WiMOD HCI

Host Controller Interface Specification (V0.9, 2010-10-26)

IMST GmbH

Carl-Friedrich-Gauss-Str. 2

D-47475 Kamp-Lintfort

Document History

Date	Version	Chapter	Comment
2009-06-10	0.1	All	First version (KvW)
2009-09-20	0.2	All	Document format and title changed (KvW)
2010-06-02	0.4	All	Rework
2010-07-08	0.5	All	Rework
2010-08-11	0.6	Applicable Documents Device Configuration RF Data Exchange Services	New Description of Device Parameter added Description updated
2010-08-12	0.7	RF Data Exchange Services	Reliable Service renamed into Acknowledged Service
2010-09-01	0.8	Chapter 2 Chapter 6	Chapter 2.3 moved to Chapter 5 Chapter 6 introduced
2010-10-26	0.9	Chapter 3 Chapter 6	Opcodes corrected for Device Configuration Services SPI Flowcharts added

Table of Contents

Abbreviations

ADC	Analog-to-Digital Converter
DIO	Digital Input/Output
DLL	Dynamic Link Library
FW	Firmware
GPIO	General Purpose Input/Output
HCI	Host Controller Interface
HW	Hardware
RF	Radio Frequency
SPI	Serial Peripheral Interface
SW	Software
UART	Universal Asynchronous Receiver/Transmitter
WiMOD	Wireless Module

1 Introduction

1.1 Purpose

This document specifies the WiMOD Host Controller Interface (HCI) Protocol.

1.2 Overview

The WiMOD HCI Protocol is designed to expose the WiMOD Radio Services to an external Host Controller. The communication between Host and WiMOD is based on so called HCI Messages which can be sent through an UART (see Fig.1). The interface and communication protocol is common to all WiMODs and provides many services for configuration and RF data exchange.

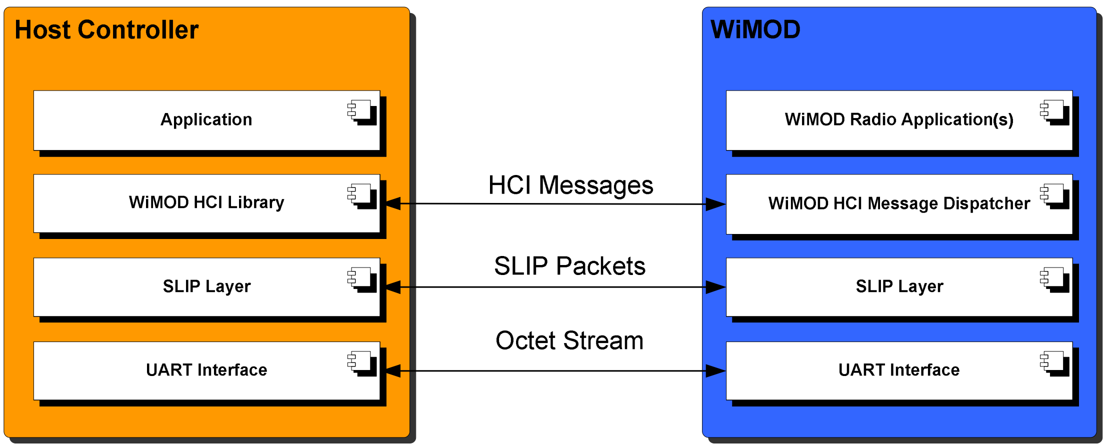


Fig. 1-1: Host Controller Communication

Document Guide

Chapter 2 explains the message flow between Host Controller and WiMOD Controller. Following a description of the general message format is given.

Chapter 3 give a detailed summary of services and HCI messages which are available for all WiMODs.

Chapter 4 lists several tables with constants and parameter.

Chapter 5 provides a C-Code example which may help to implement the Host Controller part of this protocol.

1.3 Applicable Documents

[1]	WiMODHCI_Spec.pdf	Specification of the WiMOD Host Controller Interface.
[2]	iMxxx_Data_Sheet.pdf	Data sheet for the corresponding device xxx
[3]	iMxxx_Settings.pdf	Parameter Settings and Examples for the corresponding device xxx
[4]	WiMODDLL_Spec.pdf	Specification of the WiMOD HCI DLL

2 WiMOD HCI Communication

2.1 Message Flow

The WiMOD HCI Protocol defines three different types of messages which are exchanged between Host Controller and WiMOD Controller:

1. Command Message: always sent from Host Controller to WiMOD Controller to trigger a function.

2. Response Message: sent from WiMOD Controller to Host Controller to answer a preceding HCI Command Message

3. Event Message: can be sent from WiMOD Controller to Host Controller at any time

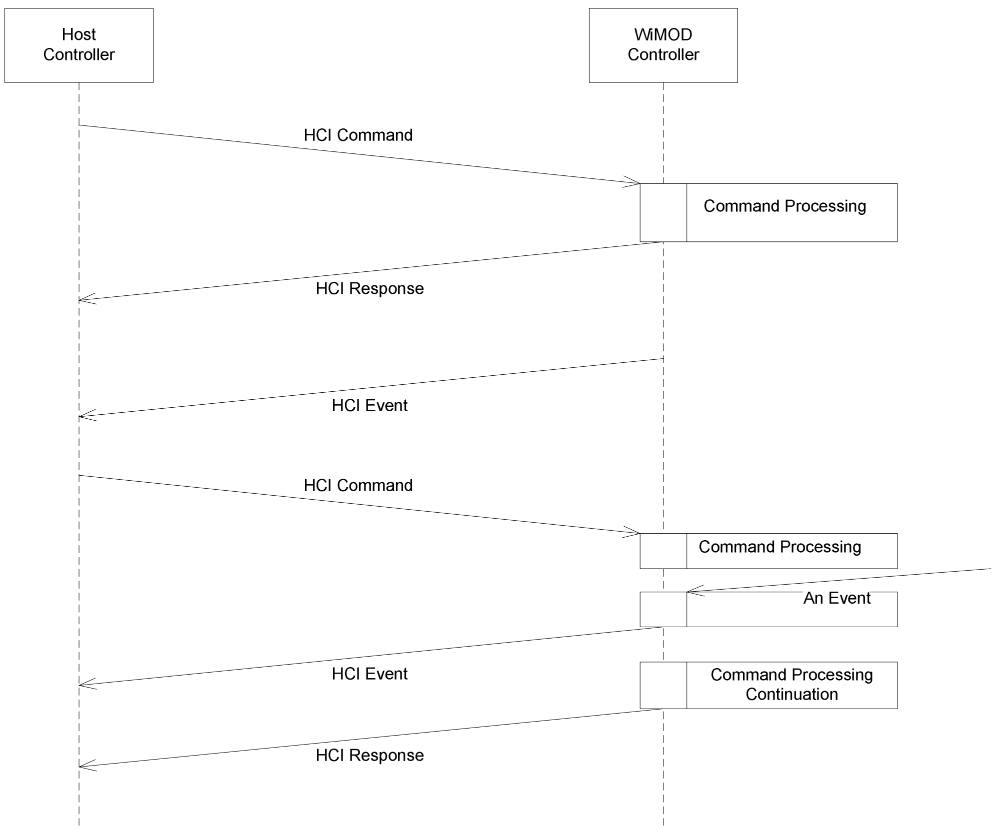


Fig. 2-1: HCI Message Flow

2.2 WiMOD HCI Message Format

The communication between a WiMOD device and a host device is realized by means of the following message format.

Type Field	Control Field	Destination ID	Source ID	Opcode	Payload Length	Payload (0...max octetts)

Fig. 2-2: WiMOD HCI Message

This message format is used to call services and exchange information. The message transmission starts with the Type Field and ends with the last byte of the Payload Field. Note: the Payload Field might be empty.

The WiMOD HCI Message requires a reliable transport layer and does not provide any kind of error checks or frame synchronisation means. More details about message transfer over UART and SPI are given in Chapter 5 and Chapter 6.

Note: The UART Interface as described in Chapter 5 is the standard Host Controller Interface of the WiMODs. The SPI Interface is an optional interface which can be ordered as a special Firmware for some WiMODs.

2.3 Elements of the WiMOD HCI Message

This chapter describes the message format in detail.

2.3.1 Type Field (8 bits)

The Type Field is used to distinguish between the different message types. This field is subdivided into a Type Information Field and an Address Field.

7 2 0

Address Field	Type Information

Fig. 2-3: Type Field

2.3.1.1 Type Information Field (3 bits)

Value	Type	Description
000b	Command Frame	Send from Host to WiMOD device to call a function.
001b	Response Frame	Send from WiMOD device to Host in consequence of a command frame.
010b	Event Frame	Send from WiMOD device to Host without a preceding command frame to indicate a system state or to pass an information element.
011b	Reserved	This Type is reserved.
1xxb	Reserved	These combinations are reserved for special purposes and must not be set for application services.

2.3.1.2 Address Field (5 bits)

The Address Field is reserved for future extensions and must be set to zero 00000b.

2.3.2 Control Field (8 bit)

The function of this field depends on the message type:

Command Message Format

This field is set to zero 0x00 for all command frames.

Event Message Format

This field is set to zero 0x00 for all event frames.

Response Message Format

Within a Response Message the Control Field contains status information about the preceding command within the two least significant bits. The status field has to be checked prior to any other following field.

7	1 0
unused	Status Information

Fig. 2-4: Control Field

2.3.2.1 Status Information Field (2 bits)

Value	Description
00b	Command failed
01b	Command successful
10b	Command not supported
11b	Reserved

2.3.3 Destination ID (8 bit)

This field identifies a logical destination endpoint within a device. The value must be unique for all nodes within one application.

2.3.4 Source ID (8 bit)

This field identifies the logical source endpoint within a device. The value must be unique for all nodes within one application.

2.3.5 Opcode Field (8 bit)

This field contains the opcode number which triggers a dedicated service function of a given endpoint. Opcode values must only be unique within one single endpoint. The value 0x00 is reserved.

2.3.6 Length Field (8 bit)

The Length Field contains the size of the following Payload Field. The size is given as number of octets. Note: The Payload Length could be zero.

2.3.7 Payload (0 ... max. octets)

The Payload Field contains service / function dependent data. The size depends on the given Firmware.

Note: The “AppStarterKit” firmware provides a maximum payload field of 32 octets.

3 Common Services

This chapter outlines the message format for the common WiMOD services. The functions are ordered according to their corresponding endpoint. The global endpoint identifiers are listed in chapter "Global System Identifier".

3.1 Device Management Services

The Device Management services are accessible through endpoint identifier DEVMGMT_ID (see chapter "Global System Identifier"). The following services are available:

- Ping
- Reset
- Device Information
- Device Configuration
- System Operation Modes
- RF Ping
- Peer Device Information
- Peer Device Configuration

3.1.1 Ping

This command is used to check if the connected device is alive. The sender should expect a Ping Response within a certain time interval.

Message Flow

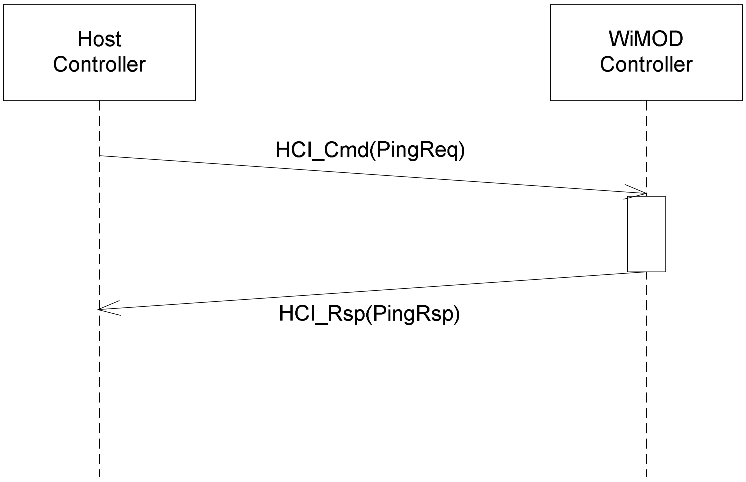


Fig. 3-1: Ping

Command Message

Field	Content	Comment
DstID	DEVMGMT_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x01	Ping Request
Length	0	No payload

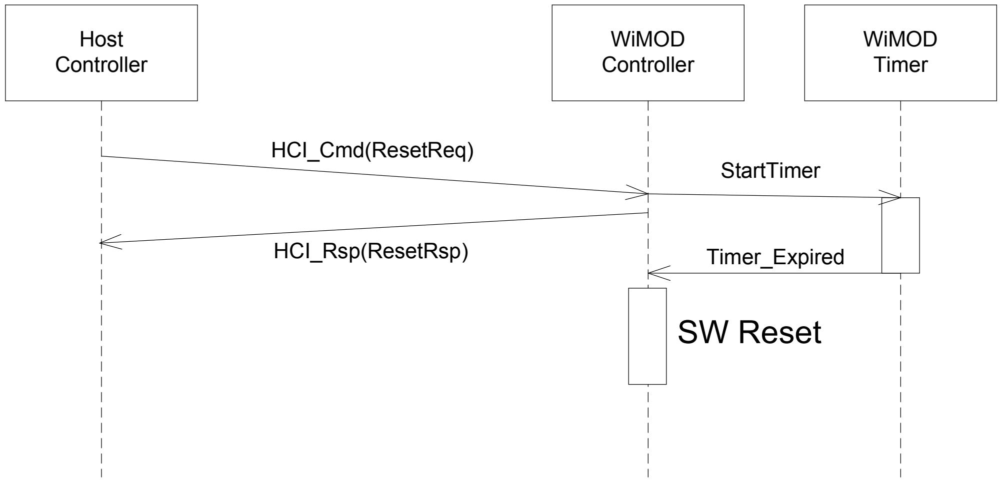
Response Message

Field	Content	Comment
DstID	Sender ID	Destination EP identifier, taken from preceding Ping Request command
SrcID	DEVMGMT_ID	Source EP identifier
Opcode	0x02	Ping Response
Length	0	No payload

3.1.2 Reset

This message can be used to reset a WiMOD. The reset will be performed after approx. 500ms.

Message Flow



3-1: Reset Request

Command Message

Field	Content	Comment
DstID	DEVMGMT_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x29	Reset Request
Length	0	No Payload

Response Message

This message acknowledges the Reset Request message.

Field	Content	Comment
DstID	Sender ID	Destination EP identifier, taken from preceding Request command.
SrcID	DEVMGMT_ID	Source EP identifier
Opcode	0x2A	Reset Response
Length	0	No Payload

3.1.3 Device Information

The WiMOD Firmware provides a service to readout some information elements for identification purposes.

3.1.3.1 Get Device Information

This service can be used to identify the local connected device. As a result the device sends a response message which contains a Device Information Field.

Command Message

Field	Content	Comment
DstID	DEVMGMT_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x05	Get Device Information Request
Length	0	No payload

Response Message

This message contains the Device Information Field which is described below.

Field	Content	Comment
DstID	Sender ID	Destination EP identifier, taken from the preceding Request command.
SrcID	DEVMGMT_ID	Source EP identifier
Opcode	0x06	Get Device Information Response
Length	sizeof(DEVIDE_INFO_FIELD)	Size of Device Information Field
Payload […]	DEVICE_INFO_FIELD	Device Information Field

3.1.3.2 Device Information Field

The Device Information Field contains the following elements:

Offset	Content
 0x00 0x01	Device Address (16 Bit) - used to address a certain device via RF link Device Address (Bits 0…7) Device Address (Bits 8…15)
0x02	Module Type (8 Bit) - identifies the type of module (see Appendix B)
0x03	Device Mode (8 Bit) - identifies the configured device mode: The firmware provides different RF operation modes: 0 = End Device (Default Configuration) 1 = Reserved 2 = Repeater: device which simply retransmits every received message 3 = Sniffer: sends every received message to its connected host
0x04	Firmware Version Number (8 Bit) - identifies the programmed firmware: Bits 7..4 -> major FW Version number Bits 3..0 -> minor FW Version number Example: 0x14 -> FW Version = 1.4
0x05	HCIProtocolVersion (8 Bit) This value identifies the version of the implemented HCI protocol. It will be incremented due to an update on Firmware side so that the Host is able to detect an incompatibility.

Example

A Device Information Field which includes the Device Address (0x1234) of an iM820A End Device with Firmware Version 1.3 and HCI Protocol Version 1 looks as follows:

1. Byte	2.Byte	3.Byte	4.Byte	5.Byte	6. Byte
LOBYTE(DevAddr)	HIBYTE(DevAddr)	Module Type	Device Mode	FW Version	HCI Version
0x34	0x12	0x01	0x00	0x13	0x01

The payload size is 6 bytes in this case.

3.1.4 Device Configuration

The Firmware provides several configurable system parameters which are stored in a none volatile memory. These configuration parameters are readout during start-up to configure the firmware components and hardware units.

3.1.4.1 Device Parameter

The following parameters are available on alll WiMODs but their range and physical meaning is WiMOD device specific. The exact meaning can be found in [3]:

Parameter	Description
RF_DataRate	Index for RF Datarate
RF_PowerLevel	Index for RF Powerlevel
RF_Channel	Index for RF Channel

The next table outlines parameters which have the same meaning on all WiMOD devices:

Parameter	Description
NetworkAddress (8 Bit)	A Network Address is used to separate groups of WiMODs from each other. A device accepts RF messages which contain its own Network Address or the BROADCAST_NETWORK_ADDRESS (0xFF). Valid range: 0x01 – 0xFE The values 0x00 and 0xFF are reserved. Note: Sniffer devices perform no Network Address filtering and must set the NetworkAddress to 0xFF
DeviceAddress (16 Bit)	The Device Address is used to address a certain device within a group of devices with same RF settings. Therefore the device address must be set to a unique value to ensure proper operation. A device accepts RF messages which contain its own configured Device Address or the BROADCAST_DEVICE_ADDRESS (0xFFFF). Valid range: 0x0001 – 0xFFFE The values 0 and 0xFFFF are reserved. Note: Repeater devices and Sniffer devices perform no Device Address filtering
DeviceMode (8 Bit)	The firmware provides different RF operation modes: 0 = End Device (Default Configuration): a standard device with no specific function 1 = Reserved 2 = Repeater: device which simply retransmits every received RF message 3 = Sniffer: sends every received message to its connected host
AckNumRetries (8 Bit)	Defines the maximum number of retransmissions for RF messages which are send by means of the Acknowledged Data Exchange Service
AckTimeoutTicks (8 Bit)	Defines the timeout window (in 10ms ticks) the sender is waiting for an ACK message before a retransmission is initiated

3.1.4.2 Get Device Parameter

This command can be used to read several device parameters.

Command Message

Field	Content	Comment
DstID	DEVMGMT_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x09	Get Device Parameter Request
Length	0	No Payload

Response Message

This message contains the Device Parameter Field which is described below.

Field	Content	Comment
DstID	Sender ID	Destination EP identifier, taken from preceding Request command.
SrcID	DEVMGMT_ID	Source EP identifier
Opcode	0x0A	Get Device Parameter Response
Length	Variable	size of Device Parameter Field
Payload […]	DEVICE_PARAM_FIELD	Device Parameter Field

3.1.4.3 Set Device Parameter

This command can be used to change several device parameters. The new settings get valid after a system reset (see Reset Request) of the connected WiMOD. The Parameter Field is described in detail below.

Command Message

Field	Content	Comment
DstID	DEVMGMT_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x0B	Set Device Parameter Request
Length	variable	Size of Device Parameter Field
Payload […]	DEVICE_PARAM_FIELD	Device Parameter Field

Response Message

This message acknowledges the Set Device Parameter Request.

Field	Content	Comment
DstID	Sender ID	Destination EP identifier, taken from preceding Request command.
SrcID	DEVMGMT_ID	Source EP identifier
Opcode	0x0C	Set Device Parameter Response
Length	0	No Payload

3.1.4.4 Device Parameter Field

The Device Parameter Field is a variable Information Element which is able to carry a complete list of Device Parameter. A preceding Information Indicator Flag indicates the exact content of the following list. If an Information Indicator Bit is set the corresponding Parameter is attached. The Indicator Flag must be parsed starting with Bit 0.

Offset	Content
0x00	Information Indicator Flag: Bit field: 0000 0001b -> Network Address 0000 0010b -> Device Address 0000 0100b -> RF Data Rate 0000 1000b -> RF Power Level 0001 0000b -> RF Channel 0010 0000b -> Device Mode 0100 0000b -> Number of retries for acknowledged data exchange service 1000 0000b -> Timeout window for expected ACK
0x01	NetworkAddress (8 Bit)
Variable	DeviceAddress (16 Bit)
variable	RF_DataRate (8 Bit)
variable	RF_PowerLevel (8 Bit)
variable	RF_Channel (8 Bit)
variable	DeviceMode (8 Bit)
variable	AckNumRetries (8 Bit)
variable	AckTimeoutTicks (8 Bit)

Note: The specific values for RF Data Rate, RF Power Level and RF Channel and the physical meaning are presented in the WiMOD specific documents [3].

Example

A Device Parameter Element which includes the Device Address (0x1234) and RF Channel (0x02) looks as follows:

1. Byte	2.Byte	3.Byte	4. Byte
IIFlag	LOBYTE(DevAddr)	HIBYTE(DevAddr)	RF Channel
0x12	0x34	0x12	0x02

The payload size is 4 bytes in this case.

3.1.4.5 Factory Reset

This message can be used to reset the device parameters. All parameters except the Device Address will be set to default factory settings. A resetFlag parameter defines if a system reset should be initiated or not. The reset will be executed after approx. 500ms

Command Message

Field	Content	Comment
DstID	DEVMGMT_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x27	Factory Reset Request
Length	1	Size of payload
Payload[0]	resetFlag	0: no reset 1: perform system reset

Response Message

This message acknowledges the Factory Reset Request message.

Field	Content	Comment
DstID	Sender ID	Destination EP identifier, taken from preceding Request command.
SrcID	DEVMGMT_ID	Source EP identifier
Opcode	0x28	Factory Reset Response
Length	0	No Payload

3.1.5 System Operation Modes

The firmware supports different kind of System Operation Modes to align its behaviour according to a certain use case e.g. Test Mode, Application Mode. The System Operation Mode is determined during start-up and requires a reset to get changed.

3.1.5.1 Set Next Operation Mode

This message can be used to set the next System Operation Mode. A password is required to perform the change. A resetFlag can be set to initiate an automatic system reset. The system reset will be executed after approx. 500ms.

Command Message

Field	Content	Comment
DstID	DEVMGMT_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x1B	Set System Operation Mode Request
Length	10	Size of payload
Payload[0..7]	Password	
Payload[8]	Next Operation Mode	8 Bit value, see below for details
Payload[9]	resetFlag	0: no reset 1: perform system reset

Response Message

This message acknowledges the Set System Operation Mode message.

Field	Content	Comment
DstID	Sender ID	Destination EP identifier, taken from preceding Request command.
SrcID	DEVMGMT_ID	Source EP identifier
Opcode	0x1C	Set System Operation Mode Response
Length	0	No Payload

3.1.5.2 Get System Operation Mode

This service can be used to readout the current System Operation Mode.

Command Message

Field	Content	Comment
DstID	DEVMGMT_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x1D	Get System Operation Mode Request
Length	0	No payload

Response Message

This message contains the current System Operation Mode.

Field	Content	Comment
DstID	Sender ID	Destination EP identifier, taken from preceding Request command.
SrcID	DEVMGMT_ID	Source EP identifier
Opcode	0x1E	Get System Operation Mode Response
Length	1	Size of Payload
Payload[0]	System Operation Mode	8 Bit value, see below for details

3.1.5.3 System Operation Modes

The following System Operation Modes are supported:

Value	Description
0	Standard Application Mode / Default Mode
1	Hardware Test Mode
2	Production Mode
3	Self test Mode

3.1.6 RF Ping

This service is used to check via RF link if a peer device is alive. The Payload Field of this message contains a 16-Bit destination/peer device address which is used to address a certain WiMOD in range. The device address is a unique value from 0x0001 to 0xFFFE. The values 0x0000 and 0xFFFF are reserved for special purposes. The later one is used as BROADCAST_DEVICE_ADDRESS for broadcast services via RF link. The local WiMOD Controller converts the incoming HCI command into a RF message which is sent over the air. If a WiMOD device with matching Device Address is in range, it answers with a corresponding RF Ping Response message which will finally be forwarded to the Host controller. If no WiMOD device answers, no response message will be generated.

Message Flow

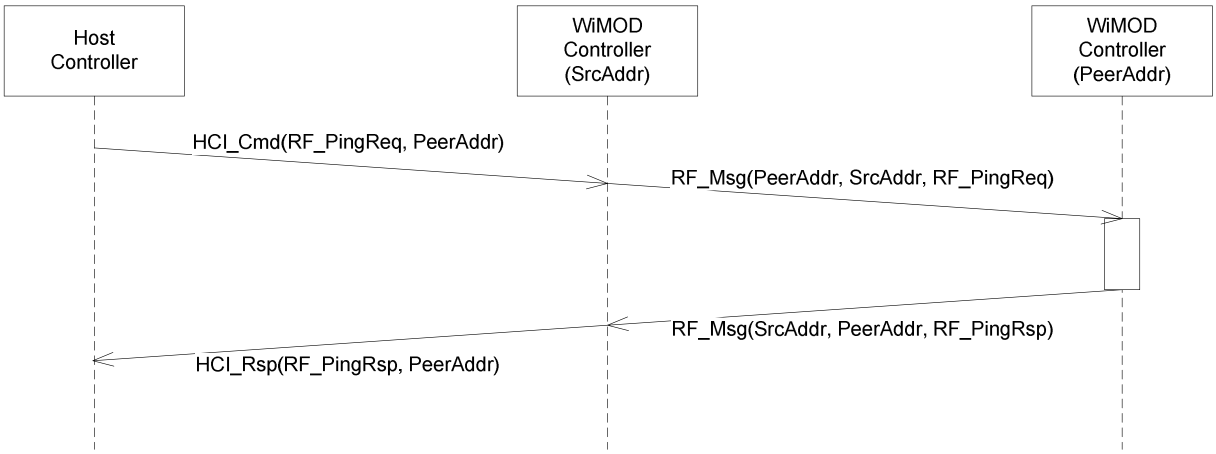


Fig. 3-2: RF Ping

Command Message

Field	Content	Comment
DstID	DEVMGMT_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x03	RF Ping Request
Length	2	Payload Length
Payload[0] Payload[1]	Device Address [Bits 0…7] Device Address [Bits 8…15]	Destination/Peer Device Address

Response Message

Field	Content	Comment
DstID	Sender ID	Destination EP identifier, taken from preceding Ping Request command
SrcID	DEVMGMT_ID	Source EP identifier
Opcode	0x04	RF Ping Response
Length	2	Payload Length
Payload[0] Payload[1]	Device Address [Bits 0…7] Device Address [Bits 8…15]	Source Device Address

3.1.7 Peer Device Information

The service can be used to get information from a peer device via RF link.

Command Message

Field	Content	Comment
DstID	DEVMGMT_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x07	Get Peer Device Information Request
Length	2	
Payload[0] Payload[1]	Device Address [Bits 0…7] Device Address [Bits 8…15]	Destination Device Address

Response Message

This message contains the Device Information Field from the peer device.

Field	Content	Comment
DstID	Sender ID	Destination EP identifier, taken from the preceding Request command.
SrcID	DEVMGMT_ID	Source EP identifier
Opcode	0x08	Get Device Information Request
Length	sizeof(DEVIDE_INFO_FIELD)	Size of Device Information Field
Payload […]	DEVICE_INFO_FIELD	Device Information Field

Note: The Device Information Field is the same as described in chapter Device Information.

3.1.8 Peer Device Configuration

This service allows configuring a peer device via RF link. The list of configurable parameters is described in chapter Device Configuration.

3.1.8.1 Get Peer Device Parameter

This command can be used to read several device parameters from a peer device.

Command Message

Field	Content	Comment
DstID	DEVMGMT_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x0D	Get Peer Device Parameter Request
Length	2	Payload Length
Payload[0] Payload[1]	Device Address [Bits 0…7] Device Address [Bits 8…15]	Destination Device Address

Response Message

This message contains the Device Parameter Field of the peer device.

Field	Content	Comment
DstID	Sender ID	Destination EP identifier, taken from preceding Request command.
SrcID	DEVMGMT_ID	Source EP identifier
Opcode	0x0E	Get Peer Device Parameter Response
Length	variable	size of Device Parameter Field
Payload […]	DEVICE_PARAM_FIELD	Device Parameter Field

Note: The Device Parameter Field is the same as described in chapter Device Configuration.

3.1.8.2 Set Peer Device Parameter

This command can be used to change several parameters on the peer device. The new settings get valid after a system reset.

Command Message

Field	Content	Comment
DstID	DEVMGMT_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x0F	Set Peer Device Parameter Request
Length	Variable	
Payload[0] Payload[1]	Device Address [Bits 0…7] Device Address [Bits 8…15]	Destination Device Address
Payload […]	DEVICE_PARAM_FIELD	Device Parameter Field

Response Message

This message acknowledges the Set Peer Device Parameter Request.

Field	Content	Comment
DstID	Sender ID	Destination EP identifier, taken from preceding Request command.
SrcID	DEVMGMT_ID	Source EP identifier
Opcode	0x10	Set Peer Device Parameter Response
Length	1	
Payload[0]	Status	1: operation successful 0: operation failed

3.2 RF Data Exchange Services

The RF Data Exchange services can be used to send data packets from one Host Controller to another one. The services are accessible through endpoint identifier DATALINK_ID (see chapter “Global System Identifier”). The following services are available:

- Unreliable Data Request
- Acknowledged Data Request

3.2.1 Unreliable Data Request

This service can be used to send data packets via RF without acknowledgement from the peer device. The local WiMOD Controller converts the incoming HCI message into a RF message which is sent over the air. If a peer WiMOD Controller is in range it accepts this RF message if the following conditions are valid:

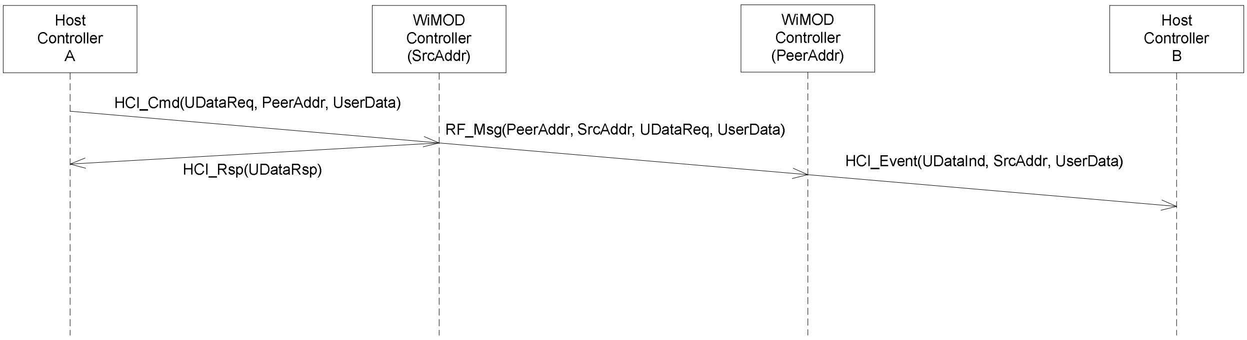
1. the sending and receiving WiMODs have the same physical RF parameter settings: RF_DataRate, RF_Channel

2. the WiMODs have the same NetworkAddress configuration

3. the RF message contains either the Destination/Peer DeviceAddress of the receiving WiMOD or the BROADCAST_DEVICE_ADDRESS

The peer Host Controller will receive a HCI event message from the peer WiMOD if the RF transmission was successful. This HCI event message contains the transmitted User Data and the device address of the transmitting WiMOD which is also part of the received RF message.

Message Flow



3-2: Unreliable Data Request

Command Message

Field	Content	Comment
DstID	DATALINK_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x01	Unreliable Data Request
Length	n, n > 2	Variable length
Payload[0] Payload[1]	Device Address [Bits 0…7] Device Address [Bits 8…15]	Destination/Peer Device Address
Payload[2…n-1]	User Data	User Data with n–2 octets

Note: it is possible to send “unreliable” messages as broadcast messages to all WiMODS in range by choosing a destination/peer device address of 0xFFFF which is the reserved BROADCAST_DEVICE_ADDRESS.

Response Message

This is the response message of the local WiMOD device.

Field	Content	Comment
DstID	Sender ID	Destination EP identifier, taken from preceding Request command.
SrcID	DATALINK_ID	Source EP identifier
Opcode	0x02	Unreliable Data Response
Length	0	No payload

Note: this message doesn’t confirm that the message is sent via RF.

Event Message

This message is send to a Host Controller as a result of a received “Unreliable Data Request” message via RF link. The first two bytes in the payload field contain the source device address of the transmitting WiMOD.

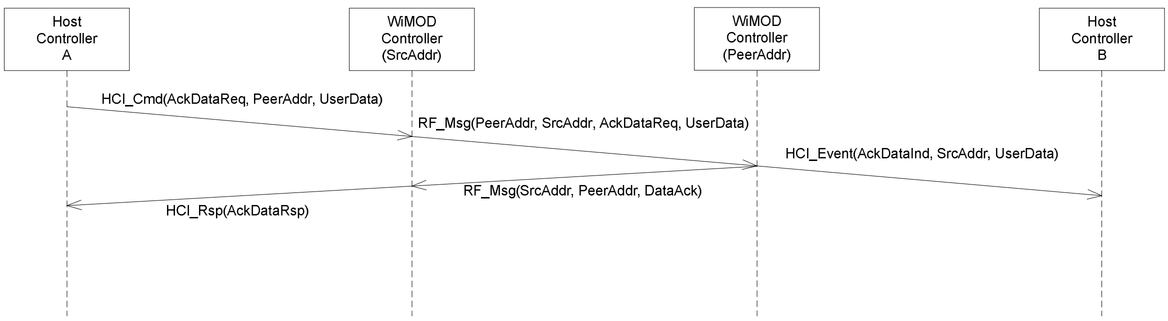
Field	Content	Comment
DstID	DATALINK_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x03	Unreliable Data Indication
Length	n, n > 2	Variable Length
Payload[0] Payload[1]	Device Address [Bits 0…7] Device Address [Bits 8…15]	Source Device Address
Payload[2…n-1]	User Data	User Data with n-2 octets

3.2.2 Acknowledged Data Request

This service can be used to send data packets via RF with acknowledgement from the peer WiMOD. If an RF Ack message is not received within a configurable timeout (see Device Configuration) a retransmission of the data packet will be initiated. The maximum number of retransmissions is configurable. On successful transmission the peer Host Controller will receive a HCI event message containing the User Data and source address of the transmitting WiMOD. Furthermore a Response Message is sent back to the transmitting Host Controller which contains the status of the packet transmission.

Note: RF retransmissions will be initiated even if only the RF Ack message isn't received i.e. a peer Host Controller must be prepared to receive an "Acknowledged Data Indication" message multiple times.

Message Flow



0-1: Acknowledged Data Request

Command Message

Field	Content	Comment
DstID	DATALINK_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x04	Acknowledged Data Request
Length	n, n > 2	Variable Length
Payload[0] Payload[1]	Device Address [Bits 0…7] Device Address [Bits 8…15]	Destination/Peer Device Address
Payload [2…n-1]	User Data	User Data with n-2 octets

Note: it makes no sense to send "acknowledged" messages as broadcast messages.

Response Message

The response message is sent to the local Host Controller if an acknowledgement from the peer WiMOD has been received. In this case the status "Command successful" is set in the Status Information Field (see Chapter 2.2). If no response has been received after the maximum number of retransmission (including timeouts) have been sent, the status is set to “Command failed”.

Field	Content	Comment
DstID	Sender ID	Destination EP identifier, taken from preceding Request command.
SrcID	DATALINK_ID	Source EP identifier
Opcode	0x05	Acknowledged Data Response
Length	0	No payload

Event Message

This message is sent to a Host Controller as a result of a received “Acknowledged Data Request” message via RF link. The first two bytes in the payload field contain the source device address of the transmitting WiMOD.

Field	Content	Comment
DstID	DATALINK_ID	Destination EP identifier
SrcID	Sender ID	Source EP identifier
Opcode	0x06	Acknowledged Data Indication
Length	n, n > 2	Variable Length
Payload[0] Payload[1]	Device Address [Bits 0…7] Device Address [Bits 8…15]	Source Device Address
Payload[2…n-1]	User Data	User Data with n-2 octets

4 Global System Parameter

This chapter outlines several parameter and constants:

4.1 Global Module Identifier

The following table shows the global module type identifiers which are part of the Device Information Field:

WiMOD	Module Type Value
iM820A	0x01
iM201A	0x02
iM240A	0x03
iM240B	0x04
iM241A	0x05
iM860A	0x07

4.2 Global System Endpoint Identifier

The following table outlines the global system endpoint identifiers which are used to access dedicated software modules.

Endpoint	Abbreviation	Value	Comment
Debug Handler	DEBUG_ID	0xFF	Endpoint for handling of debug messages. Expected on host system
Device Management	DEVMGMT_ID	0x90	Endpoint for Device Management services
RF Data Exchange	DATALINK_ID	0x91	Endpoint for RF Data Exchange services

5 Communication over UART

The WiMOD HCI Protocol uses a SLIP wrapping layer when transmitted over asynchronous serial interfaces (UART). The SLIP layer provides a mean to transmit and receive complete data packets over a serial interface. The SLIP coding is according to RFC 1055 [http://www.faqs.org/rfcs/rfc1055.html]

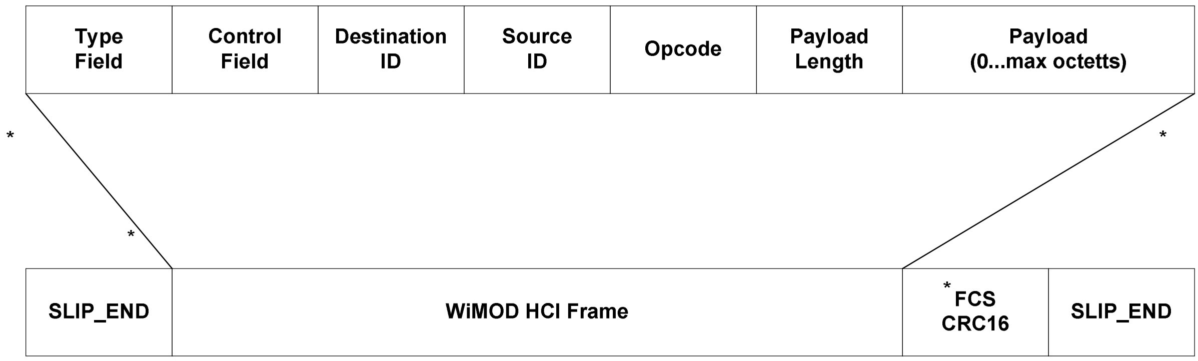


Fig. 5-1: HCI Message Format over UART

Following the HCI message a 16-Bit frame check sequence is added to support a reliable packet transmission. The FCS contains a 16-Bit CRC-CCITT cyclic redundancy check which enables the receiver to check a packet for bit errors. The CRC computation starts from the Type Field and ends with the last Payload octet.

An example implementation for SLIP Coding and CRC calculation is given below.

5.1 Physical Parameters

The standard UART settings are:

38400 bps, 8 Databits, No Parity Bit , 1 Stop Bit

5.2 Example Code

This chapter provides some example C-Code which demonstrates how to implement the WiMOD Host Controller Interface via UART.

ExampleCode.zip

Content

Source File	Comment
App/main.c	Main program,
WiMODHCI/WiMODAPI.c/.h	Application Interface of the Host Controller Interface This functions are intended to used by the application
WiMODHCI/WiMODHCI.c/.h	Message Interface of the Host Controller Interface This functions are intended to used by WiMODAPI Layer
WiMODHCI/SLIP.c/.h	SLIP Layer This functions are intended to used by WiMODHCI Layer
WiMODHCI/CRC16.c/.h	CRC16 calculation This functions are intended to used by WiMODHCI Layer
System/System.c/.h	Minimum runtime system with support for SW timers and SW tasks This functions are intended to used by WiMODHCI Layer and other Application specific software parts
System/UART.c/.h	UART example code This functions are indented to used by SLIP Layer
System/bsp.h	Standard Definitions

6 Communication over SPI

The WiMOD Firmware provides a Host Interface via UART and also via SPI. Some WiMOD Devices can either operate as SPI Master or SPI Slave. The WiMOD HCI Message will be extended by means of a leading Message Length Field which contains the message size in number of octetts.

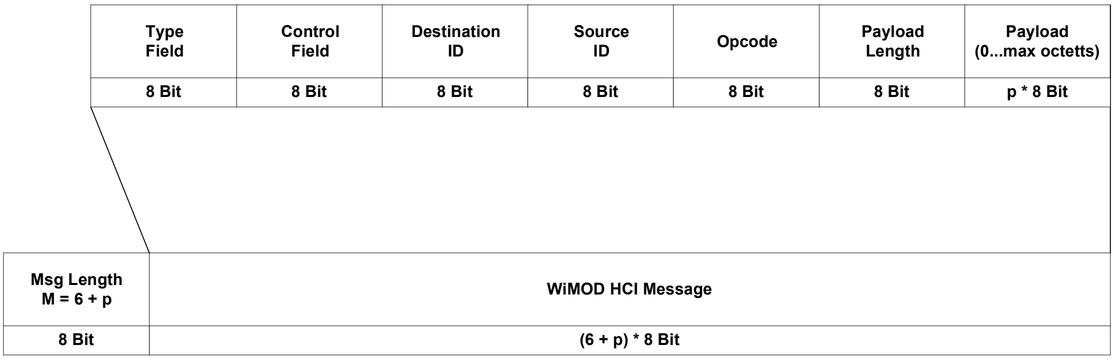


Fig.2.4 HCI Message Format over SPI

This communication link assumes an error free physical layer thus no further CRC Field is implemented.

The next chapters describe the principle timing for message transfer.

6.1 Master initiated SPI Transfer

The Master initiated SPI transfer requires the following signals:

1. Slave Select: The SPI Master indicates the start of a message transfer to the SPI Slave

2. MOSI: The **M**aster **O**ut **S**lave **I**n Signal line carries the data sent from the SPI Master to the SPI Slave

3. CLK: The Clock Signal is always driven by the SPI Master

4. Slave Ready: The SPI Slave indicates the receiption of a valid Length Field and readiness to receive a complete HCI Message.

Principle Signal Flow

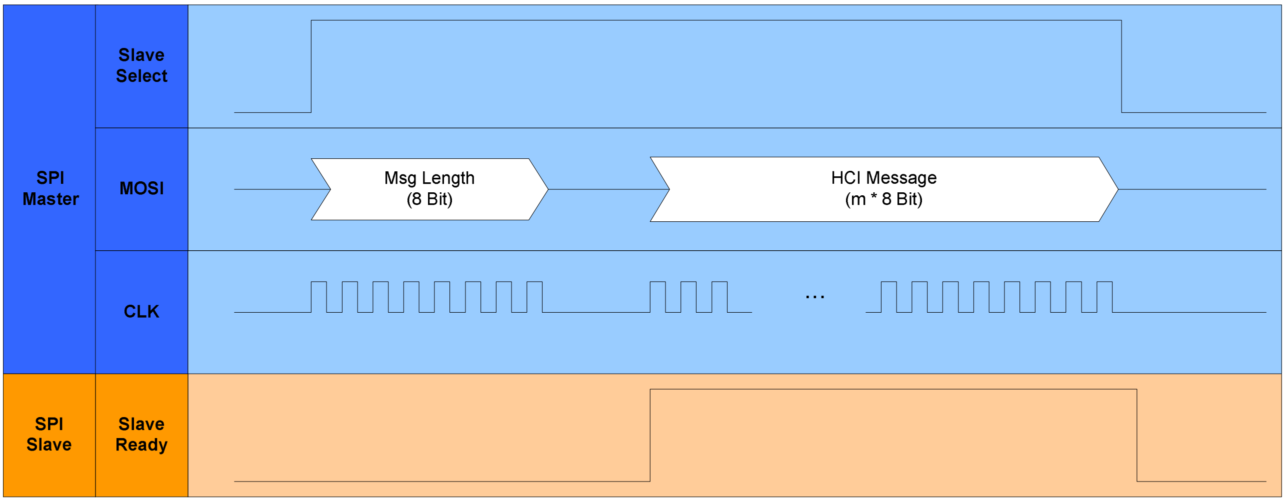


Fig. 6.1: Master initiated transfer (principle timing)

Note: Signal timings and polarities are given in the corresponding WiMOD specific document (e.g. IM820A_spi_timing.pdf)

The message transfer procedure is also outlined in the following flowchart:

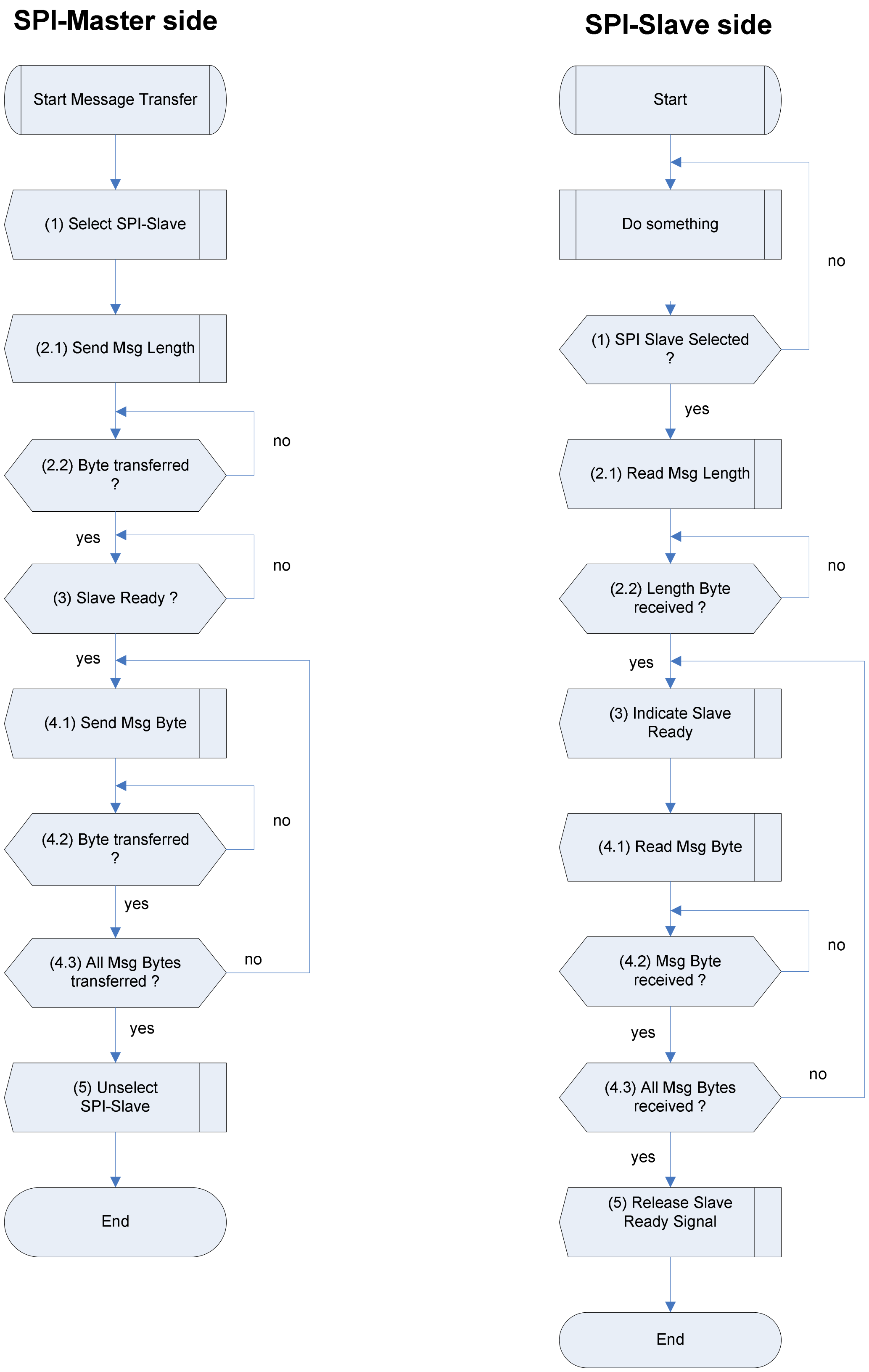


Fig. 6.2: Master initiated transfer (Flowchart)

6.2 Slave initiated SPI Transfer

The Slave initiated SPI transfer requires the following signals:

1. Slave Ready: The SPI Slave requests a message transfer to the SPI Master
2. MISO: The **M**aster **I**n **S**lave **O**ut Signal line carries the data sent from the SPI Slave to the SPI Master
3. CLK: The Clock Signal is always driven by the SPI Master
4. Slave Select: The SPI Master indicates the start of Message Length Field and HCI Message transfer to the SPI Slave

Principle Signal Flow

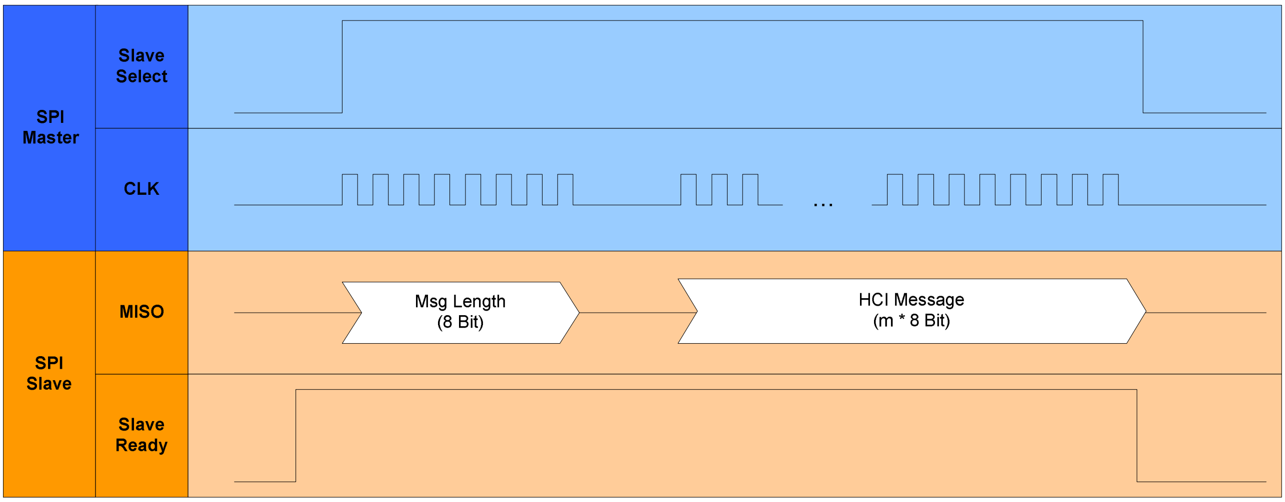


Fig. 6.3: SPI Slave initiated SPI Transfer

Note: Signal timings and polarities are given in the corresponding WiMOD specific document (e.g. IM820A_spi_timing.pdf)

The transfer procedure is also outlined in the following flowchart:

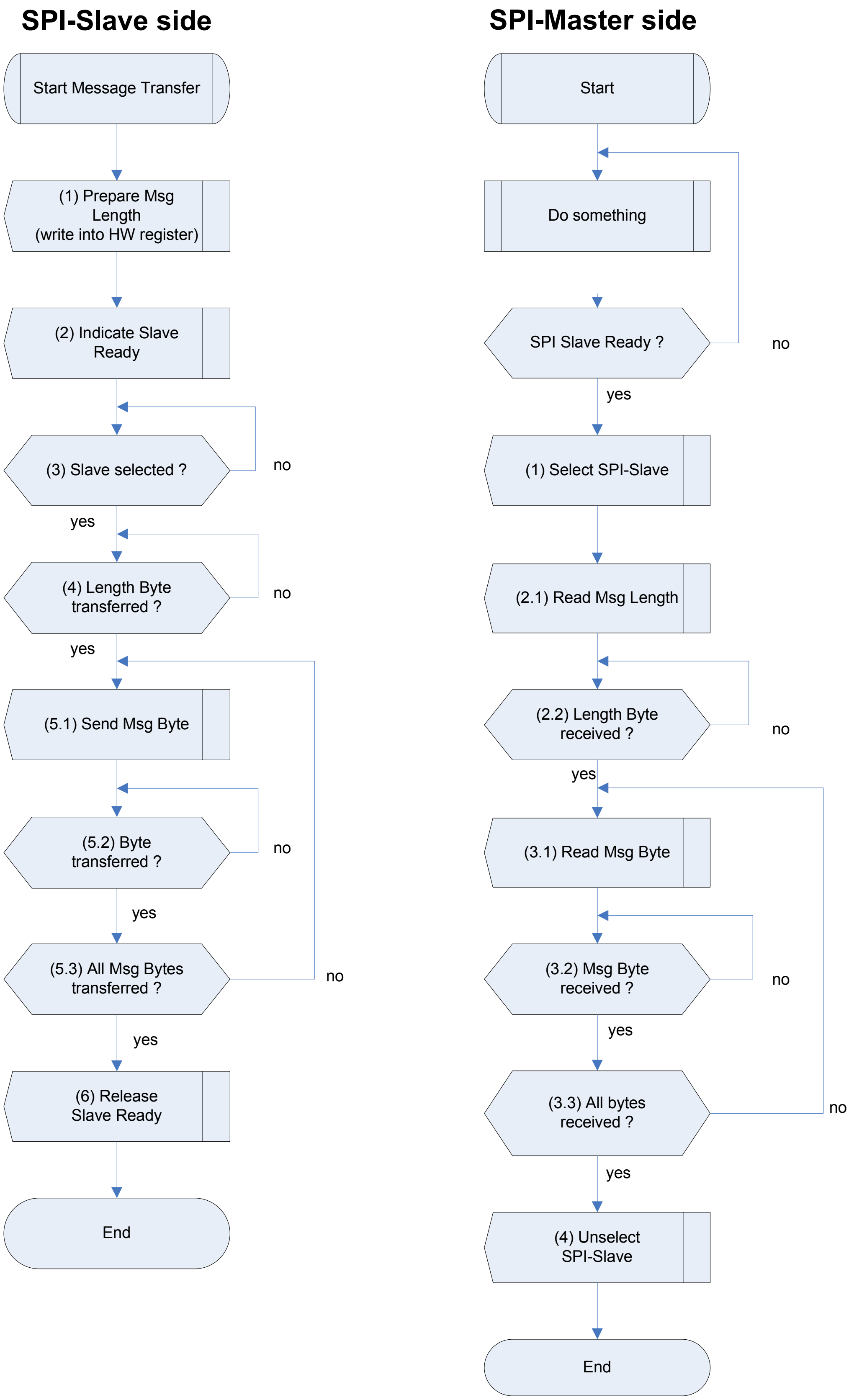


Fig. 6.4: Slave initiated transfer (Flowchart)

6.3 Physical Parameters

Note: Signal timings and polarities are given in the corresponding WiMOD specific document (e.g. IM820A_spi_timing.pdf)

7 Important Notice

7.1 Disclaimer

IMST GmbH points out that all information in this document is given on an “as is” basis. No guarantee, neither explicit nor implicit is given for the correctness at the time of publication. IMST GmbH reserves all rights to make corrections, modifications, enhancements, and other changes to its products and services at any time and to discontinue any product or service without prior notice. It is recommended for customers to refer to the latest relevant information before placing orders and to verify that such information is current and complete. All products are sold and delivered subject to “General Terms and Conditions” of IMST GmbH, supplied at the time of order acknowledgment.

IMST GmbH assumes no liability for the use of its products and does not grant any licenses for its patent rights or for any other of its intellectual property rights or third-party rights. It is the customer’s duty to bear responsibility for compliance of systems or units in which products from IMST GmbH are integrated with applicable legal regulations. Customers should provide adequate design and operating safeguards to minimize the risks associated with customer products and applications. The products are not approved for use in life supporting systems or other systems whose malfunction could result in personal injury to the user. Customers using the products within such applications do so at their own risk.

Any reproduction of information in datasheets of IMST GmbH is permissible only if reproduction is without alteration and is accompanied by all given associated warranties, conditions, limitations, and notices. Any resale of IMST GmbH products or services with statements different from or beyond the parameters stated by IMST GmbH for that product/solution or service is not allowed and voids all express and any implied warranties. The limitations on liability in favor of IMST GmbH shall also affect its employees, executive personnel and bodies in the same way. IMST GmbH is not responsible or liable for any such wrong statements.

Contact us to get information about the Declaration of Conformity.

7.2 Contact Information

IMST GmbH

Carl-Friedrich-Gauss-Str. 2
47475 Kamp-Lintfort
Germany

T +49 2842 981 0 **E** wimod@imst.de
F +49 2842 981 299 **I** www.wireless-solutions.de