64 bit IEEE address	The address of the requesting device
DisassociateReason	Integer	0 x 00–0 x ff	The reason for the disassociation (see 7.3.1.3.2).
DstAddr	Device address	As specified by the DstAddrMode parameter	The individual device address of the device for which the frame was intended.
DstAddrMode	Integer	0 x 00–0 x 03	
GTSCharacteristics	GTS characteristics	See 7.3.3.1.2	The characteristics of the GTS.
LogicalChannel	uint8_t	Selected from the available logical channels supported by the PHY	Channel to transmit on
LossReason	Enumeration	PAN_ID_CONFLICT, REALIGNMENT, or BEACON_LOST	The reason that synchronization was lost.
mpduLinkQuality	Integer	0 x 00–0 x ff	
msdu	Set of octets	—	
msduHandle	uint16_t	0 x 00–0 x ff	
msduLength	Integer	aMaxMACFrame- Size	
OrphanAddress	uint64_t	Extended 64 bit IEEE address	The address of the orphaned device
pAddrInfo			Pointer to address info structure
PANCoordinator	bool	TRUE or FALSE	Boolean to become the PAN coordinator or not
PANDescriptor	PANDescriptor value	See Table 41	The PANDescriptor for the received beacon.
PANId	Integer	0 x 0000–0 ffff	The 16 bit PAN identifier to be used by the beacon
PendAddrSpec	Bitmap	See 7.2.2.1.6	The beacon pending address specification.
PIBAttribute	Integer	See Table 71 and Table 72	The identifier of the MAC PIB attribute.
PIBAttributeValue	Various	Attribute specific; see Table 71 and Table 72	The value of the indicated MAC PIB attribute to be stored or that was read.
PIBAttributeValueSize	Uint8_t	Variable	The size of PIBAttributeValue
ResultListSize	Integer	Implementation specific	The number of elements returned in the appropriate result lists. This value is 0 for the result of an orphan scan.

Table E--1. Parameter Library

Parameter	Type	Valid Range	
RxOnDuration	uint32_t	0 x 000000–0 x ffffff	The number of symbols that the receiver is to be enabled
RxOnTime	uint32_t	0 x 000000–0 x ffffff	The number of symbols from the start of the superframe before the receiver is to be enabled
ScanChannels	uint32_t	32 bit field	The channels to scan
ScanDuration	uint8_t	0–14	The duration of each scan
ScanType	Integer	0 x 00–0 x 03	Indicates if the type of scan performed: 0 x 00 = ED scan (FFD only). 0 x 01 = active scan (FFD only). 0 x 02 = passive scan. 0 x 03 = orphan scan.
SecurityEnable	bool	TRUE or FALSE	Boolean to enable or disable security
SetDefaultPib	bool	TRUE or FALSE	true = set all PIB values to a default value
ShortAddress	uint16_t	0 x 0000–0 x ffff	The short address allocated to the orphaned device if it is associated with this coordinator
SrcAddr	uint64_t	As specified by the SrcAddrMode parameter	
SrcAddrMode	uint8_t	0 x 00–0 x 03	
status	Enumeration	The value of the status field. SUCCESS, TRANSACTION_OVERFLOW, TRANSACCTION_EXPIRED, CHANNEL_ACCESS_FAILURE, INVALID_GTS, NO_ACK, NO_DATA, UNAVAILABLE_KEY, FRAME_TOO_LONG. FAILED_SECURITY_CHECK, or INVALID_PARAMETER.	The status of the attempt.
SuperframeOrder	uint8_t	0–BO or 15	The length of the active portion of the superframe, including the beacon frame
TrackBeacon	bool	TRUE or FALSE	Boolean to synchronize with the next beacon and attempt to track all future beacons
tx_options	uint8_t	0000 xxxx (where x can be 0 or 1)	
UnscannedChannels	Bitmap	32 bit field	Indicates which channels given in the request were not scanned (1 = not scanned, 0 = scanned or not requested). This parameter is only valid for passive or active scans.

Appendix F Release Notes

F.1 General Release Notes

Release 1.0.0

2006-07-05

Supported Compilers:

– GCC 3.4.2

– IAR® 4.10.1.5

Supported Hardware Platforms:

– Radio Controller Board 230

F.2 Atmel 802.15.4 MAC Software Release Notes

GTS

Implemented and passing ZQG level 1 tests, currently undergoing stress testing. Not released.

Enhanced Modes

(Auto-ACK & Auto-Retry) not released

Device cannot act as Router

Device does not transmit Beacon frames if it gets a Start request primitive after Association or receives a Beacon Request frame

When Coordinator switches from Non-Beacon to Beacon mode it does not send out a Coordinator Realignment frame

No Power Save implemented

– Transceiver is always on

– BatteryLifetExtension parameter not evaluated at Start request

– macRxOnWhenIdle PIB attribute always considered as TRUE

ACK frames in beaconing mode during CAP are not transmitted exactly at next slot boundary

Beaconing network with SuperframeOrder=15 (no active portion) not implemented

Rx Enable not tested

Tracking of Beacon frames before Association not implemented

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

5182A–ZIGB–09/06