

**Contents**

1 Introduction.....6

    1.1. Organization of this Document .....6

2 Functional Description.....7

    2.1. Interfaces.....7

        2.1.1. UART Interfaces.....7

        2.1.2. SPI Interface .....8

        2.1.3. Network Interface.....9

    2.2. Frame Format.....9

        2.2.1. Commands and Responses.....9

        2.2.2. Checksum Computation.....10

        2.2.3. Escaping .....10

    2.3. Frame Buffering.....11

    2.4. Resets.....11

    2.5. Addressing.....11

        2.5.1. PAN Identifier .....11

        2.5.2. PAN Address.....11

        2.5.3. IPv6 Addresses .....12

    2.6. The UDP Protocol.....15

    2.7. Data Transmission and Reception .....16

    2.8. Network Configuration .....16

        2.8.1. Physical Parameters.....16

        2.8.2. Software Parameters .....16

    2.9. Network Discovery.....20

    2.10. Security.....20

        2.10.1. Internet Protocol Security (IPSec).....20

        2.10.2. Internet Key Exchange (IKEv2).....22

        2.10.3. Important Security Considerations and Recommendations .....22

    2.11. Firmware Over-the-Air Update (OTAU).....23

    2.12. General Purpose I/Os.....23

    2.13. Watchdog Timer .....25

    2.14. Persistent Parameter Storage .....26

3 Command Reference .....27

    3.1. Transmit Frame Command.....29

    3.2. Configure Receiver Command .....30

    3.3. Get TRX Statistics .....31

    3.4. Reset TRX Statistics.....31

    3.5. Configure Wake-up Command.....32

    3.6. Power-Down Command .....35

    3.7. Get Address Configuration Command .....35

    3.8. Set PAN Address.....36

3.9.	Set PAN ID .....	37
3.10.	Configure PHY .....	37
3.11.	Configure NET .....	39
3.12.	Discover Network .....	40
3.13.	Remote Execute .....	41
3.14.	Get Firmware Version.....	42
3.15.	Reset .....	42
3.16.	Network Reset .....	42
3.17.	Store Configuration.....	43
3.18.	Test.....	43
3.19.	Configure LEDs Command.....	44
3.20.	Configure GPIO .....	45
3.21.	Write GPIO .....	45
3.22.	Read GPIO .....	46
3.23.	Toggle GPIO.....	46
3.24.	Configure UART1 .....	47
3.25.	Configure UART2 .....	48
3.26.	Configure SPI .....	48
3.27.	Restore Factory Settings .....	49
3.28.	Configure Multicast.....	49
3.29.	Join Multicast Group.....	50
3.30.	Leave Multicast Group.....	50
3.31.	Add Security Policy.....	51
3.32.	Add Security Association.....	52
3.33.	Add IKEv2 Authentication Entry .....	53
3.34.	Enable Watchdog Timer .....	54
3.35.	Get Current Interface.....	54
3.36.	Get PHY Configuration .....	54
3.37.	Get NET Configuration .....	55
3.38.	Get FCC ID Command .....	55
3.39.	Enable Acknowledge Command .....	55
3.40.	Set Parameter.....	56
3.41.	Get Parameter .....	57
3.42.	Receive Packet Command .....	58
3.43.	Remote Response Command .....	58
3.44.	Reset Report Command.....	59
3.45.	Network Configured Command .....	59
4	Glossary .....	60
5	Related Documents.....	62
6	Document Revision History .....	63

## List of Figures

Figure 2.1	Frame Format .....	9
Figure 2.2	General IPv6 Address Layout.....	12
Figure 2.3	IPv6 Multicast Address Layout .....	13
Figure 2.4	Multicast Addressing: Group ID Configuration .....	15
Figure 2.5	Example of Network Topology 1 .....	17
Figure 2.6	Example of Network Topology 2.....	18
Figure 2.7	IPSec Working Principle .....	21
Figure 2.8	Layout of the Pin Configuration 8-Bit Field.....	25
Figure 3.1	Configure Wakeup - Peripheral Field Format.....	34
Figure 3.2	Remote Execute Command Frame Layout .....	41
Figure 3.3	UART Configuration Bit-Field Layout .....	47
Figure 3.4	SPI Configuration Bit-Field Layout.....	48

## List of Tables

Table 2.1	Interface Dispatch Numbers .....	7
Table 2.2	UART Configuration and Pin Description .....	7
Table 2.3	UART Configuration Options .....	8
Table 2.4	SPI Pin Description .....	8
Table 2.5	SPI Configuration Options .....	8
Table 2.6	General Error Responses .....	10
Table 2.7	GPIO Operating Modes .....	23
Table 2.8	Default I/O Configuration of Preconfigured Pins.....	24
Table 3.1	Command Overview .....	27
Table 3.2	Transmit Frame Command Fields .....	29
Table 3.3	Transmit Frame Command Responses.....	29
Table 3.4	Configure Receiver Command Fields.....	30
Table 3.5	Configure Receiver Command Responses .....	30
Table 3.6	Get TRX Statistics Response Format.....	31
Table 3.7	Transmission Duty-Cycle Requirements of Sub-GHz Devices in the European Union .....	31
Table 3.8	Reset TRX Statistics Responses.....	31
Table 3.9	Wakeup Sources .....	32
Table 3.10	Configure Wakeup Command Fields.....	33
Table 3.11	Configure Wakeup Command Responses .....	34
Table 3.12	Power-Down Command Fields .....	35
Table 3.13	Power-Down Command Responses.....	35
Table 3.14	Get Address Configuration Response Fields .....	36
Table 3.15	Set PAN Address Command Fields .....	36
Table 3.16	Set PAN Address Command Responses .....	36

Table 3.17	Set PAN ID Command Fields .....	37
Table 3.18	Set PAN ID Command Responses .....	37
Table 3.19	List of Available Radio Channels .....	37
Table 3.20	Configure PHY Command Fields .....	38
Table 3.21	Configure PHY Command Responses .....	38
Table 3.22	Configure NET Command Fields .....	39
Table 3.23	Response Codes of the Configure NET Command .....	40
Table 3.24	Remote Execute Response Codes .....	41
Table 3.25	Get Firmware Version Command Responses .....	42
Table 3.26	Reset Responses .....	42
Table 3.27	Network Reset Responses .....	43
Table 3.28	Store Configuration Command Fields .....	43
Table 3.29	Store Configuration Command Responses .....	43
Table 3.30	Test Command Fields .....	43
Table 3.31	Test Command Responses .....	43
Table 3.32	Configure LEDs Command Fields .....	44
Table 3.33	Configure LEDs Command Responses .....	44
Table 3.34	Configure GPIO Command Fields .....	45
Table 3.35	Configure GPIO Responses .....	45
Table 3.36	Write GPIO Command Fields .....	45
Table 3.37	Write GPIO Responses .....	45
Table 3.38	Read GPIO Command Fields .....	46
Table 3.39	Read GPIO Command Responses .....	46
Table 3.40	Toggle GPIO Command Fields .....	46
Table 3.41	Toggle GPIO Command Responses .....	46
Table 3.42	Configure UART Command Fields .....	47
Table 3.43	Configure UART Command Responses .....	47
Table 3.44	Configure SPI Command Fields .....	48
Table 3.45	Configure SPI Command Responses .....	48
Table 3.46	Restore Factory Settings Command Fields .....	49
Table 3.47	Restore Factory Settings Command Responses .....	49
Table 3.48	Configure Multicast Command Fields .....	49
Table 3.49	Configure Multicast Command Fields .....	49
Table 3.50	Join Multicast Group Command Fields .....	50
Table 3.51	Join Multicast Field Command Responses .....	50
Table 3.52	Leave Multicast Group Command Fields .....	50
Table 3.53	Leave Multicast Group Command Responses .....	50
Table 3.54	Add Security Policy Command Fields .....	51
Table 3.55	Add Security Policy Command Responses .....	52
Table 3.56	Add Security Association Command Fields .....	52
Table 3.57	Add Security Association Command Responses .....	53

---

Table 3.58	Add IKEv2 Authentication Entry Command Fields .....	53
Table 3.59	Add IKEv2 Authentication Entry Command Responses .....	53
Table 3.60	Enable Watchdog Timer Command Fields .....	54
Table 3.61	Enable Watchdog Timer Responses .....	54
Table 3.62	Get Current Interface Responses .....	54
Table 3.63	Get PHY Command Response .....	54
Table 3.64	Get NET Configuration Command Response .....	55
Table 3.65	Get FCC ID Response .....	55
Table 3.66	Enable Acknowledge Command Format .....	55
Table 3.67	Response for Enable Acknowledge Command .....	55
Table 3.68	Set Parameter Command Format .....	56
Table 3.69	Set Parameter Responses .....	56
Table 3.70	Network Stack Configuration Parameters .....	56
Table 3.71	Get Parameter Command Format .....	57
Table 3.72	Get Parameter Response Format .....	57
Table 3.73	Receive Packet Command Fields .....	58
Table 3.74	Remote Response Command Fields .....	58
Table 3.75	Reset Report Command Fields .....	59

## 1 Introduction

This document describes the features and the usage of the ZWIR45xx module with the Serial Command Interface (SCI) firmware. The SCI allows using ZWIR45xx modules without programming the module. The ZWIR45xx modules are delivered preprogrammed and tested and can be integrated into the application without programming. The module provides several serial interfaces that can be used independently to control the module. Refer to the data sheet for the ZWIR45xx for ordering information.

The SCI module provides firmware over-the-air update (OTAU) capability, data encryption, and data authentication in addition to the normal User Datagram Protocol (UDP) communication. Integration into normal computer networks is possible without restrictions.

### 1.1. Organization of this Document

The following section gives a functional overview of the serial command interface firmware. The different interfaces, mechanisms, and functionalities are explained in this section. Section 3 documents all available commands.

Hexadecimal values are represented with the standard programming prefix of *0x*. To identify decimal and binary number representations if there is a risk of misinterpretation, a subscript <sub>BIN</sub> or <sub>DEC</sub> is added.

In section 3, command argument fields and responses are explained in a table format. Interpret tables for command argument fields as a list: the first table row is the first command argument; the second row represents the second argument, and so on. For responses, the same rule applies if there is no code field in the table. Otherwise, the table is interpreted in the following way: The code is the first field contained in the response. All fields (second column of the table) in the same row as the code field follow the code in the response.

## 2 Functional Description

### 2.1. Interfaces

The module provides two UART and one SPI interface for communication with external hosts. The host can also execute commands over the network interface. Hosts can be computers, microcontrollers, or even single sensors that have an appropriate interface. It is possible to connect different devices to different interfaces. All communication interfaces are enabled in the SCI default configuration.

The SCI firmware allows sending incoming network data to any of the available interfaces. For this purpose, the Configure Receiver command expects a dispatch value that specifies to which of the serial interfaces incoming data is sent. Table 2.1 shows the dispatch numbers for the different interfaces.

**Table 2.1 Interface Dispatch Numbers**

Interface	Dispatch Value
UART1	0
UART2	1
SPI	2
Network	3

#### 2.1.1. UART Interfaces

The module provides two UART interfaces. By default, both interfaces operate at a data rate of 115200 baud, have 8 data bits, one stop bit, no parity bit, and no flow control. The UART configuration can be changed at any time using one of the commands Configure UART1 or Configure UART2. Any interface can be used to change the UART configuration.

**Table 2.2 UART Configuration and Pin Description**

UART-Pin Name	Usage	Direction	ZWIR4512AC1 Pin		ZWIR4512AC2 Pin	
			UART1	UART2	UART1	UART2
Transmit (TX)	Mandatory	Output	13	6	14	6
Receive (RX)	Mandatory	Input	12	5	13	5
Request to send (RTS)	Optional	Output	17	7	20	7
Clear to send (CTS)	Optional	Input	16	8	19	8

**Table 2.3 UART Configuration Options**

Option	Default Configuration	Configuration Options
Data Transfer Rate	115200 baud	61 <sup>(1)</sup> baud to 256000 <sup>(2)</sup> baud
Parity	Not enabled	Odd, even
Number of Stop Bits	1	2
Flow-Control	Not enabled	Hardware Flow Control
(1) 61 baud can only be achieved on UART2. The minimum baud rate of UART1 is 122. (2) The maximum rate of 256000 baud can only be achieved on UART1. The maximum rate of UART2 is 115200 baud.		

**2.1.2. SPI Interface**

The module provides an SPI interface that operates in slave mode. To control the SCI module via the SPI, the host computer must provide an SPI clock whenever data is to be sent or received. The availability of data from the SCI module is signaled by a dedicated line that is pulled low when data is available for read-out. If the host needs to receive data and no data is to be transmitted, sending zeroes is recommended. The ZWIR45xx SCI device will send zeroes over its MISO line if the master is transmitting a frame and no data is to be transmitted to the host.

The default configuration uses the SSN pin to activate the SCI node. Data is only received and transmitted if the SSN pin is pulled to low. The SPI is configured for MSB-first transmission and reception; the clock is low when idle; and the first clock transition is used as the data-capture edge.

**Table 2.4 SPI Pin Description**

SPI Pin Name	Direction	Usage	Module Pin
Master Out Slave In (MOSI)	Input	Mandatory	1
Master In Slave Out (MISO)	Output	Mandatory	2
Clock	Input	Mandatory	3
Slave Select	Input	Optional	4
Data Pending	Output	Optional	7

**Table 2.5 SPI Configuration Options**

Value	Default	Configuration Option
Clock Polarity	Low ( clock is low when idle )	High
Clock Phase	First edge captures data	Second edge captures data
Pending Pin Mapping	Module pin 7	



**2.1.3. Network Interface**

The network interface allows executing commands remotely. Basically, this works like the serial interfaces with the difference that command frames sent over the network interface are not required to carry the *START* byte and data escaping is not required. All available commands can be executed remotely. Restrictions apply for some commands.

Each SCI node listens to UDP port 4 for incoming command frames. If a command is received, it is executed and if there is any response from the command, the response is sent back to the device that requested execution of the command.

The SCI also implements the Remote Execute command, which allows initiation of a remote execution over one of the serial interfaces.

**2.2. Frame Format**

Command and responses are sent over the different physical communication interfaces in data frames that have a special format that allows detecting communication errors and introduces low overhead. All frames have a common format.

The frame format is illustrated in Figure 2.1. Each frame starts with the *START* delimiter *0x7E*. The following two bytes determine the length of the frame payload. Length information is stored in “little endian” format, so the least significant byte is stored first. The fourth byte determines the command to be executed. The next [LENGTH] bytes are the frame payload. At the end of the frame, a checksum byte is attached, which helps to detect communication errors.

**Figure 2.1 Frame Format**

<b>START</b> 1 byte	<b>LENGTH</b> 2 bytes	<b>CMD</b> 1 byte	<b>PAYLOAD</b> LENGTH bytes	<b>CHKS</b> 1 byte
------------------------	--------------------------	----------------------	--------------------------------	-----------------------

**2.2.1. Commands and Responses**

The Serial Command Interface firmware distinguishes command and response frames. Command frames are sent in order to trigger a specific action at the receiver of the command frame. Response frames contain status information sent in response to command frames. Typically response frames are only sent if an error was encountered in the last command frame.

A command frame has a command code in the range of *0x01* to *0x7F*. The values *0x00*, *0x1B*, and *0x7E* are not allowed. An overview of all available commands and their command codes can be found in Table 3.1.

The command code of a response frame is the command code of the command frame that triggered the response OR’ed with *0x80*. The response frame with the command code *0x80* is a special response frame that is sent when an invalid frame was received or another non-command-related error occurred.

**Table 2.6 General Error Responses**

Code	Field	Length	Description
1	Invalid Header	3	Message larger than the maximum allowed size or unknown command.
2	-		Invalid checksum (CHECKSUM_ERROR)
3	Interface	1	A framing, parity, overflow, or noise error at the UART or SPI interface occurred.
	Command	1	
	Position of Last Valid Byte	2	
4	Internal Error Code	4	An internal error occurred.
5	-		Duplicate address detection failed.
30	Failing IPv6 Address	16	The address resolution failed for the address returned in the response. There is a delay before this error occurs after an attempt to send a unicast frame to a destination.

**2.2.2. Checksum Computation**

At the end of each packet, a checksum is appended that helps to detect transmission errors on the physical transmission medium. The checksum is formed in such way that the sum of all transmitted bytes, excluding the START byte *0x7E* is zero. If the sum of all received bytes of a packet does not result in zero, a Frame Error Packet with its error code set to CHECKSUM\_ERROR is sent over the same interface on which the erroneous packet was received.

**2.2.3. Escaping**

If one of the characters *0x1B* or *0x7E* is contained in the data stream or one of the frame control fields, this character must be escaped by XOR'ing it with *0x80* and prepending it with *0x1B*. This modification has no effect on the length field of the frame, even if it is done in the payload section of the frame. However, escaping must be taken into account when calculating the checksum.

If the checksum of the device is one of the SCI firmware's two control characters, the checksum must be escaped as well. Escaping the checksum is done by sending the escape character *0x1B* followed by the computed checksum value minus *0x1B*.

The following examples illustrate how escaping must be performed:

7e 04 00 12 1a 1b 1c 1d 7c  
7e 04 00 12 1a 1b 9b 1c 1d e1

7e 1b 00 01 ff 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 7e 03 01 02 03 04 05 06 07 08 09 34  
7e 1b 9b 00 01 ff 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 1b fe 03 01 02 03 04 05 06 07 08 09 fe

7e 04 00 10 24 9f 32 79 7e  
7e 04 00 10 24 9f 32 79 1b 63

### 2.3. Frame Buffering

Each communication interface maintains its own message queue. In situations of high load, messages are queued temporarily in the message queue. Each interface has its own message queue. The message queue distinguishes between command frames and response frames. Response frames are queued at the beginning of the queue, while command frames are queued at the end of the queue. This ensures that response frames are sent immediately after the triggering command frame. However, if a command frame is being sent when a response frame is queued, the transmission of the command frame is completed before the response frame is sent.

### 2.4. Resets

The SCI module provides two types of reset: System Reset and Network Reset. System Reset puts the device into the state that it has after a power-on. It executes the complete startup procedure of the microcontroller and the transceiver in order to put them into a defined state. All settings are restored to the factory default values or to the configuration that was saved to the flash memory the last time. System Reset is also performed automatically in the event of an unrecoverable error. If System Reset is performed because of an error, an error message is sent over the interface from which the last instruction was received. This message usually contains an error description. The most common error for System Reset is a memory allocation failure, which can appear after changing the network configuration or after configuring multicast. Refer to section 3.11 regarding the Configure NET command and see section 3.28 for the Configure Multicast command.

Network Reset is used after changing the network configuration. It does not perform a System Reset. A Network Reset disconnects all open connections and applies changes in network parameters that have been applied previously. The command Configure NET performs a reset automatically to apply the settings that have been made. Network Reset does not affect the physical parameter set with Configure PHY (see section 3.10).

### 2.5. Addressing

Each SCI module has three types of addresses: the PAN identifier, the PAN address, and the IPv6 address(es), which are described in the following sections.

#### 2.5.1. PAN Identifier

This address identifies the network in which the device is operating. All devices that are part of the same network must use the same PAN identifier. Devices with different PAN identifiers are not able to communicate with each other, even if they are in physical range of each other and have the same PHY settings applied.

The PAN identifier is a 16-bit number. Its default value is `0xACCA`. This value is changed using the Set PAN ID command (see section 3.9).

#### 2.5.2. PAN Address

The PAN address is the hardware address of the device. The PAN address is also commonly known as the MAC address. The PAN address of the SCI module is a 64-bit number that is formed according to the EUI-64 rules. Each device in the network *MUST* have a unique PAN address. This address is used to generate locally and globally unique IPv6 addresses (see section 2.5.3.1). Usually applications do not have to process the PAN address directly.

IDT's ZWIR45xx SCI Modules have a globally unique address preprogrammed. This enables generation of locally and globally unique IPv6 addresses without requiring initial configuration. IDT does not recommend changing the PAN address of the device. However, if the application requires this, it is supported by the command Set PAN address (see section 3.8).

**2.5.3. IPv6 Addresses**

The IPv6 addresses are the addresses that the applications are handling. These addresses are 128-bit wide (i.e., 16-byte wide). Each device must have at least one IPv6 address in order to be able to communicate. Using IPv6 addresses, the application determines where packets are to be sent or received from. How IPv6 addresses are set up and how they are presented is explained below.

As it would be impractical to use the byte-wise notation known from the previous version IPv4, IPv6 introduces a new notation. IPv6 addresses are represented by eight 16-bit hexadecimal numbers that are separated by colons. An example for such an address is

**2001:0db8:0000:0000:1b00:0000:0ae8:52f1**

Leading zeroes of segments can be omitted as they do not carry information. The IPv6 notation allows omitting a sequence of zero-segments and representing it as double colon. With these rules, the above address can be written as

**2001:db8::1b00:0:ae8:52f1 or 2001:db8:0:0:1b00::ae8:52f1**

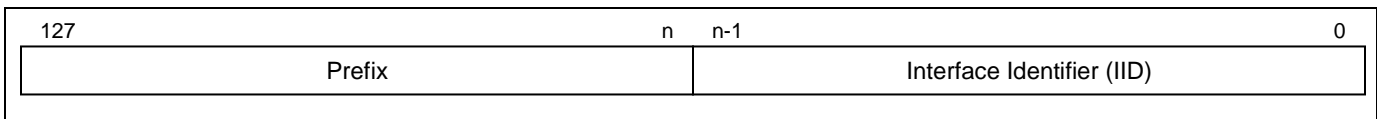
However, replacing multiple zero segments is not allowed. Thus the following address is invalid:

**2001:db8::1b00::ae8:52f1**

In order to explain IPv6 addresses, the term "link" must be defined. Nodes are said to be on the same link when they are able to communicate with each other without requiring any routers. Nodes on the same link are able to communicate on the MAC level.

An IPv6 address consists of two components: a prefix and an interface identifier. The prefix specifies the network that a device is part of; the interface identifier specifies the interface of a device. The size of the prefix varies for different address types. In the IPv6 address notation, the prefix length can be appended to the address with a slash followed by the number of prefix bits. Thus the notation 2001:db8::/64 represents a network containing the addresses from 2001:db8:: to 2001:db8::ffff:ffff:ffff:ffff.

**Figure 2.2 General IPv6 Address Layout**



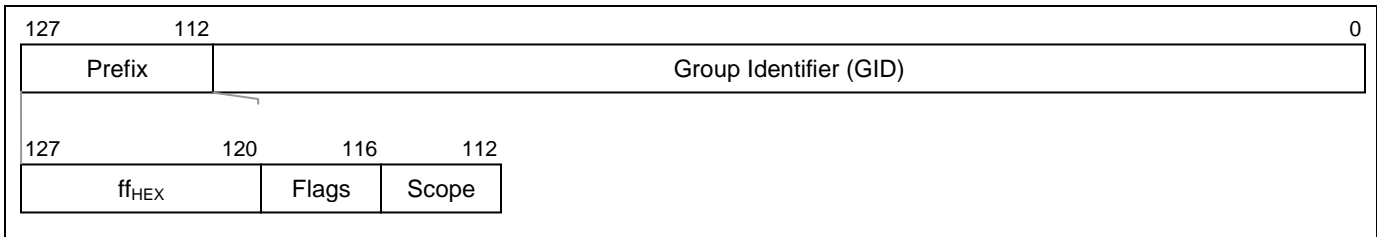
IPv6 supports three kinds of addressing methodologies: unicast addressing, multicast addressing, and anycast addressing. Unicast addressing is used to communicate to well-defined remote nodes. When multicast addressing is used, packets can be sent to multiple addresses at the same time. Anycast addressing is used to reach any node in a group of nodes that share the same anycast address.

IDT's SCI firmware does not allow nodes to have anycast addresses. However, remote anycast nodes (e.g., computers in the network) can be reached. A network node can have multiple interfaces; e.g. it might incorporate multiple radio nodes or multiple Ethernet cards.

Unicast addresses are used to address a single interface in the network. The prefix of the address determines the scope of the unicast address. This can be link-local or global. All unicast addresses with a prefix starting with 000<sub>BIN</sub> (bits 125 to 127) have a 64-bit interface identifier. If the prefix equals fe80::/64, this is a link-local unicast address. Link local addresses are valid only on the link to which the interface is connected. The prefix of global unicast addresses is typically received from a router via address auto-configuration. If the router is not connected to the Internet, the node will not be assigned a global address, only a local one. Besides local and global prefixes, there are additional prefix configurations with limited scope that are not covered by this documentation. Refer to *RFC 4291 – IP Version 6 Addressing Architecture* (see section 5) for more detailed information about alternative prefix configurations.

Multicast allows sending packets to multiple receivers at the same time. For this purpose, IPv6 provides multicast addresses. This class of addresses must only be used as destination address. It *MUST NOT* be used as a source address in IPv6 packets. Figure 2.3 shows the layout of multicast addresses. They have a 16-bit prefix with the most significant eight bits set to 0xff, followed by two 4-bit fields for flags and the scope of the multicast packet. The lower 112 bits specify the multicast group id. A device checks if it is part of a multicast group depending on the multicast group ID.

**Figure 2.3 IPv6 Multicast Address Layout**



For the flags field, IDT's IPv6 stack only supports the values 0000<sub>BIN</sub> and 0001<sub>BIN</sub>. The value 0000<sub>BIN</sub> specifies that the multicast address is a well-known address (an address that has been assigned by the IANA). The value 0001<sub>BIN</sub> marks the address as a temporarily assigned address that is not specified by internet standards. Custom multicast addresses *MUST* use the latter version of multicast addresses!

For the scope field, any allowed value is supported. However values above 0010<sub>BIN</sub> require appropriately configured routers.

Two addresses should be noted as they are used frequently:

1. The unspecified address, which is represented as :: because all segments of this address are zero. It is used by receivers to listen to any sender. This address *MUST* never be used as the destination address.
2. The link-local all nodes multicast address ff02::1. Packets sent to this address are received by all nodes on the link. Thus this multicast address is equivalent to link-local broadcasting.

For further information about IPv6 addressing, refer to *RFC 4291*.

### 2.5.3.1. IPv6 Address Auto-configuration

In order to make node configuration and setup as easy as possible, IPv6 provides a stateless auto-configuration mechanism for IPv6 addresses. Using this algorithm, IPv6 unicast addresses are generated automatically from information statically available on the interface and information provided by routers on the link. If no router is present, only an IPv6 address with the link-local scope is generated. Global addresses are generated if the router advertises prefix information. Thus, no manual address assignment is required and no server-based address assignment technology, such as DHCP, needs to be provided.

The stateless address auto-configuration mechanism uses the PAN address (refer to section 2.5.2) of the interface to generate the device's IPv6 addresses. For both address scopes, link-local and global scope, this is done by putting a 64-bit prefix in front of the modified 64-bit PAN address of the interface. The modification of the PAN address refers to toggling the universal/local bit of the interface identifier. This method is described in Appendix A of *RFC 4291*. Assuming a PAN address of 00:11:7d:00:00:12:34:56 and a prefix 2001:db8::/64, this would result in the global IPv6 address 2001:db8::211:7d00:12:3456. Accordingly the link-local address would be generated by using the link-local prefix fe80::/64 and result in the IPv6 address fe80::211:7d00:12:3456. Note that in both cases, the modification coming from the modified PAN address is highlighted in red.

IPv6 requires that addresses that are assigned to an interface are checked for their uniqueness on the link. This is done using the Duplicate Address Detection (DAD) algorithm. DAD is not required if the universal/local bit in the interface identifier of the IPv6 address is set to 1 (as in the example above). The DAD algorithm is described in more detail in section 2.5.3.3.

### 2.5.3.2. Multicast

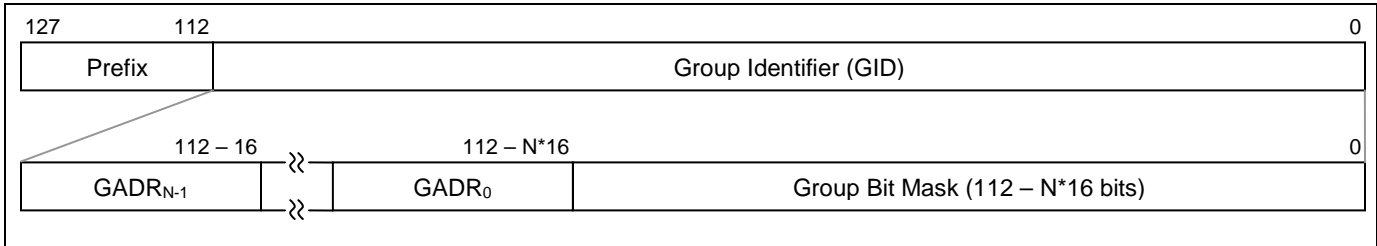
IPv6 multicast allows sending packets to multiple receivers at the same time. Sending packet to multiple receivers is done by simply using a multicast address as the destination address. All devices that want to receive packets sent to a particular multicast address must join the appropriate multicast group before they are able to receive packets. The receiver must be configured appropriately to receive packets from the sender. Refer to section 2.7 for more detailed information on configuring devices for data reception.

The ZWIR45xx SCI firmware provides a flexible multicast implementation supporting up to 16384 ( $2^{14}$ ) groups. Depending on the multicast configuration of the device, 7 to 112 groups can be addressed simultaneously. When multicast is configured, a bit-field is allocated in the devices with each bit representing the status of a certain group membership status of the device. Joining a certain group will set the corresponding bit in the bit-field internally, while leaving that group will delete the bit. Each device can join as many groups as it wishes.

In order to send data to one or multiple groups, the destination address must be formed appropriately according to the selected multicast configuration. IDT's multicast implementation allows flexible configuration of the way multicast groups are addressed in the network. Basically, the multicast group identifier (GID) is divided into two sections. The lower bits are interpreted as a bit mask that is logically AND'ed with the internally stored bit-field of a receiving device in order to determine if the device is part of the requested multicast group. The remaining bits that are not in the bit-mask are interpreted as 16-bit group addresses. To support multicast, multicast *MUST* be configured. During the multicast configuration, specify how many of the upper 16 fields of an IPv6 multicast address are interpreted as the group address (GADR). The remaining bits are interpreted as the bit-field.

The general layout of the multicast GID is shown in Figure 2.4. Parameter N is the configured number of group address fields. Note that all devices in the network must use the same multicast configuration.

**Figure 2.4 Multicast Addressing: Group ID Configuration**



### 2.5.3.3. Validation of IPv6 Address Uniqueness

Duplicate address detection (DAD) is performed to check if an IPv6 address is unique on the link. For this purpose, a node starts to send neighbor solicitation (NS) messages to the address to be checked. If another node replies to one of those messages or if another node also sends neighbor solicitation messages to this address, the assigned address is not unique and must not be used. In such cases, a system reset is performed, and an error message is sent to the interface from which the last command was received. Note that this error can only happen if some of the devices in the application use manually configured IPv6 addresses or PAN addresses. Therefore, it is recommended that manually configured addresses not be used!

## 2.6. The UDP Protocol

The User Datagram Protocol (UDP) is used for data transmission in ZWIR45xx devices. UDP is a connectionless and lightweight protocol, introducing minimal communication and processing overhead. No connection has to be created, and no network traffic is required before data transmission between nodes can be started. Instead, communication is possible immediately. UDP does not guarantee that packets that have been sent are reaching the receiver. If reliable transmission is required, acknowledges must be implemented on the application level. It is also possible that the same UDP packet is received multiple times. Furthermore, it is not guaranteed that the receive order of packets at the destination is the same as the send-order at the source. This must be considered by the application programmer.

UDP uses the concept of ports to differentiate different data streams to a node. A port can be seen as the address of a service running on the receiver of a packet. Depending on the destination port of a packet, the network stack decides to which network service the packet is routed. UDP distinguishes 65536 ports. The SCI firmware on the ZWIR45xx simply transmits data that have been received on a certain connection to one of its interfaces. It is possible to assign different interfaces to different ports.

The SCI firmware opens three UDP connections automatically. These connections are required for the remote-execution of commands, the Internet Key Exchange protocol, and the over-the-air firmware update feature (OTAU). The ports allocated for these services are listed below. They must not be used by the application.

- Port 4: Remote Execution
- Port 500: Internet Key Exchange version 2
- Port 1357: Over-the-Air Update

## 2.7. Data Transmission and Reception

Data transmission is requested with the Transmit Frame command. This command allows specification of the destination IPv6 address and the UDP port that the packet should be sent to. The source port of the transmission is selected randomly.

In order to receive packets from a remote device, the receiver must be configured appropriately. For that purpose, the command Configure Receiver is provided (see section 3.2). It allows configuring the source address of the device from which data should be received and the UDP port on which data should be received. It is possible to determine to which interface the data traffic received on the particular connection is sent. The source address argument *SHOULD* be a unicast address, if data reception from a particular device is expected. Alternatively it is possible to use the unspecified address, which will accept data from any sender.

For successful data transmission, the UDP port configured at the receiver must match the UDP port to which data is sent.

## 2.8. Network Configuration

ZWIR45xx network parameters are configurable to match the needs of the application. Especially for large installations with mesh routing, adjusting settings for optimal performance is strongly recommended.

### 2.8.1. Physical Parameters

ZWIR45xx's SCI firmware allows configuring the radio channel, the signal modulation, and the transmission power. In order to be able to communicate, all devices must use the same channel and modulation settings. All physical parameters are changed with the command Configure PHY (see section 3.10).

For the devices used in the European Union and worldwide areas that have adopted the EU's regulatory settings, the channels 0 and 100 to 102 can be selected. In this frequency band, the output power is limited to 5dBm. In the USA and worldwide areas adopting the USA regulatory settings, channels 1 to 10 are available. For these channels, a maximum transmission power of 10dBm can be selected. For both areas, the lower limit of the output power is -10dBm.

### 2.8.2. Software Parameters

The software parameters control the behavior of the network stack. All of these parameters are changed using the command Configure NET.

#### 2.8.2.1. IPv6 Network Parameters

The **Neighbor Cache Size** determines how many neighbor cache entries are allocated by the network. Neighbor cache entries are required for each remote endpoint with which a device wishes to communicate via unicast. For an example network topology, as shown in Figure 2.5, the devices must have the following number of neighbor cache entries:

D1 → 1 entry

D2 → 1 entry

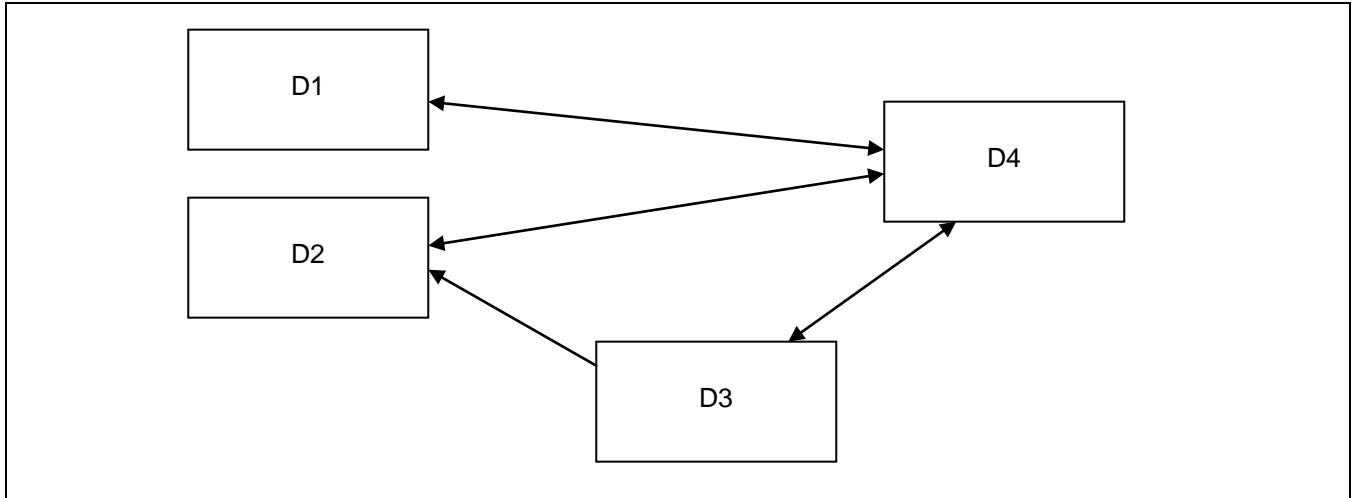
D3 → 2 entries

D4 → 3 entries

Thus for each outgoing connection (arrow is pointing away from device), a neighbor cache entry is required.



Figure 2.5 Example of Network Topology 1



The **Maximum Socket Count** parameter determines how many sockets can be open at the same time. This is important for concurrent reception of different data streams.

The **Neighbor Reachable Time** parameter determines the time a neighbor cache entry is considered as reachable before reachability is verified by the network stack automatically. Usually this parameter is configured by routers present in the network. Thus, if a router is present, the configured value is overwritten with the value received from router advertisements.

The **Duplicate Address Detection Enable** flag determines if duplicate address detection (DAD) is performed during interface initialization or when a new address is assigned to the interface. Duplicate address detection is used to verify that no other device in the network is using the same address as the address verified. DAD causes one broadcast packet to be sent during interface initialization. The interface is not available for user communication before the DAD packet is sent and some delay to receive an answer has passed. The time before availability of the interface or a newly assigned IPv6 address is in the range of one to two seconds if DAD is enabled. If DAD is disabled, the interface or newly assigned addresses are available immediately and no packet is sent out by the device. However, leaving DAD enabled is recommended for protection from network failures caused by duplicate addresses. A device that is unable to initialize its interface correctly will report an error and go to Standby Mode until it is being reset.

The **Router Solicitation Enable** flag determines if the device sends router solicitations to the network. This is typically done to obtain network configuration information such as address lifetimes and reachable times as well as global network prefixes. If there will never be a router on the network, this option can be disabled. Otherwise the option should be left enabled. Usually routers send configuration information autonomously at a certain time interval. If delayed reception of router information is acceptable, the Router Solicitation Enable flag can be unset. However, care must be taken to ensure that the router information advertised on a regular basis contains the same amount of information as the solicited advertisements.

**2.8.2.2. Mesh Routing Parameters**

The mesh routing mechanism used in the ZWIR45xx network stack is configurable to control the behavior of nodes during the establishment of routes and to get better reliability of the whole network and single connections.

The first item that is configurable is the **Maximum Hop Count**. This parameter is applicable during the reception of multicast packets. If the received packets' Hop Count is greater than or equal to the value configured in this parameter, the packet is not forwarded again. Thus, a multicast packet is only forwarded when the packet's Hop Count is lower than the configured value. If this value is set to zero, the device will never work as a mesh network relay. If not set to zero, the value should be set to the maximum hop count expected to be occurring in the network. Setting the value to a much larger value is not recommended as this value is also required to stop the circulation of packets in the network. Circulating packets could occur very rarely under special circumstances. The larger the Maximum Hop Count value of a packet is, the longer a circulating packet will travel through the network.

In networks that have many devices with direct mutual reachability, limiting the number of devices that can serve as mesh relays is strongly recommended. Otherwise the network might tend to be unreliable, especially during the startup phase. The level of unreliability increases with the number of devices with mutual reachability acting as a mesh relay. A good "rule of thumb" is having a maximum of five to ten mesh relays with mutual reachability.

It has to be noted that devices with the **Maximum Hop Count** parameter set to zero are still able to send data to remote devices that are multiple hops away! Also reception of data is possible from a node that is multiple hops away. The parameter only controls whether the device is allowed to act as a network relay.

Each device in the network maintains a routing table storing the link-layer address of the next hop to be taken to a certain destination. The **Routing Table Size** parameter determines how many routing table entries are available in the device. A routing table entry is required for each endpoint to which a device is communicating in the network. For relay-enabled nodes, this includes the communication endpoints of the nodes using the nodes relay service.

**Figure 2.6 Example of Network Topology 2**

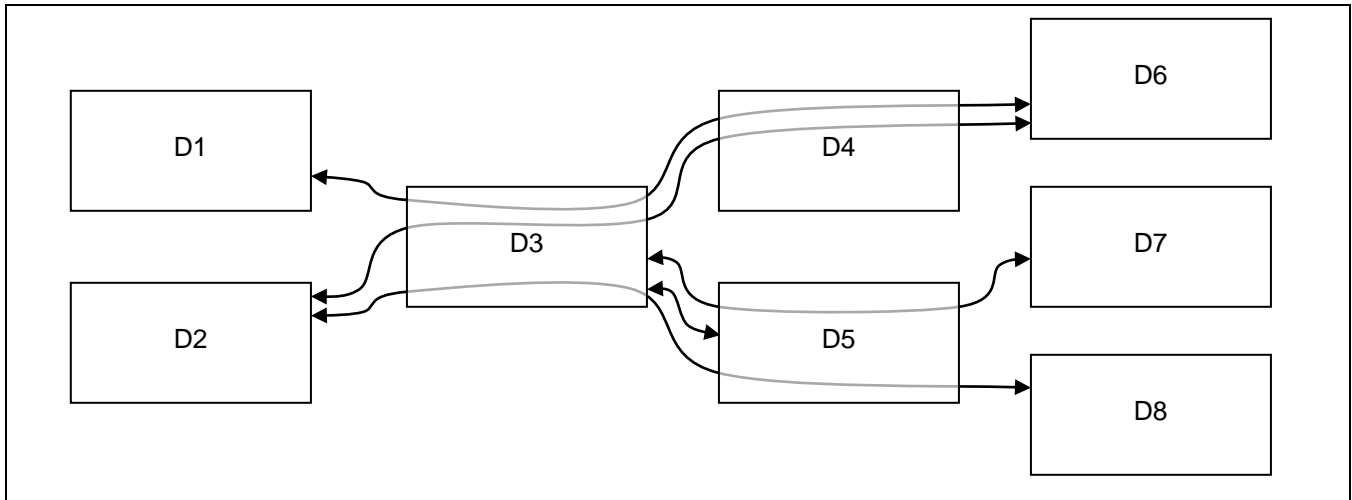


Figure 2.6 shows an example network topology with different communication paths. The lines going through D3, D4, and D5 symbolize multi-hop communication paths. For demonstration, the list below shows how many routing table entries are required as a minimum at each node:

- D1 → 1 entry (D6)
- D2 → 2 entries (D6, D8)
- D3 → 6 entries (D1, D2, D5, D6, D7, D8)
- D4 → 3 entries (D1, D2, D6)
- D5 → 4 entries (D2, D3, D7, D8)
- D6 → 2 entries (D1, D2)
- D7 → 1 entry (D3)
- D8 → 1 entry (D2)

After a route has been set up, the routing table entry persists until one of the two following events occurs:

- (1) The route is not used for **Route Timeout** seconds.
- (2) **Route Max Fail Count** attempts of using the route have been failing.

The first case will simply remove the route from the routing table and will reestablish the route when it is required again. Thus if rare usage of a route is anticipated, the **Route Timeout** parameter can be increased accordingly. In contrast, for networks with frequently changing topologies, it might be advantageous to reduce the Route Timeout parameter to avoid failing attempts at using a broken route and instead reestablish the route on demand. In these networks, it might be advantageous to reduce the **Route Max Fail Count** parameter, as the probability of a hop being gone is high and therefore the route can be reestablished faster.

The reliability of transmissions can be increased using the **Route Discovery Minimum Link RSSI** parameter. This parameter puts constraints on the route discovery process. Only paths with an RSSI value equal to or greater than the configured value are considered for route setup. If a route cannot be established using the configured Route Discovery Minimum Link RSSI, the next attempt to setup the route is made with the previously used value reduced by the amount of **Route Discovery Minimum Link RSSI Reduction**. There are **Route Discovery Attempts** made to set up the route. If RSSI constraints have been put on route discovery, the resulting routes might choose a longer path (more hops). However, if the network is static, the route is more likely to succeed than the shorter route with a higher probability of communication failures on single segments of a route.

By default, the Route Discovery Minimum Link RSSI is set to a value of -128dBm and the reduction value is set to zero. The transceiver is capable of reporting a minimum value of -100dBm. Thus, the initial value allows any connection. In order to generate a very reliable route, an initial **Route Discovery Minimum Link RSSI** value of -80dBm could be chosen with a **Route Discovery Minimum Link RSSI Reduction** value of 1 and a **Route Discovery Attempts** value of 20. If possible, the route with an overall RSSI of -80dBm is chosen if these settings are applied. However, if no such route exists, this will take a very long time to determine. Therefore, typical RSSI reduction values are chosen higher.

Note that the route discovery process is typically initiated by the receiver of a unicast packet! The reason for this is that during address resolution, the receiver of the packet must send a unicast neighbor advertisement message.

## **2.9. Network Discovery**

The Discover Network command (see section 3.12) requests all devices in the PAN to send a network discover reply message to the requesting device. For each answer received, the requesting device sends an informational message to the interface on which the Discover Network command was received. The message contains information about all assigned IPv6 addresses, the link quality, and how many hops the device is away from the requester.

Network discovery must be used rarely and carefully! Each call to Discover Network causes a multicast packet to be sent to all nodes in the network. Each node receiving the request will try to answer it with a unicast message back to the device that the network discovery request was received from. Receiving nodes must wait for a random time up to a configured limit before they send their response. This helps to avoid too many devices trying to send their response at the same time and blocking each other. Depending on the size of the network, the upper time limit must be adjusted appropriately. As a “rule of thumb,” the number of nodes in the network divided by the time limit must not exceed a value of 5. The opportunity to receive a response from each device in the network increases with lowering the quotient.

## **2.10. Security**

Security has become one of the most discussed topics in wireless sensor networks. Most applications need some amount of security. Each application has different requirements. A well-designed security system should be able to support data confidentiality, data authentication, data integrity, and protection against replay attacks. Depending on the type of application, one or more of these concepts need to be supported.

The ZWIR45xx supports all of these security concepts with the IPsec (IPsec=Internet Protocol Security) protocol suite. This protocol suite is already widely used in Virtual Private Networks (VPNs) today. Additionally, IDT provides an implementation of the Internet Key Exchange protocol version 2 (IKEv2). Using this protocol, it is possible to generate shared keys for different IPsec links. Both protocols are open standards developed by the Internet Engineering Task Force (IETF).

Both, IPsec and IKEv2 are supported by all major operating systems and therefore allow end-to-end secure communication between ZWIR45xx devices and computers anywhere on the world.

### **2.10.1. Internet Protocol Security (IPsec)**

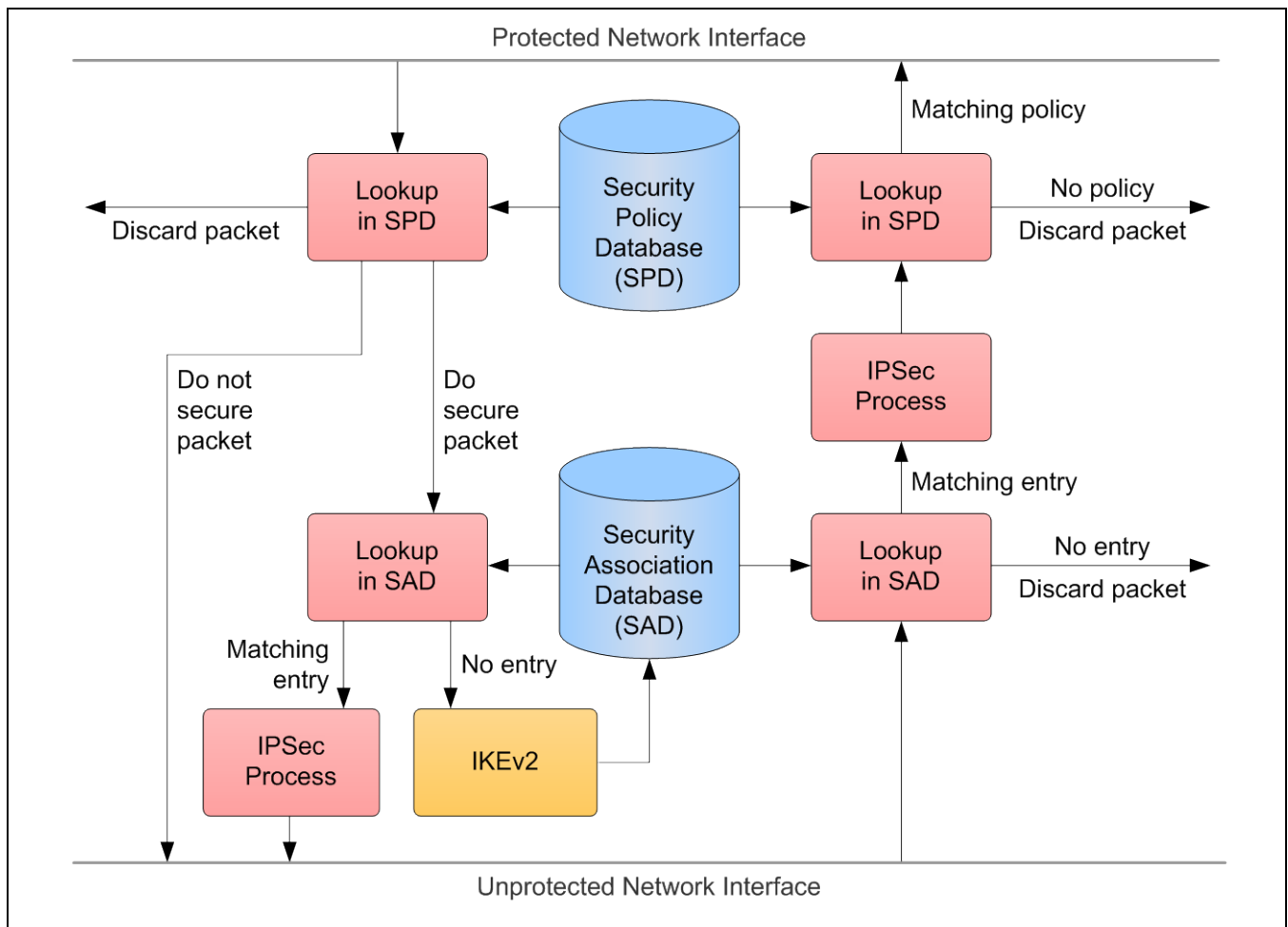
IPsec is a protocol suite for encryption and authentication of data sent over an IP network. IPsec is supported by virtually all modern operating systems running on normal PCs. IDT’s IPsec implementation supports the Encapsulating Security Payload (ESP) and Authentication Header (AH) protocols. ESP provides data encryption, data integrity, and protection against replay attacks. AH is applied if authentication of the packet sender is required in addition to the previously mentioned security services.

The basic working principle is depicted in Figure 2.7. Two databases are the central elements of IPsec: the Security Policy Database (SPD) and the Security Association Database (SAD). These databases must be configured appropriately to enable secure communication. This is done using the commands Add Security Policy (see section 3.31) and Add Security Association (see section 3.32). The latter is not required if IKEv2 is used for generating keying material (see section 2.10.2).

The information about how traffic to different destinations must be handled is stored in the Security Policy Database. Each policy entry applies for a remote address (range) with a port (range) and one or more protocols. Each packet that is going to be sent or received is compared to the different security policies stored in the SPD. If the packet does not match any of the policies, the packet is silently dropped. Thus, there *MUST* be rules for all types of traffic that need to be sent or received by the application. Such rules are added using the command Add Security Policy. If the application does not use IKEv2 for the automatic generation of keying material, a SAD entry must be available before the generation of a new security policy.

In addition to the remote device and protocol information, each SPD entry is linked to an entry in the Security Association Database. The SAD stores the keying material for encryption, decryption, and authentication as well as the encryption and authentication method to be used. The SAD can be administrated manually, or it will be filled automatically by the IKEv2 daemon. It is also possible to use a mixture of these two methods. In order to generate entries manually, use the command Add Security Association.

**Figure 2.7 IPSec Working Principle**



ZWIR45xx SCI modules are delivered with default settings that allow any incoming and outgoing traffic to pass. Thus, if the application does not need security features, there is no need to re-configure. If security is required, the first call to the Add Security Policy command will remove the default policy of allowing all traffic to pass unsecured. The newly configured policy will apply instead. Subsequent calls to Add Security Policy will add additional policies.

Note: Security policies are matched in the order of their generation! Therefore the rule with the most general traffic matching filter must be added as the last rule to avoid hiding other rules. A maximum of 10 security policies can be generated.

### **2.10.2. Internet Key Exchange (IKEv2)**

The Internet Key Exchange version 2 protocol is provided to enable automatic exchange of keying material for IPSec. In this case, security associations are generated automatically on demand. If network traffic is generated on a connection that requires some kind of security processing but there is no security association assigned, IKEv2 automatically initiates the key exchange procedure in order to establish a security association.

The need to establish a security association has two effects on the device:

1. The UDP packet that initiates the key exchange is not delivered to the receiver. Instead, an error message is returned to the host, reporting that the packet could not be delivered.
2. The key exchange procedure requires complex mathematical computations that take some time. For this reason, the internal clock frequency will be increased for the required time of about four seconds. During this time, the device is not able to execute commands. However, incoming data on the serial interfaces is buffered and executed after completion of the key exchange procedure.

In order to authenticate the communicating parties that wish to establish a security association using IKEv2, the application must register appropriate authentication entries with the IKEv2 daemon. These authentication entries consist of an identifier that is transmitted with the IKEv2 message and a 16-byte pre-shared key. This is done using the command Add IKEv2 Authentication Entry (see section 3.33). Up to five different IKEv2 authentication entries can be created.

### **2.10.3. Important Security Considerations and Recommendations**

IDT strongly recommends using security in almost any kind of application that is installed in the field. It is absolutely mandatory in applications for remote control, in applications that transmit confidential data, and in applications that transfer data that are used for billing purposes.

Applications requiring security *MUST* be designed in such way that no keying material is transferred over one of the serial links or over an unsecured wireless link. Otherwise it would be easy to break the security mechanisms. Thus, keying material of manually configured security associations and IKEv2 pre-shared keys must not be stored on the host processor but must be stored in the wireless module instead.

Getting these values into the module without giving attackers the chance of obtaining sensitive data is one of the most critical points in security-sensitive systems. How this can be handled must be considered in the starting phase of the project and must be reviewed carefully. A security system is never stronger than its weakest component!

### 2.11. Firmware Over-the-Air Update (OTAU)

The SCI firmware on the ZWIR45xx can be updated using the over-the-air (OTAU) update mechanism. This mechanism allows firmware updates for defect correction and feature extensions even if the product is already installed in the field. The firmware over-the-air daemon is permanently running on the device. It listens to UDP port 1357 for incoming firmware packets. Therefore this port must not be used by the application.

Refer to *ZWIR45xx Application Note—Enabling Firmware Over-the-Air Updates* for information on how to distribute a new firmware version to ZWIR45xx SCI devices via OTAU.

**Important Recommendation:** Put an IPSec security policy on UDP port 1357 for all remote IPv6 addresses. Otherwise, attackers might be able to install malicious software on the device using the over-the-air update mechanism. Refer to section 2.10 and *ZWIR45xx Application Note—Using IPSec and IKEv2 in 6LoWPANs*.

### 2.12. General Purpose I/Os

ZWIR45xx devices provide up to 19 general-purpose input and output (GPIOs) pins. GPIO pins can be used in different modes as shown in Table 2.7. In the default settings, some of these GPIOs are allocated for the different serial interfaces. However, if an interface is disabled, the pins that had been allocated are freed and available as GPIO.

**Table 2.7 GPIO Operating Modes**

Mode	Behavior	Configuration Options
Interface	The pin is controlled by hardware; however, if the pin is an input, it can be used as a wakeup pin.	None
Wakeup	The pin is configured as an input. The device is awakened from low-power mode on the configured activity.	Floating input or input with pull-up or pull-down <sup>1)</sup>
LED Driver	The pin is configured as an output. The SCI firmware controls output value.	Push/pull or open drain output with configurable driving strength
GPIO	The pin is freely configurable for use by the application. If the pin is an input, it can be used as a wakeup pin.	Floating or pull-up/ pull-down input Push/pull or open-drain output
Unused	All input and output circuitry is disabled for minimal power consumption.	None

1) Wakeup pins cannot be configured directly. When an I/O pin that is already configured as an input is used as a wakeup pin, the configuration is retained. If an unconfigured pin is selected as a wakeup pin, the pin is configured as a floating input.

In order to meet the needs of the application, all GPIOs are configurable easily and flexibly. Unconfigured devices only have pins configured if they belong to one of the serial interfaces. All other pins are in the unconfigured state to consume minimal power. Table 2.8 shows the default GPIO configuration of all preconfigured pins.

**Table 2.8 Default I/O Configuration of Preconfigured Pins**

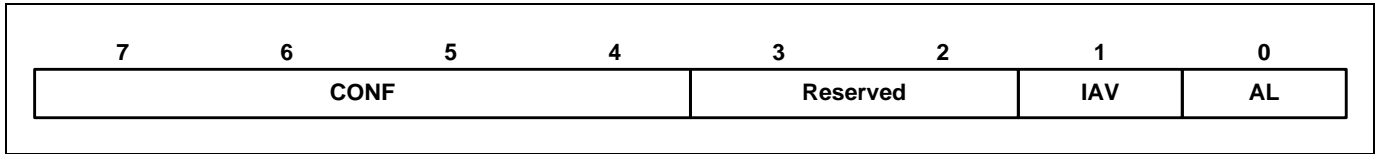
ZWIR4512AC1 Pin Number	ZWIR4512AC2 Pin Number	Interface	Function	Default Configuration
1	1	SPI	Master Out Slave In (SPI_MOSI)	Input
2	2	SPI	Master In Slave Out (SPI_MISO)	Push-pull output with high driving strength
3	3	SPI	Clock (SPI_SCK)	Input
4	4	SPI	Slave Select (SPI_NSS)	Input
5	5	UART 2	Receive (UART2_RX)	Input
6	6	UART 2	Transmit (UART2_TX)	Push-pull output with medium driving strength
7	7	SPI	Data Pending	Push-pull output with medium driving strength
12	13	UART 1	Receive (UART1_RX)	Input
13	14	UART 1	Transmit (UART1_TX)	Push-pull output with medium driving strength

The I/O pin configuration is changed with different commands. The following commands could affect I/O pin settings: Configure UART1, Configure UART2, Configure SPI, Configure Wakeup, Configure LEDs, and Configure GPIO. These commands are only executed successfully if changes to the I/O configuration can be performed without possibly changing the behavior of the application. For instance, it is not possible to set pin 4 operating mode to GPIO while the SPI is configured to be active. Configure GPIO will report an I/O conflict in this case. This is also applicable for changes in the reverse direction. If pin 4 is configured as GPIO, it will not be possible to enable the SPI interface. The only possibility for dual use of an I/O pin is configuring an already configured input as a wakeup source.



LED driver and GPIO pins permit the configuration of the pin settings. The SCI firmware uses an 8-bit field describing the pin configuration. This 8-bit field is passed to the corresponding configuration function. The layout of the bit-field is shown in Figure 2.8.

**Figure 2.8 Layout of the Pin Configuration 8-Bit Field**



The **AL** bit determines whether the pin is active HIGH or LOW.

- 0 The pin is active LOW
- 1 The pin is active HIGH

The **IAV** bit determines the initial activity value of the pin. This value only applies to outputs.

The logical level at the pin output depends on the setting of AL.

- 0 The initial value is inactive
- 1 The initial value is active

**CONF** determines the pin configuration.

- 0 Disabled
- 1 Floating input
- 2 Pull-down input
- 3 Pull-up input
- 4 Push/pull output: medium driving strength
- 5 Open drain output: medium driving strength
- 6 Push/pull output: low driving strength
- 7 Open drain output: low driving strength
- 8 Push/pull output: high driving strength
- 9 Open drain output: high driving strength

### 2.13. Watchdog Timer

The serial command interface firmware allows enabling the hardware watchdog timer of the module's micro-controller. This watchdog timer automatically restarts the module upon detection of a software deadlock. A deadlock is detected if the firmware does not execute its internal main loop for at least five seconds.

Using the watchdog timer in conjunction with the module's low-power modes is only possible if the low power interval is significantly shorter than 5 seconds. If one of the low-power modes is required for a longer timer, the watchdog must not be used.

By default, the watchdog timer is disabled after startup and any type of system reset. In order to enable the watchdog timer in any situation, the host device should enable the watchdog timer and store the configuration to the device's internal flash memory. After this configuration, the watchdog is always enabled after system or power-on reset. In this case, the only method of disabling the watchdog is setting the watchdog-enable bit to zero and storing the configuration again (see section 3.34) or by execution of the Restore Factory Settings command.

## 2.14. Persistent Parameter Storage

The SCI allows storing the configuration of the device at any time. Stored parameters are restored automatically after the following events:

- Power-on
- Wakeup from Standby Mode
- System reset, triggered by activation of reset-pin, execution of command Reset, or software reset caused by an internal error

If the Store Configuration command is called (see section 3.17), all configuration parameters that are in effect at the time are stored in the module's internal flash memory. The following parameters are stored by the persistent parameter storage mechanism:

- All parameters set with the Configure PHY command
- All parameters set with the Configure NET command
- SPI, UART1 and UART2 configuration
- GPIO configuration
- Wakeup configuration
- LED configuration
- PAN ID and PAN address configuration
- Receiver settings (all connections that have been opened for listening)
- All configured security policies
- All configured security associations
- All configured IKEv2 authentication entries
- Full multicast configuration including the group-mask
- Watchdog configuration

The status of the neighbor cache and the routing tables of the device is not stored or restored, respectively.

### 3 Command Reference

This section describes the available commands for SCI modules. Table 3.1 gives an overview of available commands. Detailed descriptions are given in the following sections.

**Table 3.1 Command Overview**

Name	Code	Description	Section
<b>Command Frames Accepted by the SCI Node</b>			
Transmit Frame	0x01	Sends a data packet to a single or multiple remote locations.	3.1
Configure Receiver	0x02	Configures the reception of data and where the data should be sent.	3.2
Get TRX Statistics	0x03	Returns information about transmitted and received data and about transmission duty cycle.	3.3
Reset TRX Statistics	0x04	Resets all TRX statistics.	3.4
Configure Wake-up	0x05	Configures the wakeup sources for the different power modes.	3.5
Power-Down	0x06	Puts the device into a low-power mode.	3.6
Get Address Configuration	0x07	Requests the MAC address and the IPv6 addresses of the device.	3.7
Set PAN Address	0x08	Sets the PAN address of the device.	3.8
Set PAN ID	0x09	Sets the PAN ID of the PAN in which the device is operating.	3.9
Configure PHY	0x0A	Configures PHY parameters (channel, modulation, TX-power).	3.10
Configure NET	0x0B	Configures network parameters.	3.11
Discover Network	0x0C	Detects other devices in the network.	3.12
Remote Execute	0x0D	Executes command on a remote device.	3.13
Get Firmware Version	0x0E	Requests the firmware version running on the module.	3.14
Reset	0x0F	Resets the module.	3.15
Network Reset	0x10	Resets the network stack on the module.	3.16
Store Configuration	0x11	Stores the current configuration of the module to be restored during startup.	3.17
Test	0x12	Requests immediate response (useful for "alive" checking).	3.18
Configure LEDs	0x13	Configures the ports at which indicator LEDs for indicating transmission and reception of data are connected.	3.19
Configure GPIO	0x14	Configures a module pin as a GPIO pin.	3.20
Write GPIO	0x15	Sets a value on a GPIO pin.	3.21
Read GPIO	0x16	Reads the value of a GPIO pin.	3.22
Toggle GPIO	0x17	Toggles a GPIO pin.	3.23
Configure UART1	0x18	Configures parameters of UART interface 1.	3.24
Configure UART2	0x19	Configures parameters of UART interface 2.	3.25

Name	Code	Description	Section
<b>Command Frames Accepted by the SCI Node</b>			
Configure SPI	0x1A	Configures parameters of the SPI interface.	3.26
Restore Factory Settings	0x1C	Restores all module default settings.	3.27
Configure Multicast	0x1D	Configures multicast settings.	3.28
Join Multicast Group	0x1E	Joins a multicast group.	3.29
Leave Multicast Group	0x1F	Leaves a multicast group.	3.30
Add Security Policy	0x20	Adds a new IPSec security policy.	3.31
Add Security Association	0x21	Adds a new IPSec security association.	3.32
Add IKEv2 Authentication Entry	0x22	Registers the IKE daemon for automatic generation of missing security associations.	3.33
Enable Watchdog Timer	0x23	Enables or disables the watchdog timer.	3.34
Get Current Interface	0x24	Requests the interface identifier of an interface.	0
Get PHY Configuration	0x26	Requests the PHY parameters (channel, modulation, TX-power).	3.36
Get NET Configuration	0x27	Requests the Network parameters.	3.37
Get FCC ID	0x28	Request the FCC ID of the module.	3.38
Set Acknowledge Enable	0x29	Enable or disable transmission of acknowledgements.	3.39
Set Parameter	0x30	Set network parameter.	3.40
Get Parameter	0x31	Get network parameter.	3.41
<b>Command Frames Sent by the SCI Node</b>			
Receive Packet	0x50	Sent to the host upon reception of a data packet.	3.38
Remote Response	0x51	Send to a host to report the response of a remotely executed command.	3.43
Reset Report	0x52	Sent to the host after module reset.	3.44
Network Configured	0x53	Sent to the host after completion of network initialization.	3.45

### 3.1. Transmit Frame Command

This command is used to transmit data frames to one or multiple destinations. Data transmission is based on the User Datagram Protocol (UDP) over Internet Protocol version 6 (IPv6). The destination IP address determines which node(s) will receive the packet.

**Table 3.2 Transmit Frame Command Fields**

Field	Width (bytes)	Description
Destination IP Address	16	This is the address of the destination node of the packet. Address can be of any type available in IPv6. Depending on the address type, one or multiple destinations are selected as the receiver of the packet. Refer to section 2.5.3 for further information on IPv6 addressing. The address is transmitted in big-endian format. Thus, the address ff02::1 would be transmitted as <b>ff 02 00 00 00 00 00 00 00 00 00 00 00 00 00 01</b> .
Destination UDP Port	2	This is the port number that the target device should be listening on. Use the Configure Receiver Command to configure the listening port on ZWIR45xx SCI nodes (see section 3.2). The port field is transmitted in little-endian format. Thus, port 1000 <sub>DEC</sub> (i.e., 0x3e8) would be transmitted as <b>e8 03</b> .
Payload Data	1 – 1232	This is the payload data to be transmitted. Packets must be in the documented range. Otherwise an error message will be returned and the packet is dropped at the sender.

**Table 3.3 Transmit Frame Command Responses**

Code	Description
0	Success – this response is only sent when acknowledges are enabled.
1	The packet that should be sent is too big (the frame payload is larger than 1250 bytes).
2	The packet that should be sent is too small (the frame payload is smaller than 19 bytes).
27	The packet could not be sent. Reasons for this error are typically temporary, so try to send the packet again. <b>Note:</b> If this response is not received, it does not automatically mean that the packet was delivered successfully to the receiver!

### 3.2. Configure Receiver Command

This command enables data reception on a device. It configures the device that data should be received from and the UDP port that is used for this communication. Furthermore, it allows selecting the communication interface to which incoming data should be sent.

**Table 3.4 Configure Receiver Command Fields**

Field	Width	Description
Source IPv6 Address	16	The address from which packets should be received. If the IPv6 unspecified address [::] is selected, packets from all nodes are accepted. Specifying a multicast address is not allowed. The address is transmitted in big-endian format. Thus, the address fe80::1 would be transmitted as fe 80 00 00 00 00 00 00 00 00 00 00 00 00 00 01.
Local UDP Port	2	This is the port on which data should be received. The UDP port allows distinguishing different connections. The port field is transmitted in little-endian format. Thus, port 1000 <sub>DEC</sub> (i.e., 0x3e8) would be transmitted as e8 03.
Dispatch Interface	1	This option specifies the interface to which received data will be dispatched.
RSSI Enable	1	If non-zero, this byte determines whether a RSSI value is appended to the data in each Receive Packet command (see section 3.42). This byte is optional and assumed to be zero if omitted.

**Table 3.5 Configure Receiver Command Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	The request was too short.
2	-	-	The request was too long.
5	-	-	The request was dropped because an internal error occurred. Try to perform a full system reset to fix the problem.
20	IID	1	A connection with the same configuration already existed with another dispatch interface. The previous entry was overwritten with the new one. The response contains an option field that specifies the former IID that has been replaced with the new value.
21	-	-	The request was dropped as the maximum number of connections is already in use. The number of connections can be increased using the Configure NET command (see section 3.11).

### 3.3. Get TRX Statistics

This command requests a response frame containing the current network statistics. Network statistics consist of the number of transmitted and received packets, the number of transmitted and received bytes, the number of failing transmissions, and the transmission duty cycle.

Note that the transmission duty cycle in European countries is limited in the different frequency bands. It is up to the application to ensure that these limitations are met. This command helps to monitor the transmission duty cycle of the device. Present duty cycle limitations are listed in Table 3.7.

The Get TRX Statistics command does not expect any additional arguments in the command frame. If data is transmitted in the payload section of the command frame, this data is ignored. There are no error responses sent by this command.

Data returned is always counted from the last device reset, network reset, reset of the TRX statistics, or from the last wakeup from Standby Mode. The values returned also consider traffic that is forwarded automatically to other nodes and is not visible to the application.

**Table 3.6** *Get TRX Statistics Response Format*

Code	Field	Length	Comment
0	TX Bytes	4	Number of bytes transmitted.
	TX Packets	4	Number of packets sent.
	RX Bytes	4	Number of bytes received.
	RX Packets	4	Number of packets received.
	TX Fail	4	Number of lost packets due to transmission failures; e.g., due to an occupied channel.
	Duty Cycle	4	The transmission duty-cycle in percent, multiplied with a factor of 1000; e.g., a value of 1000 corresponds to a duty-cycle of 1%.

**Table 3.7** *Transmission Duty-Cycle Requirements of Sub-GHz Devices in the European Union*

Frequency Band	Corresponding Channels	Required Duty-Cycle
868.0 – 868.6 MHz	0	1%
863.0 – 870.0 MHz	100, 101, 102	0.1%

### 3.4. Reset TRX Statistics

This command resets the transmission and reception statistics. All values are set to 0. The command does not expect any fields. Data received in the payload section of the command frame is ignored. There are no responses sent by this command.

**Table 3.8** *Reset TRX Statistics Responses*

Code	Description
0	Success – this response is only sent when acknowledges are enabled.

### 3.5. Configure Wake-up Command

This command configures how the device is awakened from low-power modes. The settings applied here are used for each power mode. However, for all low-power modes, only a limited set of wakeup sources is accepted. Table 3.9 lists the possible wakeup sources for the different low-power modes.

For wakeup from Sleep Mode, any of the wakeup options except pin 8 are valid. Thus, it is possible to wake up from activity on one of the serial interfaces, on any but one GPIO pin, on reception of a packet, or on timeout of the Sleep Timer. It is also possible to use a combination of these wakeup sources.

From Stop Mode, it is possible to wake up on any except one GPIO pin (see Table 3.9), on reception of a packet, or if the Sleep Timer timed out.

In Standby Mode only the Sleep Timer or one GPIO pin is available as a wakeup source (see Table 3.9).

**Table 3.9** *Wakeup Sources*

Wakeup Source		Mode			Comment
ZWIR4512AC1	ZWIR4512AC2	Sleep	Stop	Standby	
UART1		X	-	-	An incoming data transfer wakes the device.
UART2		X	-	-	An incoming data transfer wakes the device.
SPI		X	-	-	An incoming data transfer wakes the device.
Sleep Timer		X	X	X	The device sleeps for the time that is specified in the Power-Down command (see section 3.6).
Transceiver		X	X	-	The device is awakened when the transceiver receives a packet.
Pin 1		X	X	-	Not available in combination with pin 23 wakeup on ZWIR4512AC1 or pin 25 wakeup on ZWIR4512AC2.
Pin 2		X	X	-	Only usable if SPI interface is disabled. Not available in combination with pin 24 wakeup on ZWIR4512AC1 or pin 26 wakeup on ZWIR4512AC2.
Pin 3		X	X	-	
Pin 4		X	X	-	1ms before starting the SPI clock, pull down pin 4 to wake up from Stop Mode using the SPI.
Pin 5		X	X	-	Not available in combination with pin 19 wakeup on ZWIR4512AC1 or pin 21 wakeup on ZWIR4512AC2.
Pin 6		X	X	-	Only usable if UART2 interface is disabled.
Pin 7		X	X	-	Only usable if UART2 interface is disabled or if UART2 is used without flow control.



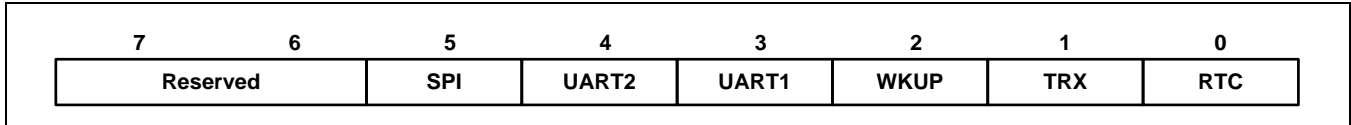
Wakeup Source		Mode			Comment
ZWIR4512AC1	ZWIR4512AC2	Sleep	Stop	Standby	
Pin 8		-	-	X	Selecting this pin as a wakeup source is accomplished by setting the WKUP bit in the peripheral field. Only rising edges can be detected!
Pin 9		X	X	-	Not available in combination with pin 20 wakeup on ZWIR4512AC1 or pin 22 wakeup on ZWIR4512AC2.
Pin 12	Pin 13	X	X	-	
Pin 13	Pin 14	X	X	-	Only usable if UART1 interface is disabled.
n/a	Pin 17	X	X	-	Not available in combination with pin 24 wakeup on ZWIR4512AC2.
n/a	Pin 18	X	X	-	Not available in combination with pin 23. wakeup on ZWIR4512AC2.
Pin 16	Pin 19	X	X	-	
Pin 17	Pin 20	X	X	-	Only usable if UART1 interface is disabled or if UART1 is used without flow-control.
Pin 19	Pin 21	X	X	-	Not available in combination with pin 5 wakeup.
Pin 20	Pin 22	X	X	-	Not available in combination with pin 9 wakeup.
Pin 21	Pin 23	X	X	-	Not available in combination with pin 18 wakeup on ZWIR4512AC2.
Pin 22	Pin 24	X	X	-	Not available in combination with pin 17 wakeup on ZWIR4512AC2.
Pin 23	Pin 25	X	X	-	Not available in combination with pin 1 wakeup.
Pin 24	Pin 26	X	X	-	Not available in combination with pin 2 wakeup.

**Table 3.10 Configure Wakeup Command Fields**

Field	Width	Description
Peripheral	1	Bit-mask controlling which peripheral is used to wake the device.
GPIO Falling Edge <sup>(1)</sup>	4	Bit-mask that selects the falling edge on a GPIO as a wakeup event.
GPIO Rising Edge <sup>(1)</sup>	4	Bit-mask that selects the rising edge on a GPIO as a wakeup event.

*(1) Note: In order to maintain compatibility with older SCI firmware versions, the command also accepts 3-byte bit-masks. Pin numbers that cannot be encoded in the 3-byte bit-mask cannot be used as a wakeup source.*

**Figure 3.1 Configure Wakeup - Peripheral Field Format**



**RTC** determines if the wakeup timer is used as a wakeup source:

- 0 Do not use the wakeup timer.
- 1 Wakeup on wakeup timer timeout.

**TRX** determines if the transceiver is used as a wakeup source:

- 0 Do not use the transceiver (transceiver is in Standby Mode).
- 1 Use the transceiver (transceiver remains in Receive Mode).

**WKUP** determines if the wakeup pin (pin 8) is used as a wakeup source:

- 0 Do not use the wakeup pin.
- 1 Wakeup on rising edge on wakeup pin.

**UART1** determines if the device is awakened on data reception on the UART1 interface:

- 0 Do not use UART1 as wakeup source.
- 1 Wakeup on reception of data on UART1.

**UART2** determines if the device is awakened on data reception on the UART2 interface:

- 0 Do not use UART2 as wakeup source.
- 1 Wakeup on reception of data on UART2.

**SPI** determines if the device is awakened on reception of data on the SPI interface:

- 0 Do not use SPI as wakeup source.
- 1 Wakeup on reception of data on SPI.

**Table 3.11 Configure Wakeup Command Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	The command frame is too short.
2	-	-	The command frame is too long.
8	Pin Number	1	The pin specified in the response field is not a GPIO.
9	Pin Number	1	The pin specified in the response field is already configured as an output pin, and therefore cannot be used as a wakeup pin.
12	Pin Number	1	The pin specified conflicts with another GPIO pin selected as a wakeup source.

### 3.6. Power-Down Command

This command sends the device into one of the low-power modes. The power mode is specified in the first option field. The duration of the power-down state is specified in the second field. The Configure Wakeup command (see section 3.5) specifies on which events the low-power mode is exited. The duration field has no effect if the sleep timer is not selected as a wakeup source.

For the low-power mode, three values are available:

- 0 – Sleep Mode
- 2 – Stop Mode
- 4 – Standby Mode

**Table 3.12 Power-Down Command Fields**

Field	Length	Description
Mode	1	This field specifies the low-power mode to be entered.
Duration	4	This field specifies for how many seconds the device remains in the low power mode. This field is ignored if the sleep timer is not selected as a wakeup source.

**Table 3.13 Power-Down Command Responses**

Code	Field	Length	Description
1	-	-	The command frame is too short.
2	-	-	The command frame is too long.
3	-	-	Invalid low-power mode.

### 3.7. Get Address Configuration Command

This command requests the current address configuration of the device. The EUI64 address and all IPv6 addresses of the device are returned in the response. This command does not require additional fields. Extra fields in the command frame are ignored.

The number of IPv6 addresses ( $N_{IPv6}$ ) returned by the command must be computed from the length field of the response frame. This is done using the following formula:

$$N_{IPv6} = \frac{Length - 9}{16}$$

**Table 3.14 Get Address Configuration Response Fields**

Code	Field	Length	Description
0	EUI64	8	The EUI64 address of the device. If not changed manually, this function will return the factory programmed, globally unique EUI64. The address is returned in big-endian format.
	IPv6 [1]	16	The first IPv6 address of the device. The address is returned in big-endian format. If this field is not present, the device is unable to communicate.
	IPv6 [2]	16	The second IPv6 address of the device (optional). The address is returned in big-endian format.
	...		
	IPv6 [N]	16	IPv6 address N (optional). The address is returned in big-endian format.

### 3.8. Set PAN Address

This command sets the PAN address of the device. The PAN address must be unique in the network. Changing this address is not recommended as the module has a unique preprogrammed PAN address. The addresses 0 and 0xFFFFFFFFFFFFFFFF must not be used as PAN addresses.

Performing a network reset is required after changing the PAN address. For this purpose, the Set PAN Address command has one optional field that allows performing a network reset immediately after changing the address. If the optional field is set to a value other than 0, the network reset is performed. If the field is set to 0 or not included in the message, the network reset is not performed.

**Table 3.15 Set PAN Address Command Fields**

Field	Width	Description
PAN Address	8	The new PAN address to be assigned to the device.
Reset [Optional]	1	Performs a network reset immediately after changing the PAN address if this field is set to a value other than 0. If set to 0 or omitted, no reset is performed.

**Table 3.16 Set PAN Address Command Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	The message is too short.
2	-	-	The message is too long.
19	-	-	The PAN address is invalid.

### 3.9. Set PAN ID

This command sets the PAN identifier of the node. The PAN identifier is a 16-bit number that is used to differentiate networks. Nodes that wish to communicate must use the same PAN ID. Otherwise they will not be able to communicate even if they can physically reach each other. The PAN identifier is set immediately after calling this command. No network reset is required. The command accepts only a single argument field. This field specifies the new PAN ID. The PAN identifier *0xFFFF* must not be used!

Note: If the PAN ID is set after some packets have already been sent, it is possible that packets that have been queued using the old PAN ID setting might be sent to the network with the new PAN ID.

**Table 3.17 Set PAN ID Command Fields**

Field	Width	Description
PAN ID	2	The new PAN identifier to be set.

**Table 3.18 Set PAN ID Command Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	The message is too short.
2	-	-	The message is too long.
3	-	-	The PAN ID is invalid.

### 3.10. Configure PHY

Using this command it is possible to change the physical communication channel, the modulation scheme, and the output power of the module. For successful communication between two nodes, both nodes must have the same channel and the same modulation scheme selected. All parameters are configured using one command.

Table 3.20 shows the command format. Table 3.21 lists the possible error responses.

If one of the fields contains an invalid value, the whole command is not executed. Invalid parameters are reported by a command response. Note that for different channels the output power is limited.

**Table 3.19 List of Available Radio Channels**

EU Channels		US Channels			
Channel	Frequency	Channel	Frequency	Channel	Frequency
0	868.3MHz	1	906.0MHz	6	916.0MHz
100	865.3MHz	2	908.0MHz	7	918.0MHz
101	866.3MHz	3	910.0MHz	8	920.0MHz
102	867.3MHz	4	912.0MHz	9	922.0MHz
		5	914.0MHz	10	924.0MHz

**Table 3.20 Configure PHY Command Fields**

Field	Width	Description
Channel	1	<p>This is the channel number to be selected. Possible values include</p> <ul style="list-style-type: none"> <li>▪ Europe 0 and 100 to 102</li> <li>▪ North America 1 to 10</li> </ul> <p>Note that regulations do not allow selecting a channel that does not correspond to the location of the installation of the application. Default value: 0</p>
Modulation	1	<p>This is the modulation scheme to be used. BPSK modulation is selected using the value 0; any other value will result in selecting QPSK. Default value: 0</p>
TX-Power	1	<p>This value determines the output power of the device. Depending on the channel being selected, the maximum output power can vary from 0 to 10 dBm. All US channels (0 &lt; channel &lt; 11) allow 10dBm to be selected. For the European channels, a maximum of 5dBm is allowed. Default value: 0</p>

**Table 3.21 Configure PHY Command Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	The message is too short.
2	-	-	The message is too long.
24	-	-	Invalid channel.
25	-	-	Invalid power.

### 3.11. Configure NET

This command is used to configure the network parameters of the device. All parameters are set at the same time. The execution of this command automatically performs a network reset as this is required for the parameters to take effect.

**Table 3.22 Configure NET Command Fields**

Nr.	Field	Width	Description
0	Max Socket Count	1	This value determines how many RX sockets may be open at the same time. Default value: 4
1	Neighbor Cache Size	1	This value determines the number of the neighbor cache entries. The neighbor cache must have at least one entry. Default value: 4
2	Neighbor Reachable Time	2	This value determines how long a neighbor is considered to be reachable before reachability detection is performed by the network stack. Default value: 3600s
3	Max Hop Count	1	This number determines how many hops a packet is allowed to take through the network. Setting this value to zero disables mesh networking. Default value: 8
4	Routing Table Size	2	This number determines the maximum number of entries in the routing table. The minimum Routing Table Size is 1. Default value: 8
5	Route Timeout	2	This number determines how long a routing table entry stays valid. If a route is not validated by network traffic for this amount of time, the route entry is removed from the routing table. Default value: 3600s
6	Route Max Fail Count	1	This number determines how many times routing a packet over an existing route is allowed to fail before the route is considered as broken and the entry is removed from the routing table. Default value: 3
7	Route Min RSSI	1	During route discovery, this number determines the minimum RSSI that must be achieved during route discovery. If one link on the route has an RSSI below this value, the route is not established. Default value: -128
8	Route RSSI Reduction	1	This value determines how much the Route Min RSSI value is decreased in the event of a failing route setup due to RSSI constraints. Default value: 0

Nr.	Field	Width	Description
9	Route Request Attempts	1	This number determines how many attempts are made to establish a route before the request is reported to be failing. Default value: 3
10	Do Duplicate Address Detection	1	This Boolean value determines if the device performs Duplicate Address Detection when being powered on or awakened from Standby Mode. Default value: true
11	Do Router Solicitation	1	This Boolean value determines if the device performs Router Solicitations when being powered on or awakened from Standby Mode. Default value: true

**Table 3.23 Response Codes of the Configure NET Command**

Code	Field	Length	Description
0	-		Success – this response is only sent when acknowledges are enabled.
1	-		The message is too short.
2	-		The message is too long.
26	-		Socket count too high. The maximum number of sockets that can be used is 251.
29	Parameter Mask	2	At least one requested parameter is out of range. Each enabled bit at the returned parameter mask represents the adjacent failed parameter. Note that all invalid parameters were ignored.

### 3.12. Discover Network

This command initiates a network discovery. Network discovery is used to identify other ZWIR45xx modules in the network. Each device that receives a network discovery request responds with a message containing information about the IPv6 addresses of the device, the hop distance of the remote device, and a link indicator. Refer to section 2.8 for more detailed information about network discovery.

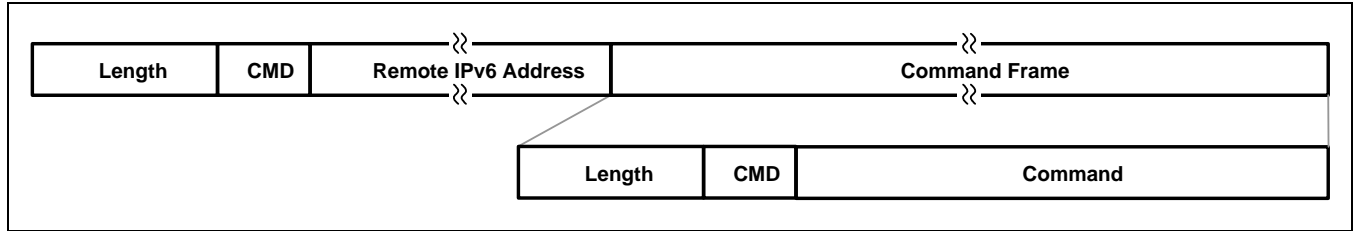
The only argument that is passed to the Discover Network command is the upper time limit by which nodes must respond to the network discovery request. If this parameter is omitted, a default value of 3 is assumed for the time limit. If 0 is passed as the time limit, the default value of 3 is used as well.



### 3.13. Remote Execute

The remote execute command allows executing a command on a remote device. The first argument is the IPv6 device of the remote node that a command should be executed on. The second argument is the command frame that must be executed on the remote device. Figure 3.2 shows the general layout of a Remote Execute command frame. The whole command frame that is requested to be executed remotely must not be longer than 1232 bytes.

**Figure 3.2 Remote Execute Command Frame Layout**



**Table 3.24 Remote Execute Response Codes**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	The message is too short.
2	-	-	The message is too long.
27	-	-	Failed to send out the remote command over the network interface. Note: If this response is not received, it does not automatically mean that the command was successfully delivered to the remote host.

If the command executed remotely generates a response message, this message is delivered through the Remote Response command (see section 3.43).

### 3.14. Get Firmware Version

This command requests the SCI to send a firmware Version String to the host. The command does not require any argument fields. Any argument field present in the command frame is ignored.

**Table 3.25 Get Firmware Version Command Responses**

Field	Width	Description
Status	1	This field should always contain a value of 0. If a non-zero value is returned in this field, the response must be ignored.
Product ID	2	This is the Product ID of the IDT Product.
SCI Major Version	1	This is the major version of the SCI.
SCI Minor Version	1	This is the minor version of the SCI.
SCI Version Extension	2	This is the version extension of the SCI.
Stack Major Version	1	This is the major stack version.
Stack Minor Version	1	This is the minor stack version.
Stack Version Extension	2	This is the version extension.

### 3.15. Reset

This command performs a System Reset. After the System Reset, the stack loads the configuration parameters stored in the flash memory and configures the device accordingly.

This command does not accept any parameters. If extra fields are present in the command frame, the device returns error code 2. This command is never acknowledged. However, after reset execution, the SCI module sends the unsolicited Reset Report command with a response code of 3 (see section 3.44), followed by the Network Configured command (see section 3.45).

**Table 3.26 Reset Responses**

Code	Field	Length	Description
2	-	-	The message is too long.

### 3.16. Network Reset

This command performs a Network Reset. This operation disconnects all network sockets and reinitializes the network stack with the configured parameters. The device will send out Router Solicitation and Duplicate Address Detection messages if enabled.

The command does not accept any parameters. The module sends the unsolicited Network Configured command when re-initialization of the network stack is complete.

**Table 3.27 Network Reset Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
2	-	-	The message is too long.

### 3.17. Store Configuration

This command stores the current device configuration to the device’s flash memory. During the next startup phase, the stack is configured using the stored parameters. Refer to section 2.14 for a list of parameters that can be stored persistently in the device’s flash memory.

In order to prevent accidental storage of the current settings, transmitting a four-byte validation string is required. If the string is incorrect, nothing is stored and an error message is returned.

**Table 3.28 Store Configuration Command Fields**

Field	Width	Description
Validation String	4	This field must contain the number <i>0x681c51ab</i> . The number must be transmitted in little-endian byte order (least significant byte first).

**Table 3.29 Store Configuration Command Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	The message is too short.
2	-	-	The message is too long.
7	-	-	Invalid validation string.
27	-	-	Failed to write parameters to the flash.

### 3.18. Test

This command causes the device to reply with the same packet immediately (with the command field set to Test response). This is typically used for “alive” testing.

**Table 3.30 Test Command Fields**

Field	Width	Description
Data	0 – 1250	This field contains arbitrary data.

**Table 3.31 Test Command Responses**

Field	Width	Description
Data	0 – 1250	This field contains exactly the same data that have been sent to the device.

### 3.19. Configure LEDs Command

It is possible to configure several indicator LEDs using this command. There are indicator LEDs available for incoming and outgoing network data, incoming and outgoing serial data, and the power state. Each LED output can be configured separately using the I/O configuration byte described in section 2.12. The IAV bit in the I/O configuration byte is ignored. LED pins must be configured as output. The selected I/O pin must be unconfigured.

**Table 3.32 Configure LEDs Command Fields**

Field	Width	Description
Network RX LED Pin	1	Pin number at which the network RX LED is attached. Set to 0 if network RX indication is disabled.
Network RX LED Config <sup>1)</sup>	1	The I/O configuration of the network RX LED pin.
Network TX LED Pin	1	Pin number at which the network TX LED is attached. Set to 0 if network TX indication is disabled.
Network TX LED Config <sup>1)</sup>	1	The I/O configuration of the network TX LED pin.
Serial RX LED Pin	1	Pin number at which the Serial RX LED is attached. Set to 0 if network RX indication is disabled.
Serial RX LED Config <sup>1)</sup>	1	The I/O configuration of the Serial RX LED pin.
Serial TX LED Pin	1	Pin number at which the Serial TX LED is attached. Set to 0 if network TX indication is disabled.
Serial TX LED Config <sup>1)</sup>	1	The I/O configuration of the Serial TX LED pin.
Active LED Pin	1	Pin number at which the Active LED is attached. Set to 0 if active LED is disabled.
Active LED Config <sup>1)</sup>	1	The I/O configuration of the Active LED pin.
1) Refer to section 2.12 for a description of the configuration byte. The IAV bit in the configuration byte is ignored.		

**Table 3.33 Configure LEDs Command Responses**

Code	Field	Length	Description
0	-		Success – this response is only sent when acknowledges are enabled.
1	-		Message is too short.
2	-		Message is too long.
8	Pin Number	1	The pin specified in the response field cannot be used as an LED driver.
9	Pin Number	1	The pin specified in the response field is already used for another purpose.
18	-		Invalid GPIO configuration.

### 3.20. Configure GPIO

This command configures an I/O pin to be used as application programmable GPIO. The pin to be configured must be in the unused operating mode before configuration can be successful. Trying to configure an already configured pin will cause the stack to send an error message and leave the pin configuration in the previous state. Refer to section 2.12 for a detailed description of available GPIO configurations.

**Table 3.34 Configure GPIO Command Fields**

Field	Width	Description
Pin Nr.	1	The pin number of the GPIO to be configured.
Configuration	1	The intended GPIO configuration (refer to Table 2.8).

**Table 3.35 Configure GPIO Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	Message is too short.
2	-	-	Message is too long.
8	Pin Number	1	The pin specified in the response field cannot be used as a GPIO.
9	Pin Number	1	The pin specified in the response field is already used for another purpose.
18	-	-	Invalid GPIO Configuration.

### 3.21. Write GPIO

The output value of a GPIO pin is set using this command. This command can only be executed on pins that are outputs in GPIO operating mode.

**Table 3.36 Write GPIO Command Fields**

Field	Width	Description
Pin Nr.	1	The pin number of the GPIO to be written.
Configuration	1	The intended GPIO value (0 or 1).

**Table 3.37 Write GPIO Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	Message is too short.
2	-	-	Message is too long.
8	Pin Number	1	The pin specified in the response is not a GPIO pin.
22	Pin Number	1	The pin specified in the response is not configured as GPIO output.

### 3.22. Read GPIO

This command reads a value from a GPIO port. The command can be executed on any GPIO pin regardless of whether it is configured as input or output. Also interface, wakeup, and LED pins can be read.

**Table 3.38 Read GPIO Command Fields**

Field	Width	Description
Pin Number	1	The pin number of the GPIO to be read.

**Table 3.39 Read GPIO Command Responses**

Code	Field	Length	Description
0	Value	1	If the command is successful, the response code is 0 and the value is returned in the first field of the response.
1	-	-	Message is too short.
2	-	-	Message is too long.
8	Pin Number	1	The pin specified in the response is not a GPIO pin.
23	Pin Number	1	The pin specified in the response is unconfigured.

### 3.23. Toggle GPIO

This command toggles the output value of a GPIO pin. This command can only be executed on pins that are outputs in GPIO operating mode.

**Table 3.40 Toggle GPIO Command Fields**

Field	Width	Description
Pin Nr.	1	The pin number of the GPIO to be toggled.

**Table 3.41 Toggle GPIO Command Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	Message is too short
2	-	-	Message is too long.
8	Pin Number	1	The pin specified in the response is not a GPIO pin.
22	Pin Number	1	The pin specified in the response is not configured as GPIO output.

### 3.24. Configure UART1

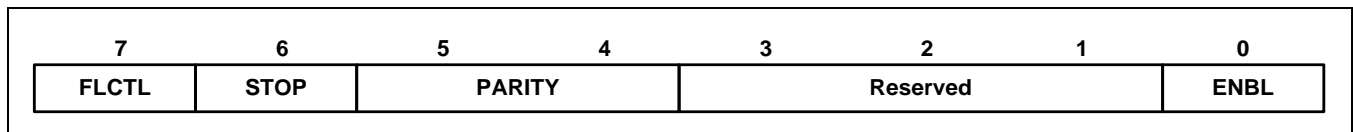
This command configures the settings of UART1. The first argument field contains a configuration bit-field controlling the parity bit, stop bit, and flow-control settings as well as the enable status of the UART. The second field is a three-byte number controlling the baud-rate settings.

The layout of the configuration field is shown in Figure 3.3. If the ENBL field of the configuration bit-field is set to zero, the baud-rate field can be omitted.

**Table 3.42 Configure UART Command Fields**

Field	Width	Description
Configuration	1	This byte controls the behavior of the module.
Baud Rate	3	Baud-rate in bits per second. This field can be omitted if the ENBL bit in the configuration field is set to zero. The number is required to be in little-endian format (least significant byte first).

**Figure 3.3 UART Configuration Bit-Field Layout**



The **ENBL** bit selects whether the interface is enabled or disabled.

- 0 The interface is disabled.
- 1 The interface is enabled.

The **PARITY** bits control transmission of parity bits with UART frames.

- 0 Parity is disabled.
- 2 Even parity.
- 3 Odd Parity.

The **STOP** bit controls the transmission of stop bits.

- 0 One stop bit.
- 1 Two stop bits.

The **FLCTL** bit controls whether flow control is enabled or disabled.

- 0 Flow control is disabled.
- 1 Flow control is enabled.

**Table 3.43 Configure UART Command Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	Message is too short.
2	-	-	Message is too long.
9	Pin Number	1	Pin is already configured.

### 3.25. Configure UART2

This command is used to configure the UART2 interface. This command functions the same as the Configure UART1 command (see section 3.24).

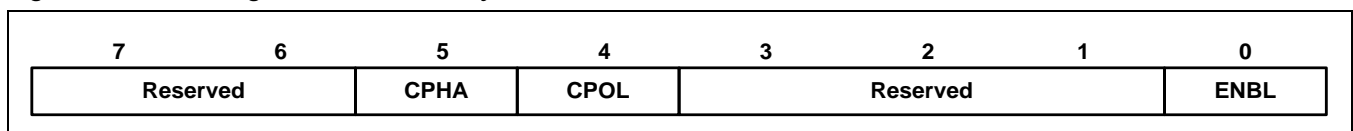
### 3.26. Configure SPI

This command is used to configure the SPI interface. The first argument field is a configuration bit-field that controls the clock polarity and clock edge of the SPI as well as the enable status. The second field specifies which pin is used as the pending data indicator pin. The pending pin field can be omitted if the ENBL field of the configuration bit-field is set to zero.

**Table 3.44 Configure SPI Command Fields**

Field	Width	Description
Configuration	1	Bit-field configuring SPI behavior and enable status. Refer to Figure 3.4 for the layout of the bit-field.
Pending Data Pin	1	Specifies which GPIO pin is used as the pending data indicator. This field can be omitted if the ENBL bit in the configuration bit-field is set to zero.

**Figure 3.4 SPI Configuration Bit-Field Layout**



The **ENBL** bit selects whether the interface is enabled or disabled.

- 0 The interface is disabled.
- 1 The interface is enabled.

The **CPOL** bit controls the clock polarity.

- 0 Clock input is low when idle.
- 1 Clock input is high when idle.

The **CPHA** bit controls the clock phase

- 0 Data are sampled on the first clock edge.
- 1 Data are sampled on the second clock edge.

**Table 3.45 Configure SPI Command Responses**

Code	Field	Length	Description
0	-		Success – this response is only sent when acknowledges are enabled.
1	-		Message too short.
2	-		Message too long.
8	Pin Number	1	Pin is not a GPIO.
9	Pin Number	1	Pin is already configured.



### 3.27. Restore Factory Settings

This command deletes all settings stored in the flash memory and restores the original settings instead. A system reset is executed after restoring the default settings. To avoid accidental execution of the command, passing a four-byte validation string is required.

This command is never acknowledged. However, after reset execution, the SCI module sends the unsolicited Reset Report command with a response code of 3 (see section 3.44), followed by the Network Configured command after successful re-initialization of the network stack (see section 3.45).

**Table 3.46 Restore Factory Settings Command Fields**

Field	Width	Description
Validation String	4	This field must contain the number <i>0xf85a23e1</i> . The number must be transmitted in little-endian byte order (least significant byte first).

**Table 3.47 Restore Factory Settings Command Responses**

Code	Field	Length	Description
1	-	-	Message is too short.
2	-	-	Message is too long.
7	-	-	Invalid validation string.

### 3.28. Configure Multicast

This command configures the multicast settings. It configures how many multicast groups are present in the system and how many 16-bit fields of the IPv6 multicast address group identifier are used as full group ID fields. Refer to section 2.5.3.2 for a general description of IPv6 multicast and its implementation in ZWIR45xx SCI devices.

Multicast group counting starts at 1. Thus, attempts to join or leave multicast group 0 will result in an error.

Note that the memory for the internal bit-field storing group membership is allocated dynamically at runtime. SCI allocates one byte per 8 groups. For example, a configuration with 1024 groups requires 128 bytes of memory. If sufficient memory is not available, the SCI device will be reset and the stored settings will be restored. After the reset, the corresponding error response is returned.

**Table 3.48 Configure Multicast Command Fields**

Field	Width	Description
Group Count	2	Specifies the maximum number of available groups.
Group Address Count	1	Specifies how many Group Address fields are in an IPv6 address.

**Table 3.49 Configure Multicast Command Fields**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	Message is too short.

Code	Field	Length	Description
2	-	-	Message is too long.
5	-	-	Memory exhaustion.
10	-	-	Maximum group count exceeded.
11	-	-	Invalid group address number.

### 3.29. Join Multicast Group

This command is used to join a multicast group. The device will receive multicast messages that are sent to the Group Address immediately after executing this command.

**Table 3.50 Join Multicast Group Command Fields**

Field	Width	Description
Group Address	2	Specifies the address of the group the device is attempting to join.

**Table 3.51 Join Multicast Field Command Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	Message is too short.
2	-	-	Message is too long.
10	-	-	Invalid Group Address: either the Group Address is zero or it is larger than the configured group count.

### 3.30. Leave Multicast Group

This command is used to leave a multicast group. The device will no longer receive multicast messages that are sent to the Group Address immediately after executing this command.

**Table 3.52 Leave Multicast Group Command Fields**

Field	Width	Description
Group Address	2	Specifies the address of the group the device is leaving.

**Table 3.53 Leave Multicast Group Command Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	Message is too short.
2	-	-	Message is too long.
10	-	-	Invalid Group Address: either the Group Address is zero or it is larger than configured group count.

### 3.31. Add Security Policy

This command adds a security policy to the security policy database. Outgoing and incoming traffic is matched with the Remote Address, Prefix, Protocol, and Lower and Upper Port. If a rule is found in the security policy database, the traffic is processed by IPSec according to the specified security association. For manual administration of security associations, a security association must be generated before executing this command. The response value received from the Add Security Association must be passed as the last parameter. If automatic generation of security associations is desired, the last parameter must be zero. Refer to the IDT document *ZWIR45xx Application Note—Using IPSec and IKEv2 in 6LoWPANs* for further information about security processing.

Without a call to this command, a default rule is in effect that allows all outgoing and incoming traffic to pass unsecured. The first call to Add Security Policy will disable this rule and install the new one instead. Subsequent calls will add further rules.

IDT's SCI firmware allows the generation of a maximum of ten security policies. Policy matching is done in the order that the policies have been generated. If multiple policy matching parameters are overlapping, ensure that the most general rule is inserted last.

**Table 3.54 Add Security Policy Command Fields**

Field	Width	Description
Type	1	Defines the type of the rule: <i>0x11</i> Output Secure <i>0x12</i> Output Bypass <i>0x13</i> Output Drop <i>0x21</i> Input Secure <i>0x22</i> Input Bypass <i>0x23</i> Input Drop
Remote Address	16	The remote address to which this rule applies. Note: Only the <i>Prefix</i> number of higher bits are used in the matching algorithm.
Prefix	1	Defines how many of the upper bits of <i>Remote Address</i> are considered for matching traffic. This value can be in the range of 0 to 128.
Protocol	1	This field defines the protocol that the rule applies to: <i>0x00</i> Policy applies to any protocol <i>0x17</i> Policy applies to UDP traffic only <i>0x58</i> Policy applies to ICMPv6 traffic only
Lower Port	2	Defines the lower port of port-based protocols that the policy applies to
Upper Port	2	Defines the upper port of port-based protocols that the policy applies to.
SA Handle	4	Security association handle returned by the Add Security Association command (see section 3.32).

**Table 3.55 Add Security Policy Command Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	Message is too short.
2	-	-	Message is too long.
4	-	-	No more space in the SPD.
13	-	-	Invalid type.
14	-	-	Invalid prefix length.

### 3.32. Add Security Association

This command manually adds a security association to the security association database. On successful completion of this operation, the command returns a security index that must be passed to the Add Security Policy command to link the security association to a certain security policy. A security association can be assigned to multiple security policies.

Each security association requires a unique identifier that is passed with the first argument field. A maximum of ten security associations can be generated. If an item is added that has a security parameter ID that is already stored in the SAD, then the SAD entry is overwritten with the new security association.

Note that if no encryption is selected, packets are still sent with encryption headers. However, packets are not encrypted in this case. This feature can be used for debug purposes.

**Table 3.56 Add Security Association Command Fields**

Field	Width	Description
Security Parameter ID	4	A value identifying the security association.
Encryption Algorithm	1	Selects the encryption algorithm: 1 <sub>DEC</sub> No Encryption 13 <sub>DEC</sub> AES128, Counter Mode
Encryption Key	16	This is the 128-bit key used for encryption—must be the same for all communicating parties.
Encryption Nonce	4	This is a 32-bit value used for encryption. Must be the same for all communicating parties.
Authentication Algorithm	1	This determines the authentication algorithm: 0 No Authentication 5 AES XCBC96 Authentication
Authentication Key	16	This is the 128-bit key used for authentication—must be the same for all communicating parties.

**Table 3.57 Add Security Association Command Responses**

Code	Field	Length	Description
0	SA Handle	4	Handle of the newly generated security association.
1	-	-	Message is too short.
2	-	-	Message is too long.
4	-	-	No more space in the SAD.
15	-	-	Invalid Encryption Algorithm.
16	-	-	Invalid Authentication Algorithm.

### 3.33. Add IKEv2 Authentication Entry

This command generates a new IKEv2 authentication entry. These entries are used to authenticate the device in the process of key negotiation for an IPSec security association. A maximum of 5 IKEv2 authentication entries can be created on each device.

**Table 3.58 Add IKEv2 Authentication Entry Command Fields**

Field	Width	Description
Remote Address	16	The remote address that this rule applies to. Note: Only the <i>Prefix</i> number of higher bits are used in the matching algorithm.
Prefix	1	Defines how many of the upper bits of <i>Remote Address</i> are considered for matching traffic. This value can be in the range of 0 to 128.
Pre-Shared Key	16	This is the pre-shared key that is used for authentication.
ID-Length	1	Determines the size of the ID.
ID	ID- Length	This is the unique identifier of the IKEv2 authentication entry. It must not exceed 16 bytes.

**Table 3.59 Add IKEv2 Authentication Entry Command Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	Message is too short.
2	-	-	Message is too long.
4	-	-	No more space in the IKEv2 authentication database.
14	-	-	Invalid Prefix length.
17	-	-	Invalid ID-Length.

### 3.34. Enable Watchdog Timer

This command is used to enable the watchdog timer or to prepare for disabling it. Enable requests are performed immediately on reception of this command. Disabling the watchdog is done by a simple system reset when the watchdog is not enabled permanently (stored configuration does not have the watchdog enable bit set) or by resetting the watchdog enable bit, storing the new configuration, and resetting the device. It is not possible to disable a permanently configured watchdog without changing the stored configuration. For more information about the watchdog timer, refer to section 2.13.

**Table 3.60 Enable Watchdog Timer Command Fields**

Field	Width	Description
Enable Status	1	Watchdog is disabled when set to zero; otherwise it is enabled.
Validation String	4	The local UDP port over which the packet has been received.

**Table 3.61 Enable Watchdog Timer Responses**

Code	Field	Length	Description
0	-	-	Success – this response is only sent when acknowledges are enabled.
1	-	-	Message is too short.
2	-	-	Message is too long.
7	-	-	Invalid validation string.

### 3.35. Get Current Interface

This command returns the interface identifier of the interface over which this command was issued. For example, if the module receives this command over the UART1 interface, it returns the interface identifier byte *0x00*. This command does not expect any parameters. Extra parameters sent with the command are ignored.

**Table 3.62 Get Current Interface Responses**

Code	Field	Length	Description
0	IID	1	Success – this response is only sent when acknowledges are enabled.

### 3.36. Get PHY Configuration

This command returns the current PHY configuration. This command does not expect any parameters. Extra parameters sent with the command are ignored.

**Table 3.63 Get PHY Command Response**

Code	Field	Length	Description
0	Values	3	If the command is successful, the response code is 0 and the PHY configuration is returned in the response. See Table 3.213.20 in section 3.10 for the structure of the returned PHY parameter configuration.

### 3.37. Get NET Configuration

This command returns the current NET configuration. This command does not expect any parameters. Extra parameters sent with the command are ignored. This command is deprecated by the Get Parameter Command (3.41).

**Table 3.64 Get NET Configuration Command Response**

Code	Field	Length	Description
0	Values	15	If the command is successful, the response code is 0 and the NET configuration is returned in the response. See Table 3.23 in section 3.11 for the structure of the returned NET parameter configuration.

### 3.38. Get FCC ID Command

This command returns the module's FCC ID under which the modular certification with the FCC has been completed. The command does not accept any arguments. Extra arguments are ignored.

**Table 3.65 Get FCC ID Response**

Code	Field	Length	Description
0	FCC ID	Variable	The ASCII-encoded FCC ID of the module.

### 3.39. Enable Acknowledge Command

This command is used to enable or disable the transmission of acknowledgement packets for commands that have been executed successfully, and no response is sent otherwise. By default, acknowledgement transmission is disabled.

This command is always acknowledged, except for the case when the acknowledge-enable field is set to 0 and the device was previously configured to send no acknowledgements.

**Table 3.66 Enable Acknowledge Command Format**

Field	Width	Description
Enable	1	Acknowledges are disabled when the Enable field is set to 0; otherwise acknowledges are enabled.

**Table 3.67 Response for Enable Acknowledge Command**

Code	Field	Length	Description
0	-	-	Acknowledge transmission has been enabled or disabled successfully.

### 3.40. Set Parameter

Using this command, it is possible to modify single network stack parameters. This command deprecates the Configure NET command (see section 3.453.11). It provides an extended set of parameters. Table 3.70 lists the configurable parameters.

Note: For some parameters, a network reset is required to activate changes. For these parameters, if the network reset has not been performed and the Get Parameter command is sent (see section 3.41), the previous value of the parameter will be returned.

**Table 3.68 Set Parameter Command Format**

Field	Width	Description
Parameter ID	1	Unique ID of the parameter to be configured. See Table 3.70 for a list of available parameters.
Value	Variable	This is the actual value to be assigned. The length of this field depends on the parameter to be set. See Table 3.70 for length information.

**Table 3.69 Set Parameter Responses**

Code	Field	Length	Description
0	-	-	The parameter has been set successfully. Depending on the parameter being set, a network reset might be required in order to activate the changes.
1	-	-	Message is too short.
2	-	-	Message is too long.
3	-	-	Invalid Parameter ID.

**Table 3.70 Network Stack Configuration Parameters**

Name	ID	Length	Reset Required?	Description
Routing Table Size	0	2	Yes	
Neighbor Cache Size	1	1	Yes	
Max Socket Count	2	1	Yes	
Route Timeout	3	2	Yes	
Neighbor Reachable Time	4	2	Yes	
Max Hop Count	5	1	Yes	
Route Max Fail Count	6	1	Yes	
Route Min RSSI	7	1	Yes	
Route Min RSSI Red.	8	1	Yes	



Name	ID	Length	Reset Required?	Description
DAD Enable	9	1	Yes	
RS Enable	10	1	Yes	
Max Route Requests	11	1	Yes	
Hdr. Compr. Context 1	12	8	No	Header Compression Contexts 1 – 3 may be used to allow more efficient compression of global IPv6 addresses.
Hdr. Compr. Context 2	13	8	No	
Hdr. Compr. Context 3	14	8	No	
Neigh. Retransmit Time	15	2	Yes	
AAC Enable	16	1	No	

### 3.41. Get Parameter

This command is used to query currently configured network parameters. It deprecates the Get NET Configuration command (see section 3.37). A list of parameters that can be queried is given in Table 3.70.

Note: This command reports the active configuration. For parameter changes requiring a network reset, the previous value continues to be read if no network reset has been performed to activate the changes.

The response length of this command depends on the parameter being read. This command does not accept any arguments. Extra arguments are ignored without notification.

**Table 3.71 Get Parameter Command Format**

Field	Width	Description
Parameter ID	1	Unique ID of the parameter to be read. See Table 3.70 for a list of available parameters.
Value	Variable	This is the actual value to be assigned. The length of this field depends on the parameter set. Please refer to Table 3.70 for length information.

**Table 3.72 Get Parameter Response Format**

Code	Field	Length	Description
0	Value	Variable	Zero success code, followed by the value of the parameter being requested. The length of the value depends on the parameter. See Table 3.70 for parameter length information.
1	-	-	Message is too short.
3	-	-	Invalid Parameter ID.

### 3.42. Receive Packet Command

This command is sent by the SCI device when a data packet has been received over one of the configured receive sockets. The packet is sent to the interface that has been configured using the Configure Receiver command (see section 3.2).

**Table 3.73 Receive Packet Command Fields**

Field	Width	Description
Source IPv6 Address	16	The address from which the packet has been received.
Local UDP Port	2	The local UDP port over which the packet has been received.
Data	1 - 1232	The data payload that has been received with the packet.
RSSI	1	When RSSI transmission has been enabled with the Configure Receiver command, this byte-field is included.

### 3.43. Remote Response Command

This command is sent by the SCI device when a reply of a remotely executed command is received. The packet is sent to the interface over which the triggering command had been sent.

**Table 3.74 Remote Response Command Fields**

Field	Width	Description
Source IPv6 Address	16	The address from which the packet has been received.
Response Frame	1 - 1232	The command response that has been received.

### 3.44. Reset Report Command

This command is sent by the SCI device whenever it is reset. The source of the reset can be determined from the reset reason field.

Note: All module settings, except the acknowledge enable, are reset to the last stored configuration and all network sockets are closed after reception of this command.

The command is sent to the serial interface that received the last command before execution of the reset.

**Table 3.75 Reset Report Command Fields**

Field	Width	Description
Reset Reason	1	This field determines the source of the reset.

Possible Reset Reason values are

- 0 → Power On
- 1 → Standby Wakeup
- 2 → Watchdog
- 3 → Software
- 4 → Reset Pin
- 6 → Brown Out

### 3.45. Network Configured Command

This command is sent by the SCI after successful initialization of the network stack. Network stack initialization happens as a result of a full reset or a network reset. No packets can be sent or received before this command is received. However, configuring network parameters and opening communication sockets is possible before reception of this command.

If no commands are sent to the SCI after reset, this command is sent to the serial interface that received the last command before execution of the reset. Otherwise, the interface on which the last command was received is used.

## 4 Glossary

Term	Description
6LoWPAN	IPv6 over Low Power Wireless Personal Area Networks
AES	Advanced Encryption Standard
AH	Authentication Header
API	Application Programming Interface
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
BPSK	Binary Phase-Shift Keying
CBC	Cyclic Block Cipher
DAD	Duplicate Address Detection
DHCP	Dynamic Host Configuration Protocol
ESP	Encapsulating Security Payload
FCC	Federal Communications Commission
GID	Group Identifier
GPIO	General Purpose Input/Output
IETF	Internet Engineering Task Force
IKEv2	Internet Key Exchange version 2
IPSec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
MAC	Media Access Control
LAN	Local Area Network
MCU	Microcontroller Unit
NDP	Neighbor Discovery Protocol
NA	Neighbor Advertisement
NS	Neighbor Solicitation
OSI	Open Systems Interconnection
OTAU	Over-The-Air Update
PAN	Personal Area Network
PLL	Phase-Locked Loop
PSK	Pre-Shared Key
QPSK	Quadrature Phase-Shift Keying

<b>Term</b>	<b>Description</b>
RA	Router Advertisement
RS	Router Solicitation
RSSI	Receive Signal Strength Indicator
SA	Security Association
SAD	Security Association Database
SCI	Serial Command Interface
SP	Security Policy
SPD	Security Policy Database
SPI	Serial Peripheral Interface
TRX	Transceiver
UART	Universal Asynchronous Receiver Transmitter
UDP	User Datagram Protocol
WAN	Wide Area Network
WPAN	Wireless Personal Area Network

## 5 Related Documents

Document Name	Source
<b>Public Documents</b>	
Internet Protocol, Version 6 (IPv6) Specification	RFC 2460, <a href="http://tools.ietf.org/html/rfc2460">http://tools.ietf.org/html/rfc2460</a>
IP Version 6 Addressing Architecture	RFC 4291, <a href="http://tools.ietf.org/html/rfc4291">http://tools.ietf.org/html/rfc4291</a>
Security Architecture for the Internet Protocol	RFC 4301, <a href="http://tools.ietf.org/html/rfc4301">http://tools.ietf.org/html/rfc4301</a>
Internet Key Exchange (IKEv2) Protocol	RFC 5996, <a href="http://tools.ietf.org/html/rfc5996">http://tools.ietf.org/html/rfc5996</a>
Neighbor Discovery for IP Version 6 (IPv6)	RFC 4861, <a href="http://tools.ietf.org/html/rfc4861">http://tools.ietf.org/html/rfc4861</a>
IPv6 Stateless Address Auto-configuration	RFC 4862, <a href="http://tools.ietf.org/html/rfc4862">http://tools.ietf.org/html/rfc4862</a>
Transmission of IPv6 Packets over IEEE 802.15.4 Networks	RFC 4944, <a href="http://tools.ietf.org/html/rfc4944">http://tools.ietf.org/html/rfc4944</a>

IDT Documents
<i>ZWIR4512 Data Sheet</i>
<i>ZWIR451x Programming Guide *</i>
<i>ZWIR451x Application Note: Using IPSec and IKEv2 in 6LoWPANs *</i>
<i>ZWIR451x Application Note: Using Over-the-Air Updates in 6LoWPANs *</i>

Visit the ZWIR4512 product page [www.IDT.com/ZWIR4512](http://www.IDT.com/ZWIR4512) or contact your nearest sales office for the latest version of the ZMD documents.

\* Note: Documents marked with an asterisk (\*) require a free customer login account for access.

## 6 Document Revision History

Revision	Date	Description
1.00	May 9, 2012	Initial public release. Added documentation of configurable network parameters. Added over-the-air update security notice.
1.10	July 17, 2012	Added documentation for new commands in firmware version 1.1 (Get PHY Configuration and Get NET Configuration).
1.11	July 30, 2012	Clarified documentation of security configuration command. Minor edits
1.20	July 30, 2014	Added documentation for RSSI feature for received network packets. Added documentation for new commands: <ul style="list-style-type: none"> <li>• Get FCC ID</li> <li>• Acknowledge Enable</li> <li>• Reset Report</li> <li>• Network Configured</li> <li>• Set Parameter</li> <li>• Get Parameter</li> </ul> Updated command response tables to include acknowledgements. Included ZWIR4512AC2-related pin descriptions. Added radio channel table and updated power limits of EU channels. Fixed initial value for maximum socket count. Minor edits for clarity. Update for contact information and imagery on cover and headers.
	April 15, 2016	Changed to IDT branding.

## IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

### Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,  
Koto-ku, Tokyo 135-0061, Japan  
[www.renesas.com](http://www.renesas.com)

### Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

### Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit [www.renesas.com/contact-us/](http://www.renesas.com/contact-us/).