



Software Design Specification

Z-Wave Device Class Specification

Document No.:	SDS10242
Version:	17
Description:	This document describes the Device and Command Classes used by Z-Wave enabled products ensuring that compliant products will be interoperable.
Written By:	JFR
Date:	2009-05-05
Reviewed By:	CHL;JRM;SGR;ABR
Restrictions:	None

Approved by:

Date	CET	Initials	Name	Justification
2009-05-05	16:21:04	NTJ	Niels Thybo Johansen	

This document is the property of Zensys A/S. The data contained herein, in whole or in part, may not be duplicated, used or disclosed outside the recipient for any purpose. This restriction does not limit the recipient's right to use information contained in the data if it is obtained from another source without restriction.



CONFIDENTIAL

REVISION RECORD

Doc. Rev	Date	By	Pages affected	Brief description of changes
1	20050224	HEH JFR	See SDS11060	Details about when the Association command class is necessary added. Described details about assigning return routes in conjunction with the Association Set command. Described how to respond on an Association Get command.
1	20050301	JFR	See SDS11060	Described the No Operation command class
1	20050405	JFR	See SDS11060	manufacturer IDs updated
1	20050409	JFR	See SDS11060	Clarified when the Proprietary command class can be used
2	20050625	JFR	See SDS11060	Multilevel Switch Do Level Change Command removed in Multilevel Switch Command Class
3	20051207	JFR	See SDS11060	Discontinued roll over bit in Multilevel Switch Start Level Change Command in Multilevel Switch Command Class
3	20051208	JFR	Section 7.2	Rocker switch example updated with support of the Association Command Class
3	20051208	JFR	Section 5.18 See SDS11060	Handling of a command class coming with different versions added
3	20051209	JFR	See SDS11060	Hail Command Class added
3	20051209	JFR	See SDS11060	Manufacturer Proprietary Command Class added
3	20051209	JFR	Section 5.3.2.2 Section 5.4.3.2 Section 5.2.3.3 Section 5.6.3.3 See SDS11060	Scene Device Class added
4	20060105	MVO	All	New 1 st page/header/footer contents. New Doc No
4	20060131	JFR	See SDS11060	Clarified how the reports to follow field must be interpreted.
4	20060131	JFR	See SDS11060	Manufacturer Proprietary Command Class can optionally use the API call ZW-SendDataMeta.
4	20060212	JFR	See SDS11060	Node Naming and Location Command Class added
4	20060229	JFR	See SDS11060	Multilevel Switch command class version 2 added
4	20060229	JFR	See SDS11060	Interpretation of the "Ignore start level" bit updated
4	20060229	JFR	Section 5.18	Devices supporting older versions of a command class
4	20060229	JFR	Section 7.1.2.2	Updated to Multilevel Switch command class version 2 in the example
4	20060317	JFR	Section 3.3	the Optional Functionality flag added to the node information frame
5	20060425	JFR	Front page	Fixed front page formatting problems
6	20060425	JFR	Front page	Fixed front page formatting problems again
7	20060426	BGR JFR	Section 5.2 See SDS11060 Section 5.6 See SDS11060 See SDS11060	Basic command class mapping moved to the specific device class level for Binary Switch Generic Device Class. Binary Switch Command Class parameters clarified Basic command class mapping moved to the specific device class level for Multilevel Switch Generic Device Class. Multilevel Switch Command Class parameters clarified Basic Command Class parameters clarified
8	20060518	JFR	See SDS11060 See SDS11060 See SDS11060	Time Command Class added Time Parameters Command Class added Geographical Location Command Class added
9	20060710	JFR	Section 5.6.3.4	Multiposition Motor Device Class added to the Multilevel Switch Generic Device Class
9	20060825	JFR	See SDS11060 See SDS11060	Composite Command Class added Association Command Class version 2 added
9	20060825	JFR	Section 5.16.3.4 See SDS11060	AV Control Point Device Class added Simple AV Control Command Class added
10	20061011	JFR	See SDS11060	Manufacturer IDs updated
10	20061013	JFR	See SDS11060 See SDS11060 Section 5	Screen Meta Data Command Class added Screen Attributes Data Command Class added Basic command class mapping for devices clarified
11	20061218	JFR	See SDS11060 Section 2.3 Section 2.4	Manufacturers ID updated Precedence of definitions added Terms used added

11	20070208	JFR	Section 5.16.3.4 See SDS11060 See SDS11060 See SDS11060 See SDS11060	Setback Schedule Thermostat Specific Device Class added Wake Up Command Class Version 2 added Multi Command Command Class added Climate Control Schedule Command Class added Multi Instance Association Command Class added
11	20070302	JFR	Section 5.3.2.2 & 5.3.2.2.1 Section 5.4.3.2, 5.4.3.2.1 & 5.4.3.2.2	Scene Activation Command Class was erroneously listed as a mandatory command class to support for a portable scene controller, which typically is battery-operated. Scene Activation Command Class should not be listed as a mandatory command class but recommended to handle scene controllers which control a load locally.
12	20070509	JFR	5.16.3.4.1.1 See SDS11060	Basic Cmd Class mapping clarified for a Setback Schedule Thermostat device. Motor controls behaviour clarified in the Multilevel Switch Set command.
12	20070715	JFR	See SDS11060	Added support for Windows Vista Media Center and Media Center 2005 remote controls in Simple AV Control command class
12	20070730	SBD	Section 4 Section 5	Basic Command Class requirements and mappings partially updated. SUC and SIS criteria added for controllers. Network maintenance functionality added for slaves and controllers. Thermostat General version 2 Specific Device Class added. Satellite Receiver version 2 Specific Device Class added.
12	20070813	JFR	See SDS11060	Manufacturers ID updated
12	20070816	JFR	See SDS11060 Section 5.18 See SDS11060 See SDS11060 Section 5.13.4.3 See SDS11060 See SDS11060 See SDS11060 See SDS11060 See SDS11060 See SDS11060 See SDS11060	Multilevel Sensor Command Class Version 2 added Meter Generic Device Class added Meter Command Class added IP Configuration Command Class added Advanced Door Lock Specific Device Class added Door Lock Command Class added User Code Command Class added Grouping Name Command Class added Remote Association Configuration Command Class added Remote Association Activate Command Class added Typo fixed in End Point Mask addressing
12	20070913	JFR	See SDS11060 See SDS11060 See SDS11060 See SDS11060 See SDS11060	Screen Meta Data Command Class Version 2 added. Screen Attributes Command Class Version 2 added. Protection Command Class Version 2 added. Firmware Update Meta Data Command Class added. Table with Z-Wave Protocol version listed for a given Developer's Kit version added.
13	20070920	JFR	Section removed	All Command Classes moved to document SDS11060
13	20071023	SBD	Section 5.13.4.3	Advanced door lock specification device class clarification added.
13	20071227	JFR	Section 5.1	Alarm Sensor Device Class added Basic Report mapping clarified
13	20071227	JFR	Section 5.16.3.5	Setback Thermostat Specific Device Class added
13	20080102	JFR	Section 5.16.3.6	Setpoint Thermostat Specific Device Class added
14	20080519	SBD	Section 3 Section 3.2	References updated. Added note about selecting Device and Command Classes for devices.
14	20080812	JFR	Section 5.19 Section 5.21	Z/IP Gateway Generic Device Class added Z/IP Node Generic Device Class added
14	20080813	JFR	Section 5.3.2.3 Section 5.4.3.3	Portable Installer Tool Specific Device Class added Static Installer Tool Specific Device Class added
14	20080814	JFR	Section 5.12	Display Generic Device Class added
15	20090217	JFR	Section 5.17.3.4	Doorbell Specific Device Class added
16	20090218	JFR	Section 5.17.3 Section 5.13.4.4	Doorbell Specific Device Class identifier added Secure Keypad Door Lock Specific Device Class added
17	20090405	JFR	Section 5.21 Section 5.20 Section 5.19	Z/IP Node Specific Device Class renamed to Z/IP Tunneling Node Z/IP Gateway Specific Device Class renamed to Z/IP Tunneling Gateway Ventilation Generic Device Class added
17	20090430	JFR	Section 5.6.3.5, 5.6.3.6 & 5.6.3.7 Section 5.6.3.4	Multilevel Switch Generic Device Class extended with Motor Control A, B & C specific device classes. Multiposition Motor specific device class not recommended anymore.

Table of Contents

1	ABBREVIATIONS.....	1
2	INTRODUCTION.....	2
2.1	Purpose	2
2.2	Audience and Prerequisites	2
2.3	Precedence of definitions	2
2.4	Terms used in this document	2
3	OVERVIEW	3
3.1	Device Classes.....	5
3.2	Command Classes	6
3.3	Node Information Frame	7
3.3.1	Z-Wave Protocol Specific Part.....	7
3.3.2	Application Specific Part.....	7
4	BASIC DEVICE CLASSES.....	10
4.1	Portable Controller	11
4.2	Static Controller	12
4.3	Slave.....	13
4.4	Routing Slave	14
5	GENERIC AND SPECIFIC DEVICE CLASSES	15
5.1	Alarm Sensor Generic Device Class	20
5.1.1	Mandatory Command Classes to Support	20
5.1.1.1	Basic Command Class Implementation	20
5.1.2	Mandatory Command Classes to Control	20
5.1.3	Specific Device Classes	21
5.1.3.1	No Specific Device Class defined	22
5.1.3.2	Basic Routing Alarm Sensor Specific Device Class	23
5.1.3.3	Routing Alarm Sensor Specific Device Class	25
5.1.3.4	Basic Zensor Net Alarm Sensor Specific Device Class	27
5.1.3.5	Zensor Net Alarm Sensor Specific Device Class	29
5.1.3.6	Advanced Zensor Net Alarm Sensor Specific Device Class.....	31
5.1.3.7	Basic Routing Smoke Sensor Specific Device Class.....	33
5.1.3.8	Routing Smoke Sensor Specific Device Class	35
5.1.3.9	Basic Zensor Net Smoke Sensor Specific Device Class	37
5.1.3.10	Zensor Net Smoke Sensor Specific Device Class	39
5.1.3.11	Advanced Zensor Net Smoke Sensor Specific Device Class	41
5.2	Binary Switch Generic Device Class	43
5.2.1	Mandatory Command Classes to Support	43
5.2.2	Mandatory Command Classes to Control	43
5.2.3	Specific Device Classes	44
5.2.3.1	No Specific Device Class defined	45
5.2.3.2	Binary Power Switch Specific Device Class.....	46
5.2.3.3	Binary Scene Switch Specific Device Class.....	48
5.3	Remote Controller Generic Device Class	50
5.3.1	Mandatory Command Classes to Control	50
5.3.2	Specific Device Classes	51
5.3.2.1	Portable Remote Controller Specific Device Class	52
5.3.2.2	Portable Scene Controller Specific Device Class	53
5.3.2.3	Portable Installer Tool Specific Device Class	55
5.4	Static Controller Generic Device Class	56
5.4.1	Recommended Command Classes to Support	56
5.4.1.1	Basic Command Class Implementation	56

5.4.2	Mandatory Command Classes to Control	56
5.4.3	Specific Device Classes	57
5.4.3.1	PC Controller Specific Device Class	58
5.4.3.2	Scene Controller Specific Device Class	59
5.4.3.3	Static Installer Tool Specific Device Class	61
5.5	Repeater Slave Generic Device Class	62
5.5.1	Mandatory Command Classes to Support	62
5.5.1.1	Basic Command Class Implementation	62
5.5.2	Mandatory Command Classes to Control	62
5.5.3	Specific Device Classes	63
5.5.3.1	Basic Repeater Slave Specific Device Class	64
5.6	Multilevel Switch Generic Device Class	66
5.6.1	Mandatory Command Classes to Support	66
5.6.2	Mandatory Command Classes to Control	66
5.6.3	Specific Device Classes	67
5.6.3.1	No Specific Device Class defined	68
5.6.3.2	Multilevel Power Switch Specific Device Class	69
5.6.3.3	Multilevel Scene Switch Specific Device Class	70
5.6.3.4	Multiposition Motor Specific Device Class (Not recommended)	72
5.6.3.5	Motor Control Class A Specific Device Class	73
5.6.3.6	Motor Control Class B Specific Device Class	75
5.6.3.7	Motor Control Class C Specific Device Class	77
5.7	Toggle Switch Generic Device Class (Not recommended)	79
5.7.1	Mandatory Command Classes to Support	79
5.7.1.1	Basic Command Class Implementation	79
5.7.2	Mandatory Command Classes to Control	79
5.7.3	Specific Device Classes	80
5.7.3.1	Binary Toggle Switch Specific Device Class (Not recommended)	81
5.7.3.2	Multilevel Toggle Switch Specific Device Class (Not recommended)	82
5.8	Remote Switch Generic Device Class	83
5.8.1	Mandatory Command Classes to Support	83
5.8.2	Mandatory Command Classes to Control	83
5.8.3	Specific Device Classes	84
5.8.3.1	Binary Remote Switch Specific Device Class	85
5.8.3.2	Multilevel Remote Switch Specific Device Class	85
5.8.3.3	Binary Toggle Remote Switch Specific Device Class (Not recommended)	86
5.8.3.4	Multilevel Toggle Remote Switch Specific Device Class (Not recommended)	86
5.9	Binary Sensor Generic Device Class	87
5.9.1	Mandatory Command Classes to Support	87
5.9.1.1	Basic Command Class Implementation	87
5.9.2	Mandatory Command Classes to Control	87
5.9.3	Specific Device Classes	88
5.9.3.1	Routing Binary Sensor Specific Device Class	89
5.10	Multilevel Sensor Generic Device Class	90
5.10.1	Mandatory Command Classes to Support	90
5.10.1.1	Basic Command Class Implementation	90
5.10.2	Mandatory Command Classes to Control	90
5.10.3	Specific Device Classes	91
5.10.3.1	Routing Multilevel Sensor Specific Device Class	92
5.11	Pulse Meter Generic Device Class	93
5.11.1	Mandatory Command Classes to Support	93
5.11.1.1	Basic Command Class Implementation	93
5.11.2	Mandatory Command Classes to Control	93
5.11.3	Specific Device Classes	94
5.12	Display Generic Device Class	95
5.12.1	Mandatory Command Classes to Support	95
5.12.1.1	Basic Command Class Implementation	95

5.12.2	Mandatory Command Classes to Control	95
5.12.3	Specific Device Classes	96
5.12.3.1	No Specific Device Class defined	97
5.12.3.2	Simple Display Specific Device Class	98
5.13	Entry Control Generic Device Class	100
5.13.1	Mandatory Command Classes to Support	100
5.13.1.1	Basic Command Class Implementation	100
5.13.2	Recommended Command Classes to Support	100
5.13.3	Mandatory Command Classes to Control	100
5.13.4	Specific Device Classes	101
5.13.4.1	Specific Device Class Not Used	102
5.13.4.2	Door Lock Specific Device Class	103
5.13.4.3	Advanced Door Lock Specific Device Class	104
5.13.4.4	Secure Keypad Door Lock Specific Device Class	105
5.14	Semi Interoperable Generic Device Class	107
5.14.1	Mandatory Command Classes to Support	107
5.14.1.1	Basic Command Class Implementation	108
5.14.2	Mandatory Command Classes to Control	108
5.14.3	Specific Device Classes	109
5.14.3.1	Energy Production Specific Device Class	110
5.15	Window Covering Generic Device Class (Not recommended)	111
5.15.1	Mandatory Command Classes to Support	111
5.15.1.1	Basic Command Class Implementation	111
5.15.2	Mandatory Command Classes to Control	111
5.15.3	Specific Device Classes	111
5.15.3.1	Simple Window Covering Control Specific Device Class (Not recommended)	112
5.16	Thermostat Generic Device Class	113
5.16.1	Mandatory Command Classes to Support	113
5.16.1.1	Basic Command Class Implementation	113
5.16.2	Mandatory Command Classes to Control	113
5.16.3	Specific Device Classes	113
5.16.3.1	Thermostat Heating Specific Device Class (Not permitted)	114
5.16.3.2	Thermostat General Specific Device Class (Not permitted)	115
5.16.3.3	Thermostat General V2 Specific Device Class	116
5.16.3.4	Setback Schedule Thermostat Specific Device Class	117
5.16.3.5	Setback Thermostat Specific Device Class	124
5.16.3.6	Setpoint Thermostat Specific Device Class	126
5.17	AV Control Point Generic Device Class	129
5.17.1	Mandatory Command Classes to Support	129
5.17.1.1	Basic Command Class Implementation	129
5.17.2	Mandatory Command Classes to Control	129
5.17.3	Specific Device Classes	130
5.17.3.1	No Specific Device Class defined	131
5.17.3.2	Satellite Receiver Specific Device Class (Not recommended)	132
5.17.3.3	Satellite Receiver V2 Specific Device Class	133
5.17.3.4	Doorbell Specific Device Class	134
5.18	Meter Generic Device Class	136
5.18.1	Mandatory Command Classes to Support	136
5.18.1.1	Basic Command Class Implementation	136
5.18.2	Mandatory Command Classes to Control	136
5.18.3	Specific Device Classes	137
5.18.3.1	No Specific Device Class defined	138
5.18.3.2	Simple Meter Specific Device Class	139
5.19	Ventilation Generic Device Class	140
5.19.1	Mandatory Command Classes to Support	140
5.19.2	Mandatory Command Classes to Control	140
5.19.3	Specific Device Classes	140

5.19.3.1	No Specific Device Class defined	141
5.19.3.2	Residential Heat Recovery Ventilation Specific Device Class	142
5.20	Z/IP Gateway Generic Device Class	143
5.20.1	Mandatory Command Classes to support	143
5.20.1.1	Basic Command Class implementation	143
5.20.2	Mandatory Command Classes to control	143
5.20.3	Specific Device Classes	144
5.20.3.1	Z/IP Tunneling Gateway Specific Device Class	145
5.20.3.2	Advanced Z/IP Gateway Specific Device Class	147
5.21	Z/IP Node Generic Device Class	148
5.21.1	Mandatory Command Classes to support	148
5.21.2	Mandatory Command Classes to control	148
5.21.3	Specific Device Classes	149
5.21.3.1	Z/IP Tunneling Node Specific Device Class	150
5.21.3.2	Advanced Z/IP Node Specific Device Class	151
6	COMMAND CLASSES	152
7	DEVICE EXAMPLES	153
7.1	Outlet adapter with dimming capability	153
7.1.1	Node Information	154
7.1.2	Switch Functionality	154
7.1.2.1	Switching On and Off	154
7.1.2.2	Dimming	156
7.1.2.3	All On/All Off	158
7.2	Battery-powered rocker switch	159
7.2.1	Node Information	160
7.2.2	Rocker Switch Functionality	160
7.2.2.1	Association	160
7.2.2.2	Wake Up	164
7.2.2.3	Battery	165
7.2.2.4	Switching On/Off and Dimming	166
	REFERENCES	167
	INDEX	168

List of Figures

Figure 1.	Z-Wave Home Control Network	3
Figure 2.	Z-Wave enabled application and underlying device structure	4
Figure 3.	Generic and Specific Device Class Structure	5
Figure 4.	Command class and related commands	6
Figure 5.	Node Information frame format	7
Figure 6.	Support/control command class list format	9
Figure 7.	Sequence diagram of operation after wake up	121
Figure 8.	Sequence diagram of operation for getting schedules	123
Figure 9.	Sequence diagram of operation after wake up	128
Figure 10.	Outlet adapter with dimming capabilities	153
Figure 11.	Battery-powered rocker switch	159

List of Tables

Table 1.	Basic Device Class identifiers	10
Table 2.	Valid combinations of Generic and Specific Device Classes	17

Table 3. Generic Device Class identifiers	18
Table 4. Specific Device Class identifiers for the Binary Switch Generic Device Class	44
Table 5. Specific Device Class identifiers for the Remote Controller Generic Device Class	51
Table 6. Specific Device Class identifiers for the Static Controller Generic Device Class.....	57
Table 7. Specific Device Class identifiers for the Repeater Slave Generic Device Class	63
Table 8. Specific Device Class identifiers for Multilevel Switch Generic Device Class.....	67
Table 9. Specific Device Class identifiers for the Toggle Switch Generic Device Class.....	80
Table 10. Specific Device Class identifiers for the Remote Switch Generic Device Class.....	84
Table 11. Specific Device Class identifiers for the Binary Sensor Generic Device Class.....	88
Table 12. Specific Device Class identifiers for the Multilevel Sensor Generic Device Class.....	91
Table 13. Specific Device Class identifiers for the Pulse Meter Generic Device Class	94
Table 14. Specific Device Class identifiers for the Entry Control Generic Device Class	101
Table 15. Specific Device Class identifiers for the Semi Interoperable Generic Device Class.....	109
Table 16. Specific Device Class identifiers for the Window Covering Generic Device Class.....	111
Table 17. Specific Device Class identifiers for the Thermostat Generic Device Class	113

1 ABBREVIATIONS

Abbreviation	Explanation
AMR	Automatic Meter Reading
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange. An ASCII code is the numerical representation of a character.
AV	Audio/Video
DHCP	Dynamic Host Configuration Protocol.
DNS	Dynamic Host Service
DST	Daylight Savings Time
HRV	Heat Recovery Ventilation
ID	Identifier
IP	Internet Protocol
IPV4	Internet Protocol version 4
IPV6	Internet Protocol version 6
LF	Linefeed character.
LSB	Less significant byte
MSB	Most significant byte
PIR	Pyroelectric Infrared Motion Sensor
SUC	Static Update Controller
TZO	Time Zone Offset
Unicode	Unicode is a standard for encoding of characters. For more information please visit http://www.unicode.org/
UTC	Universal Time (sometimes also called "Zulu Time") was called Greenwich Mean Time (GMT) before 1972
WMC	Windows Vista Media Center and Media Center 2005 remote controls

2 INTRODUCTION

This document describes the device classes that must be used when designing and implementing Z-Wave™ products. The document also contains a description of the command classes and associated commands that must be used by Z-Wave products when communicating with each other. A subset of command classes is typically mandatory for a given type of device. All commands are handled by the application layer of the Z-Wave protocol.

A certification program based partly on self-certification and partly on 3rd party certification is put in place to enable manufacturers to get their product Z-Wave certified in order to ensure interoperability and get permission to carry the Z-Wave logo on the product. Interoperability is the successful interworking of multiple products from multiple manufacturers, for multiple applications, that may be based on multiple versions of Z-Wave. For details about the certification program refer to 0.

In case the current device and command classes described in this document do not sufficiently support the product to be developed, please contact Zensys to initiate the required development process. For details about the device/command class development process refer to [2].

2.1 Purpose

The purpose of this document is to describe the device and command classes used by the application layer of the Z-Wave protocol.

2.2 Audience and Prerequisites

The audience of this document is Z-Wave partners and Zensys.

2.3 Precedence of definitions

In terms of reviewing products for Z-Wave Compliance, definitions in this document have precedence over the header file ZW_classcmd.h distributed as part of the Z-Wave Developer's Kit. However, the assignment of all device and command class hex identifiers can only be found in the header file ZW_classcmd.h.

Device and Command Class Specifications approved as final version (ver. 1.00) during the device/command class development process have precedence over this document temporarily until integrated into this document.

2.4 Terms used in this document

This document describes mandatory and optional aspects of the required compliance of a Z-Wave product to the Z-Wave standard.

The words “shall” and “must” specify aspects that are mandatory for compliance. Equally, “must not” has to be adhered to for compliance. Products that are in violation any such statement are considered to be **not** Z-Wave compliant.

The words “may”, “could”, and “may not” leave the choice to the implementer. “Recommended” also leaves the choice formally to the manufacturer, but provides additional guidance.

3 OVERVIEW

In order to achieve multi-vendor operation within and across subsystems in a home control system a standardized way to identify devices and interaction between devices must be defined. This allows a remote control from one vendor to control an outlet with dimming capabilities from another vendor within a lighting subsystem. Further, when a burglar subsystem is armed it can ask a lighting subsystem to start home simulation in order to exploit interoperability across different subsystems.

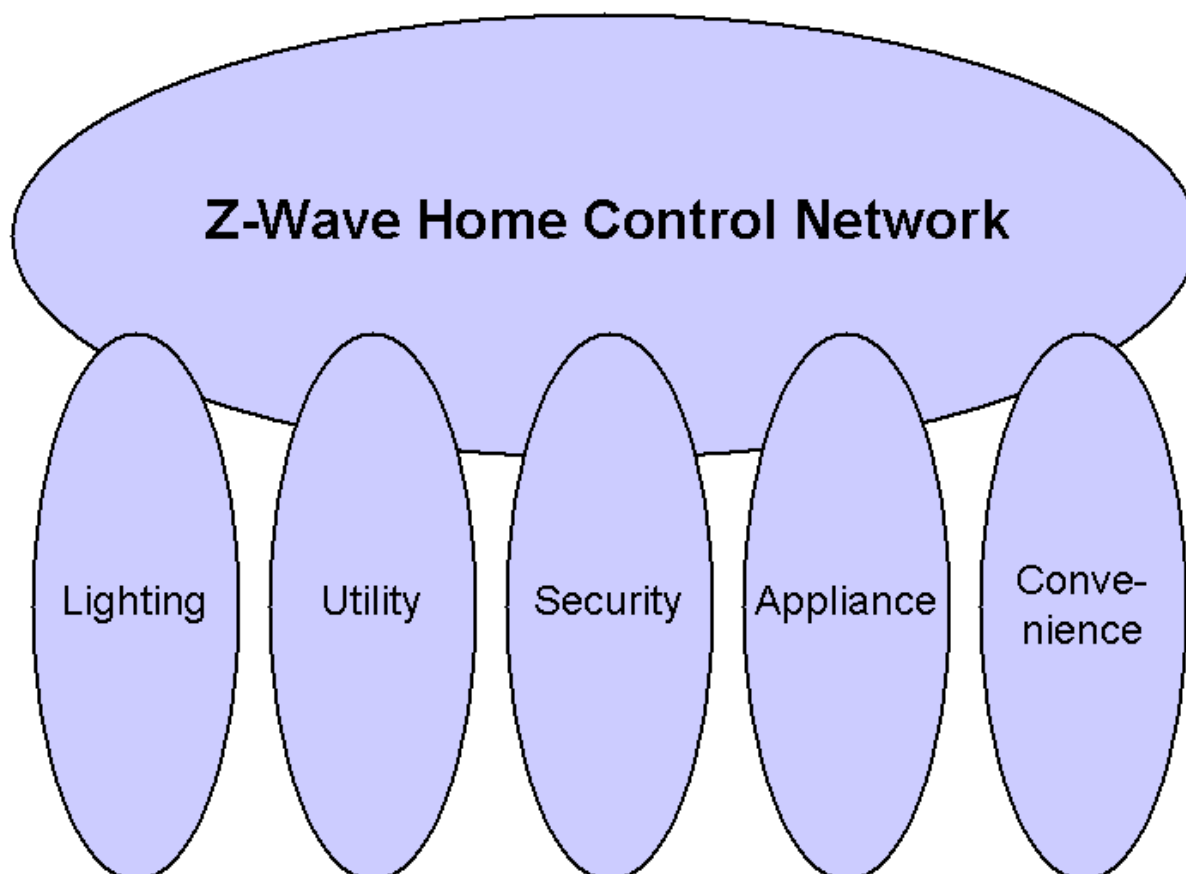


Figure 1. Z-Wave Home Control Network

The Z-Wave technology use basic, generic and specific device classes to identify a device and its role in the home control network. The basic device class determines the available functionality of the Z-Wave protocol depending on the library used. The Generic Device Class defines on a very high level the main functionality of a device. Generic device classes only defines the absolute minimum of functionality of a given type of device, so it is rare that any real products will be sufficiently described based on the Generic Device Class solely. The generic device class is therefore typically extended with a specific device class to define a set of additional mandatory and recommended command classes that a given product must support to obtain the wanted functionality.

Communication between devices is carried out by a number of commands organized into a range of command classes. Command classes are the most fundamental grouping of commands, and these include all the necessary commands required to implement a given functionality in a device. A device typically contains a number of different functionalities and consist therefore of a logic grouping of the necessary command classes into one device. The wanted functionality of a device is achieved by selecting the appropriate command classes in addition to the mandatory command classes for the

selected generic/specific device. This enables each vendor to provide devices with features that differentiate their product in the marketplace and at the same time achieve a high degree of interoperability.

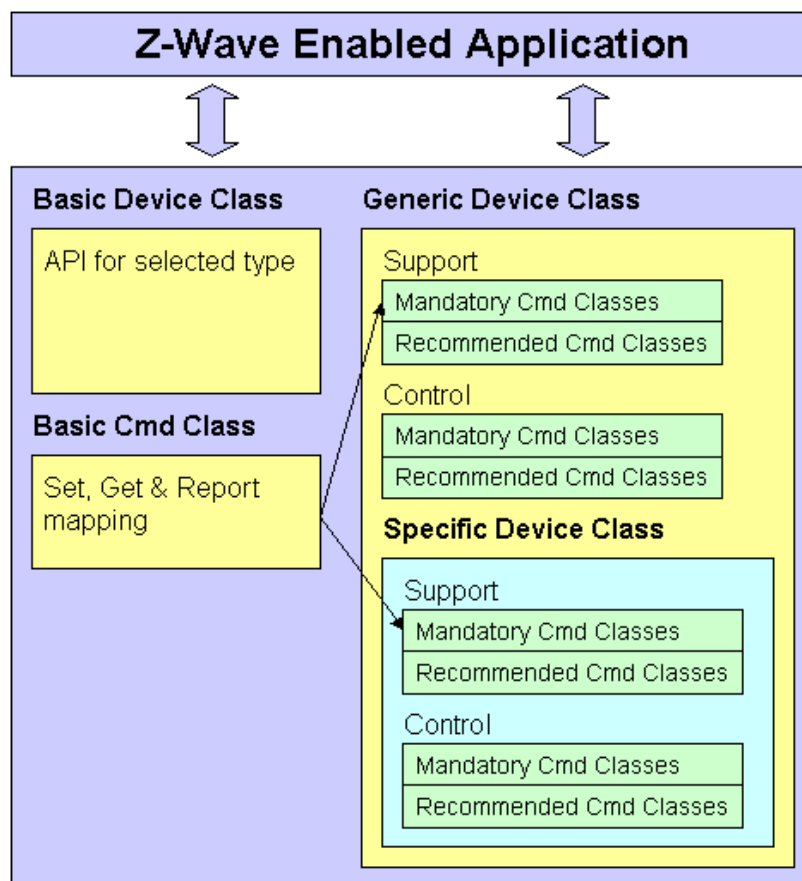


Figure 2. Z-Wave enabled application and underlying device structure

The device application is based upon the selected combination of the basic and generic/specific device class defining a number of mandatory command classes. Additional command classes can be selected to extend the functionality of the device.

The command classes are divided into two different groups. One group of command classes are supported by the device itself, e.g. a temperature sensor will support a command class allowing other devices to read the temperature. Another group of command classes are used to control functionality in other devices, e.g. a battery operated switch controlling a ceiling outlet to a lamp. All combinations of generic/specific device class support the Basic Command Class as default and specify how it is mapped into one of the mandatory command classes supported by the device itself.

NOTICE: Compliance in controlling via Basic Command Class means the controller is able to configure all basic set values or able to learn the values by inquiring the device in question by a basic get command.

NOTICE: It is mandatory to use the specified device and command classes in the customer applications. If further device and command classes are needed to implement the appropriate functionality in the application, please contact Zensys to obtain the necessary classes.

3.1 Device Classes

A device class groups devices with the same functionality together.

The Basic Device Class provides the device with a certain role/functionality in the Z-Wave network based on the type of library used. A detailed description of all available Basic Device Classes is given in Chapter 4. In conjunction with the Basic Device Class a Generic/Specific Device Class is selected to achieve the wanted functionality. Generic Device Classes contain Command Classes that are mandatory for all devices that are within this device class.

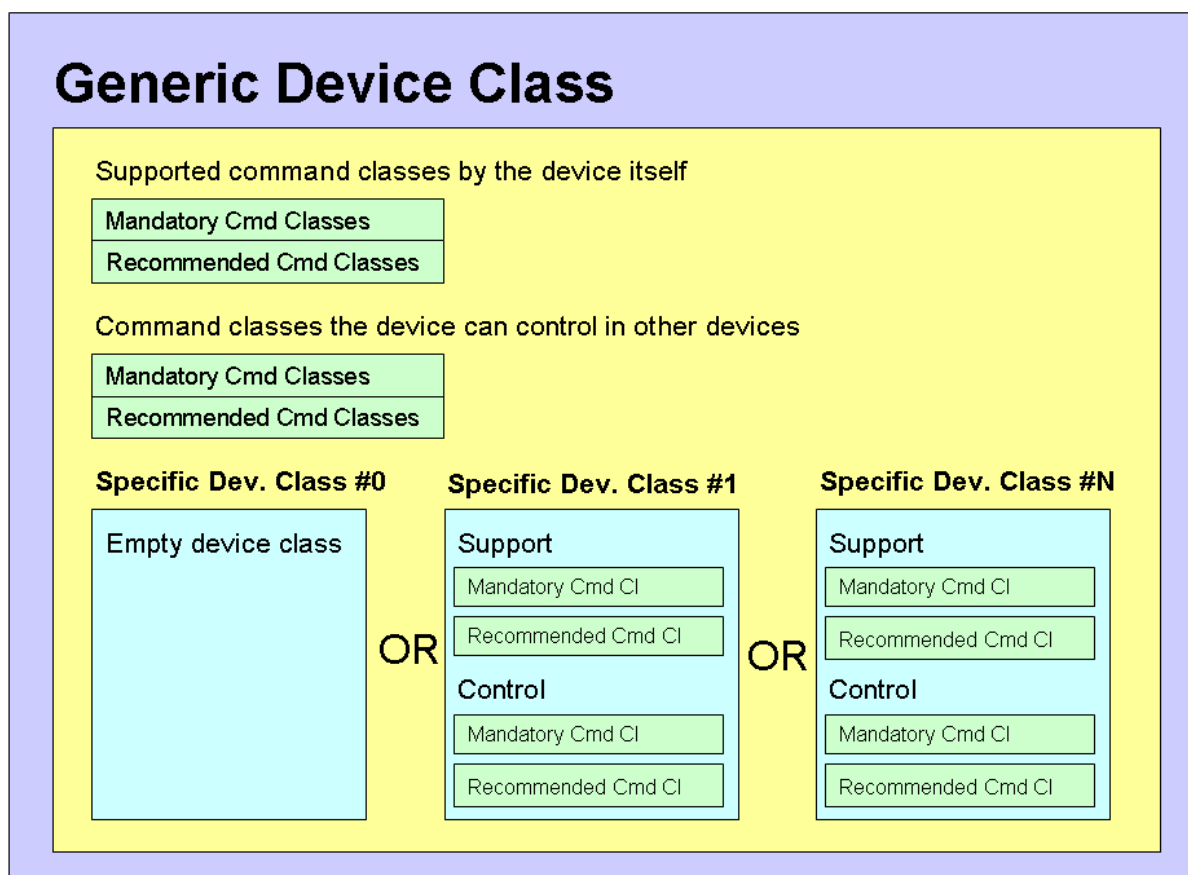


Figure 3. Generic and Specific Device Class Structure

Specific Device Classes are a more detailed description/definition of a device, and are based on a Generic Device Class. The Specific Device Class inherits all the mandatory commands from the Generic Device Class. An extra set of mandatory Command Classes can be specified for each Specific Device Class. The Specific Device Classes can also contain recommended Command Classes.

A Specific Device Class is therefore always a specialization of a Generic Device Class and the Specific Device Class always inherits all the commands that are specified for the Generic Device Class.

A number of Device Classes will be specified in this document. Devices will have to comply with one of the predefined Device Classes to ensure interoperability. This will be surveyed through the Z-Wave Certification Program.

The strategy of building Device Classes allows all devices originating from the same Generic Device Class to be controlled by the same controller. Some more specialized commands can however only be used by specialized controllers, that knows how to control the Specific Device Class.

All the Generic and Specific Device Classes that have been defined to date are described in Chapter 5.

3.2 Command Classes

This document describes all the commands that are used by devices when communicating with other devices. These commands are divided into functionally related groups called command classes. Each command class contains the commands associated with a given functionality.

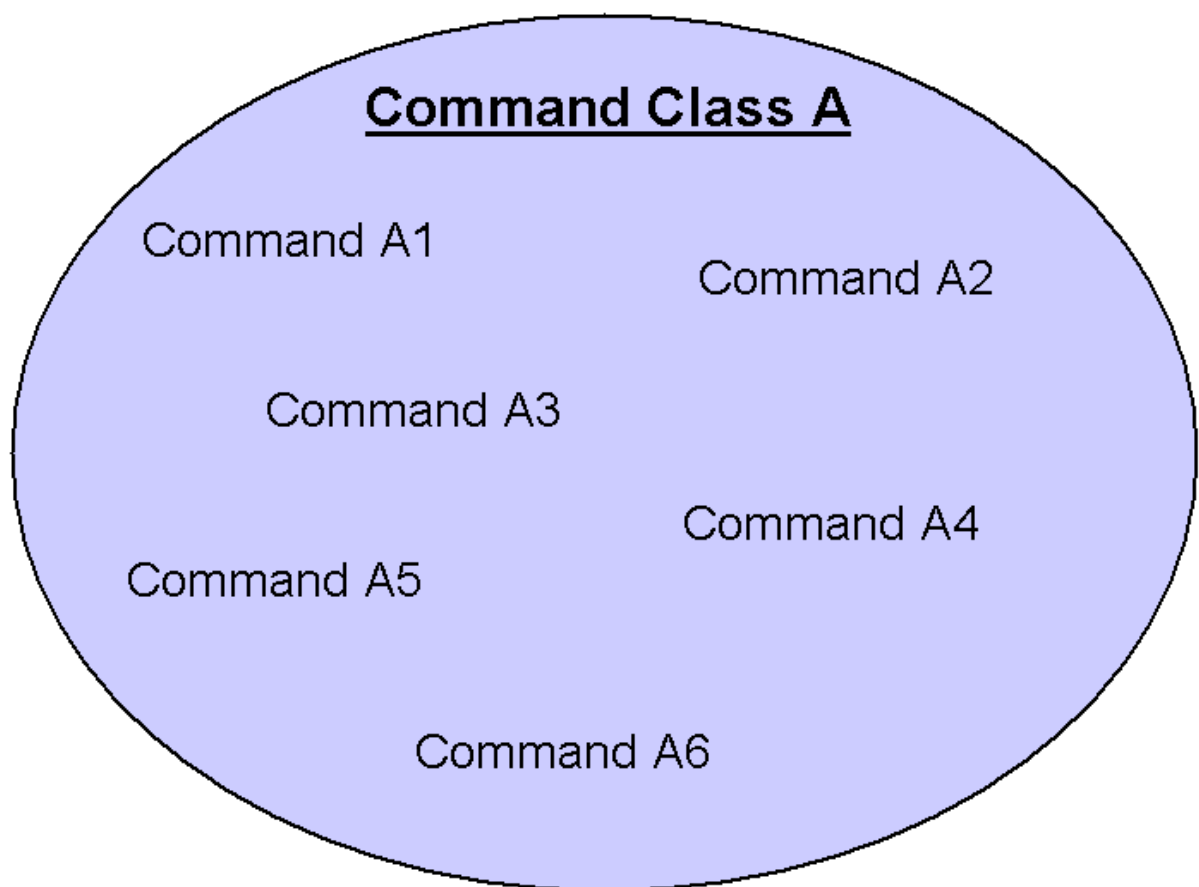


Figure 4. Command class and related commands

A detailed description of all the available command classes are given in [8]

NOTICE: It is mandatory to select and implement the device and command classes that accurately reflect the functionality available in device.

E.g. a dimmable light module must be implemented as a “Multilevel Switch Generic Device Class” and “Multilevel Power Switch Specific Device Class”.

3.3 Node Information Frame

The Node Information frame is used to inform other devices about node capabilities. The Node Information frame contains a structure with a protocol specific part that is handled by the Z-Wave protocol and an application specific part that is filled in by the application. The protocol specific part consists of a bit telling if the node is a continuously listening device, the Basic Device Class the device is based on etc. The application specific part consists of the Generic and Specific Device Class and the Command Classes that are supported and/or controlled by the device.

A Node Information frame will be sent to the controller when a node is to be included in the network, excluded from the network or upon request.

The Node information frame is shown below.

Byte descriptor \ bit number	7	6	5	4	3	2	1	0
Capability	Liste- ning	Z-Wave Protocol Specific Part						
Security	Opt. Func.	Z-Wave Protocol Specific Part						
Reserved	Z-Wave Protocol Specific Part							
Basic	Basic Device Class (Z-Wave Protocol Specific Part)							
Generic	Generic Device Class							
Specific	Specific Device Class							
NodeInfo[0]	Command Class 1							
...	...							
NodeInfo[n-1]	Command Class n							

Figure 5. Node Information frame format

The Z-Wave Protocol in a controller saves all the Node Information except the supported and controlled Command Classes when a node is included in the network. The reserved field is for future use. The Z-Wave protocol zero these fields and applications shall make no assumptions on the values of these fields nor perform processing based on their content.

3.3.1 Z-Wave Protocol Specific Part

The protocol specific part of the Node Information frame is handled by the Z-Wave protocol. This information is automatically inserted in the packet by the protocol layer when transferring data using the API.

Basic Device Class

The Basic Device Class field contains an identifier that identifies what Basic Device Class this node is based on and is set by the Z-Wave protocol. A detailed description of all available Basic Device Classes is given in Chapter 4.

3.3.2 Application Specific Part

The application specific part of the Node Information frame is handled by the application. The information must be in accordance with the defined classes to obtain interoperability.

Listening Flag

The Listening flag is used to indicate that the node is always listening if set. An always listening node must be powered continuously and reside on a fixed position in the installation. An always listening node is also included in the routing table to assist as router in the network. The routing table is static during normal operation and this is the reason an always listening node must not be moved around in the network. In case the Listening flag is cleared then the node is non-listening and this is typically used for battery operated nodes being asleep when the protocol is idle to prolong battery lifetime. The battery operated node is not included in the routing table and is not as a router in the network, but in some instances the node's position in the network is still determined, and stored by the protocol.

Optional Functionality Flag

The Optional Functionality flag is used to inform that this node supports other command classes than the mandatory for the selected generic/specific device class and that it needs to look at the supported command classes to fully control this device. The application must set the Optional Functionality flag in the packet.

Generic Device Class

The Generic Device Class field contains an identifier that identifies what Generic Device Class this node is part of and must be set by the application. For a detailed description of all available Generic Device Classes, refer to Chapter 5.

Specific Device Class

The Specific Device Class field specifies what Specific Device Class this application is part of and must be set by the application. For a detailed description of all available Specific Device Classes, refer to Chapter 5.

Command Class

The Command Class field specifies a list of Command Class identifiers and is set by the application. There can be from 1 to n different Command Classes included. Be aware of that n can differ depending on the library version used.

The list starts with Command Classes that describe functionality the device support itself. The mandatory Command Classes the device support itself must be shown in the list. The Basic Command Class is implicitly supported by all generic devices and is therefore not listed in the Node Information frame.

Optionally the list of Command Classes can be appended with a list of the functionality the device can control in other devices. Even though mandatory Command Classes are defined describing functionality the device must control in other devices, it is optional to include them in the list. To distinguish between the two lists of command classes the mark `COMMAND_CLASS_MARK` is used. The figure below shows how both lists are contained in the Node Information frame:

Byte descriptor \ bit number	7	6	5	4	3	2	1	0
NodeInfo[0]	Command Class 1 (Support)							
...	...							
NodeInfo[x-1]	Command Class x (Support)							
NodeInfo[x]	Command Class x+1 (COMMAND_CLASS_MARK)							
NodeInfo[x+1]	Command Class x+2 (Control)							
...	...							
NodeInfo[n-1]	Command Class n (Control)							

Figure 6. Support/control command class list format

In case the device have no supported command classes and only can control other devices, the list start with the identifier COMMAND_CLASS_MARK. A detailed description of all available Command Classes is given in Chapter 5.18.

4 BASIC DEVICE CLASSES

This chapter describes the Basic Device Classes used to identify the Z-Wave library used by the application for a given device. The valid Basic Device Classes are:

Basic Device Class	ZW_classcmd.h
Node is a portable controller	BASIC_TYPE_CONTROLLER
Node is a static controller	BASIC_TYPE_STATIC_CONTROLLER
Node is a slave	BASIC_TYPE_SLAVE
Node is a slave with routing capabilities	BASIC_TYPE_ROUTING_SLAVE

Table 1. Basic Device Class identifiers

Refer to ZW_classcmd.h source code file for the assigned Basic Device Class identifiers and a detailed description of the capabilities of the Basic Device's can be found in [6].

4.1 Portable Controller

The portable controller can be used to control other nodes in a Z-Wave network. The portable controller can freely be moved around but it should of course always remain within direct range of minimum one node in the Z-Wave network. An important feature is the ability to include other devices locally into the Z-Wave network. A typical portable controller application could be a remote control.

The Basic Device Class identifier for a portable controller is BASIC_TYPE_CONTROLLER.

It's mandatory that the application on a portable controller supports the following functionalities:

- Node address assignment
- Full routing support
- Controller replication of protocol related data

The mandatory supported functionalities are described in the following:

Node address assignment

The portable controller has the ability to include/exclude other controllers and slaves in the Z-Wave network. This functionality is application level initiated. The capability to include/exclude other devices requires it is a primary controller. Controllers included by a primary controller are named secondary controllers, or inclusion controllers in case the primary controller also has the SIS role in the network.

Full routing support

The portable controller shall support the full routing part of the Z-Wave protocol. This includes collecting Node Information, maintaining a routing table, creating routing lists and using routing lists for data transmissions. The controller will not assume a certain position in the network, but will try to discover which nodes can be reached directly and used for routing further out into the network.

If the controller is a primary or inclusions controller it is recommended to initiate a rediscovery when noticing routing problems in the networks. If a SUC / SIS is present in the network a device is recommended to call ZW_RequestNetworkUpdate once per day and to call ZW_RequestNetworkUpdate before configuring associations.

Controller replication of protocol related data

The portable controller must be able to copy node information, routing information and other protocol related data to another controller in the Z-Wave network. Both replication send and replication receive on protocol level should be supported.

SUC / SIS Capabilities

Inclusion controllers shall always try to enable an included controller as SIS, alternatively as SUC.

4.2 Static Controller

The static controller is tailored to controllers that are in a fixed position in the Z-Wave network. The static controller can serve as a receiver for e.g. sensors or other battery-operated devices that needs to send unsolicited reports to a controller. It can also be used in a system where the controller needs to know the status of each controlled device in the network. A system like this could be an internet gateway, which can be accessed remotely. By being able to learn and store the best route to all nodes in the system it can also significantly reduce the latency in larger systems. A static controller will typically be a secondary controller in a Z-Wave network, and can also take on the roles of SUC and SIS in the network.

The Basic Device Class identifier for a static controller is `BASIC_TYPE_STATIC_CONTROLLER`.

It's mandatory that the application on a static controller supports the following functionalities:

- Node address assignment
- Full routing support
- Router support
- Controller replication of protocol related data

The mandatory supported functionalities are described in the following:

Node address assignment

The portable controller has the ability to include/exclude other controllers and slaves in the Z-Wave network. This functionality is application level initiated. The capability to include/exclude other devices requires it is a primary controller. Controllers included by a primary controller are named secondary controllers, or inclusion controllers in case the primary controller also has the SIS role in the network.

Full routing support

The portable controller shall support the full routing part of the Z-Wave protocol. This includes collecting Node Information, maintaining a routing table, creating routing lists and using routing lists for data transmissions. The controller will not assume a certain position in the network, but will try to discover which nodes can be reached directly and used for routing further out into the network.

If the controller is a primary or inclusions controller it is recommended to initiate a rediscovery when noticing routing problems in the networks. If a SUC / SIS is present in the network a device is recommended to call `ZW_RequestNetworkUpdate` once per day and to call `ZW_RequestNetworkUpdate` before configuring associations.

Router support

The portable controller has the ability to act as router when a source node tries to reach a destination node out of direct range.

Controller replication of protocol related data

The portable controller must be able to copy node information, routing information and other protocol related data to another controller in the Z-Wave network. Both replication send and replication receive on protocol level should be supported.

SUC / SIS Capabilities

Every suitable static controller shall be capable to operate as a SUC. Inclusion controllers shall always try to enable an included controller as SIS, alternatively as SUC.

4.3 Slave

The slave is the simplest node available in a Z-Wave network. The slave node is unable to initiate transmission of data to other nodes in a Z-Wave network unless it is as a response to a request. A slave can only be mains powered. A typical slave application could be a light dimmer or router.

The Basic Device Class identifier for a slave is BASIC_TYPE_SLAVE.

It's mandatory that the application on a slave supports following functionalities:

- Node inclusion/exclusion
- Router support

The mandatory supported functionalities at application level are described in the following:

Node inclusion/exclusion

The slave must have the ability to be included/excluded in a Z-Wave network by a primary or inclusion controller.

All devices shall operate at normal transmit power when joining an existing network. All devices shall support to reduce transmit power in inclusion when asked by the inclusion controller.

Router support

The slave has the ability to act as router when a source node tries to reach a destination node out of direct range.

It is recommended that slave devices should ask to be rediscovered when multiple communication attempts have failed, the number of attempts can be application depended.

If a SUC / SIS is present in the network a device is recommended to call ZW_RequestNetworkUpdate once per day and to call ZW_RequestNetworkUpdate before configuring associations.

4.4 Routing Slave

The routing slave has the same Z-Wave protocol functionality as a slave, but in addition the node can initiate transmission of data to a limited number of other nodes in the Z-Wave network. A routing slave can both be mains or battery powered. A typical routing slave application could be a movement detector.

An enhanced slave has the same Z-Wave protocol functionality as a routing slave, but in addition the enhanced slave is configured with an external EEPROM for storing application data. An enhanced and routing slave can be mains or battery powered. A typical enhanced slave application could be a push button, motion detector (PIR), and temperature sensor.

The Basic Device Class identifier for a slave with routing capabilities is BASIC_TYPE_ROUTING_SLAVE.

It's mandatory that the routing slave supports following functionalities:

- Node inclusion/exclusion
- Router support

The mandatory supported functionalities are described in the following:

Node inclusion/exclusion

The routing slave has the ability to be included/excluded in a Z-Wave network using a primary or inclusion controller.

All devices shall operate at normal transmit power when joining a existing network. All devices shall support to reduce transmit power in inclusion when asked by the inclusion controller.

Router support

The mains powered routing slave has the ability to act as router when a source node tries to reach a destination node out of direct range. A battery powered routing slave will not act as router.

It is recommended that slave devices should ask to be rediscovered when multiple communication attempts have failed, the number of attempts can be application depended.

If a SUC / SIS is present in the network a device is recommended to call ZW_RequestNetworkUpdate once per day and to call ZW_RequestNetworkUpdate before configuring associations.

5 GENERIC AND SPECIFIC DEVICE CLASSES

This chapter describes the Generic Device Classes together with the accompanying Specific Device Classes. The Generic Device Classes forms the main functionality for a given Z-Wave devices. Each Generic Device Class has a number of Specific Device Classes to define different variants of the given Z-Wave device.

The description of each Generic Device Class contains a description of the main functionality. The Generic Device Classes contain a set of mandatory Command Classes. If a device needs to implement other commands than described in the Generic Device Classes, a Specific Device Class is needed. Each Generic Device Class has an identifier assigned to it.

The table below shows the current list of valid combinations of Generic and Specific Device Classes:

Generic Device Classes	Specific Device Classes
Alarm Sensor	Basic Routing Alarm Sensor
	Routing Alarm Sensor
	Basic Zensor Net Alarm Sensor
	Zensor Net Alarm Sensor
	Advanced Zensor Net Alarm Sensor
	Basic Routing Smoke Sensor
	Routing Smoke Sensor
	Basic Zensor Net Smoke Sensor
	Zensor Net Smoke Sensor
	Advanced Zensor Net Smoke Sensor
AV Control Point	Specific Device Class not used
	Satellite Receiver
	Satellite Receiver V2
	Doorbell
Binary Sensor	Specific Device Class not used
	Routing Binary Sensor
Binary Switch	Specific Device Class not used
	Binary Power Switch
	Binary Scene Switch
Display	Specific Device Class not used
	Simple Display
Entry Control	Specific Device Class not used
	Door Lock
	Advanced Door Lock
	Secure Keypad Door Lock
Meter	Specific Device Class not used
	Simple Meter
Multilevel Sensor	Specific Device Class not used
	Routing Multilevel Sensor
Multilevel Switch	Specific Device Class not used
	Multilevel Power Switch
	Multilevel Scene Switch
	Multiposition Motor
	Motor Control A
	Motor Control B
	Motor Control C
Non interoperable	
Pulse Meter	Specific Device Class not used

Generic Device Classes	Specific Device Classes
Remote Controller	Specific Device Class not used
	Portable Remote Controller
	Portable Scene Controller
	Portable Installer Tool
Remote Switch	Specific Device Class not used
	Binary Remote Switch
	Multilevel Remote Switch
	Binary Toggle Remote Switch
	Multilevel Toggle Remote Switch
Repeater Slave	Specific Device Class not used
	Basic Repeater Slave
Semi Interoperable	Specific Device Class not used
	Energy Production
Static Controller	Specific Device Class not used
	PC Controller
	Scene Controller
	Static Installer Tool
Thermostat	Specific Device Class not used
	Heating Thermostat
	General Thermostat
	General Thermostat V2
	Setback Schedule Thermostat
	Setback Thermostat
	Setpoint Thermostat
Toggle Switch	Specific Device Class not used
	Binary Toggle Switch
	Multilevel Toggle Switch
Ventilation	Specific Device Class not used
	Residential Heat Recovery Ventilation
Window Covering	Specific Device Class not used
	Simple Window Covering Control
Z/IP Gateway	Specific Device Class not used
	Z/IP Tunneling Gateway
	Advanced Z/IP Gateway
Z/IP Node	Specific Device Class not used
	Z/IP Tunneling Node
	Advanced Z/IP Node

Table 2. Valid combinations of Generic and Specific Device Classes

A given combination of a Generic/Specific Device Class can optionally support all command classes. However, the recommended command classes specified in the specific device classes are the most obvious for such a device class.

Although a number of functionalities are optional, it should be noted that if a function is supported, then all the commands of the corresponding Command Class must be supported.

The Generic Device Class identifiers are show in the table below:

Generic Device Classes	ZW_classcmd.h
Alarm Sensor	GENERIC_TYPE_SENSOR_ALARM
AV Control Point	GENERIC_TYPE_AV_CONTROL_POINT
Binary Sensor	GENERIC_TYPE_SENSOR_BINARY
Binary Switch	GENERIC_TYPE_SWITCH_BINARY
Display	GENERIC_TYPE_DISPLAY
Entry Control	GENERIC_TYPE_ENTRY_CONTROL
Meter	GENERIC_TYPE_METER
Multilevel Sensor	GENERIC_TYPE_SENSOR_MULTILEVEL
Multilevel Switch	GENERIC_TYPE_SWITCH_MULTILEVEL
Non interoperable	GENERIC_TYPE_NON_INTEROPERABLE
Pulse Meter	GENERIC_TYPE_METER_PULSE
Remote Controller	GENERIC_TYPE_GENERIC_CONTROLLER
Remote Switch	GENERIC_TYPE_SWITCH_REMOTE
Repeater Slave	GENERIC_TYPE_REPEATER_SLAVE
Semi Interoperable	GENERIC_TYPE_SEMI_INTEROPERABLE
Static Controller	GENERIC_TYPE_STATIC_CONTROLLER
Thermostat	GENERIC_TYPE_THERMOSTAT
Toggle Switch	GENERIC_TYPE_SWITCH_TOGGLE
Ventilation	GENERIC_TYPE_VENTILATION
Window Covering	GENERIC_TYPE_WINDOW_COVERING
Z/IP Gateway	GENERIC_TYPE_ZIP_GATEWAY
Z/IP Node	GENERIC_TYPE_ZIP_NODE

Table 3. Generic Device Class identifiers

Refer to ZW_classcmd.h source code file for the assigned Generic Device Class identifiers. In ZW_classcmd.h additional constants and variables are defined to support the implementation.

Regarding the Specific Device Class identifiers refer to the following sections describing the valid combinations of Generic and Specific Device Classes.

NOTICE: The following device classes are considered legacy device classes that are superseded by newer Z-Wave specifications:

- Toggle Switch Generic Device Class
- Binary Toggle Remote Switch Specific Device Class
- Multilevel Toggle Remote Switch Specific Device Class
- Window Covering Generic Device Class
- Chimney Fan Device Class [4]
- Thermostat Heating Specific Device Class [5]
- General Thermostat Specific Device Class
- Satellite Receiver Specific Device Class

New Z-Wave enabled devices should not be based on these device classes.
See in the corresponding sections below for further details.

5.1 Alarm Sensor Generic Device Class

The Alarm Sensor Generic Device Class supports a range of alarm sensors, such as smoke, CO, CO₂, flood, heat, etc.

The Alarm Sensor Generic Device Class identifier is equal to `GENERIC_TYPE_SENSOR_ALARM` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_ROUTING_SLAVE`.

5.1.1 Mandatory Command Classes to Support

The generic device must support the following command classes:

- Basic Command Class

5.1.1.1 Basic Command Class Implementation

Refer to the specific device classes.

5.1.2 Mandatory Command Classes to Control

No mandatory command classes to control functionality in other devices from the generic device.

5.1.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Alarm Sensor Generic Device Class:

Table 1, Specific Device Class identifiers for the Alarm Sensor Generic Device Class

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Basic Routing Alarm Sensor	SPECIFIC_TYPE_BASIC_ROUTING_ALARM_SENSOR
Routing Alarm Sensor	SPECIFIC_TYPE_ROUTING_ALARM_SENSOR
Basic Zensor Net Alarm Sensor	SPECIFIC_TYPE_BASIC_ZENSOR_NET_ALARM_SENSOR
Zensor Net Alarm Sensor	SPECIFIC_TYPE_ZENSOR_NET_ALARM_SENSOR
Advanced Zensor Net Alarm Sensor	SPECIFIC_TYPE_ADV_ZENSOR_NET_ALARM_SENSOR
Basic Routing Smoke Sensor	SPECIFIC_TYPE_BASIC_ROUTING_SMOKE_SENSOR
Routing Smoke Sensor	SPECIFIC_TYPE_ROUTING_SMOKE_SENSOR
Basic Zensor Net Smoke Sensor	SPECIFIC_TYPE_BASIC_ZENSOR_NET_SMOKE_SENSOR
Zensor Net Smoke Sensor	SPECIFIC_TYPE_ZENSOR_NET_SMOKE_SENSOR
Advanced Zensor Net Smoke Sensor	SPECIFIC_TYPE_ADV_ZENSOR_NET_SMOKE_SENSOR

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.1.3.1 No Specific Device Class defined

This option is selected in case the generic level is appropriate for defining the device.

5.1.3.1.1 Mandatory Command Classes to Support

No mandatory command classes are specified to access functionality in the specific device itself.

5.1.3.1.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set and Basic Get = The Basic Set and basic Get can freely be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Set and Get must be mapped within the same command class. In case the manufacture deems that no relevant commands are available for mapping the command a device receiving these commands must ignore them.

Basic Report = The Basic Report can be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Report must be mapped within the same command class as the Basic Get and must be in accordance with the mapped Get. In case the manufacture deems that no relevant commands are available for mapping the Basic Get command a device is not allowed to respond with a Basic Report under any circumstances.

5.1.3.2 Basic Routing Alarm Sensor Specific Device Class

A Basic Routing Alarm Sensor Specific Device Class inherits all the commands from the Alarm Sensor Generic Device Class.

The Basic Routing Alarm Sensor Specific Device Class supports simple battery operated alarm sensor devices based on typically a routing slave or static controller. The device supports the following functionalities:

- Alarm Sensor (mandatory)
- Association (mandatory)
- Manufacturer Specific (mandatory)
- Version (mandatory)

Alarm Sensor

The alarm sensor is used to signal an event in case the detector detects an alarm.

Association

The alarm sensor sends the detected alarm to the defined associations. The alarm can alternatively be broadcasted in case no associations are defined. In both cases are the Alarm Sensor Command Class used for this purpose.

5.1.3.2.1 Mandatory Command Classes to Support

In addition to the inherited command classes from the generic device the following must also be supported:

- Alarm Sensor Command Class
- Association Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.1.3.2.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set	=	Cannot be mapped because the mapped command class doesn't have a Set command. A device receiving this command must simply ignore it.
Basic Get	=	Alarm Sensor Get
Basic Report	=	Alarm Sensor Report (Sensor State field is mapped into the Basic Reports Value field)

5.1.3.2.2 Recommended Command Classes to Support

No recommended command classes supported by the device.

5.1.3.2.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Alarm Sensor Command Class

5.1.3.2.4 Recommended Command Classes to Control

No recommended command classes to control functionality in other devices.

5.1.3.3 Routing Alarm Sensor Specific Device Class

A Routing Alarm Sensor Specific Device Class inherits all the commands from the Alarm Sensor Generic Device Class.

The Routing Alarm Sensor Specific Device Class supports simple battery operated alarm sensor devices based on typically a routing slave or static controller. The device supports the following functionalities:

- Alarm Sensor (mandatory)
- Association (mandatory)
- Battery (mandatory)
- Manufacturer Specific (mandatory)
- Version (mandatory)

Alarm Sensor

The alarm sensor is used to signal an event in case the detector detects an alarm.

Association

The alarm sensor sends the detected alarm to the defined associations. The alarm can alternatively be broadcasted in case no associations are defined. In both cases are the Alarm Sensor Command Class used for this purpose.

5.1.3.3.1 Mandatory Command Classes to Support

In addition to the inherited command classes from the generic device the following must also be supported:

- Alarm Sensor Command Class
- Association Command Class
- Battery Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.1.3.3.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set	=	Cannot be mapped because the mapped command class doesn't have a Set command. A device receiving this command must simply ignore it.
Basic Get	=	Alarm Sensor Get
Basic Report	=	Alarm Sensor Report (Sensor State field is mapped into the Basic Reports Value field)

5.1.3.3.2 Recommended Command Classes to Support

The Silence Alarm command class is recommended in case the sensor alarm needs to be silenced.

5.1.3.3.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Alarm Sensor Command Class

5.1.3.3.4 Recommended Command Classes to Control

No recommended command classes to control functionality in other devices.

5.1.3.4 Basic Zensor Net Alarm Sensor Specific Device Class

A Basic Zensor Net Alarm Sensor Specific Device Class inherits all the commands from the Alarm Sensor Generic Device Class.

The Basic Zensor Net Alarm Sensor Specific Device Class supports battery operated Alarm Sensor devices in a Zensor Net. The devices can not be configured as Frequently Listening Routing Slaves (FLIRS). Notice that Zensor Net nodes don't support the node information frame because the broadcast forwarding mechanism is the only transport mechanism available. The device supports the following functionalities:

- Alarm Sensor (mandatory)
- Manufacturer Specific (mandatory)
- Version (mandatory)

Alarm Sensor

The alarm sensor is used to signal an event in case the detector detects an alarm. The alarm is distributed to the other nodes in the Zensor Net using broadcast forwarding.

5.1.3.4.1 Mandatory Command Classes to Support

In addition to the inherited command classes from the generic device the following must also be supported:

- Alarm Sensor Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.1.3.4.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set	=	Cannot be mapped because the mapped command class doesn't have a Set command. A device receiving this command must simply ignore it.
Basic Get	=	Alarm Sensor Get
Basic Report	=	Alarm Sensor Report (Sensor State field is mapped into the Basic Reports Value field)

5.1.3.4.2 Recommended Command Classes to Support

The Silence Alarm command class is recommended in case the sensor alarm needs to be silenced.

5.1.3.4.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Alarm Sensor Command Class

5.1.3.4.4 Recommended Command Classes to Control

No recommended command classes to control functionality in other devices.

5.1.3.4.5 Recommended Command Classes to Control

No recommended command classes to control functionality in other devices.

5.1.3.5 Zensor Net Alarm Sensor Specific Device Class

A Zensor Net Alarm Sensor Specific Device Class inherits all the commands from the Alarm Sensor Generic Device Class.

The Zensor Net Alarm Sensor Specific Device Class supports battery operated Alarm Sensor devices in a Zensor Net. The devices can not be configured as Frequently Listening Routing Slaves (FLiRS). Notice that Zensor Net nodes don't support the node information frame because the broadcast forwarding mechanism is the only transport mechanism available. The device supports the following functionalities:

- Alarm Sensor (mandatory)
- Battery (mandatory)
- Manufacturer Specific (mandatory)
- Version (mandatory)

Alarm Sensor

The alarm sensor is used to signal an event in case the detector detects an alarm. The alarm is distributed to the other nodes in the Zensor Net using broadcast forwarding.

5.1.3.5.1 Mandatory Command Classes to Support

In addition to the inherited command classes from the generic device the following must also be supported:

- Alarm Sensor Command Class
- Battery Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.1.3.5.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set	=	Cannot be mapped because the mapped command class doesn't have a Set command. A device receiving this command must simply ignore it.
Basic Get	=	Alarm Sensor Get
Basic Report	=	Alarm Sensor Report (Sensor State field is mapped into the Basic Reports Value field)

5.1.3.5.2 Recommended Command Classes to Support

The Silence Alarm command class is recommended in case the sensor alarm needs to be silenced.

5.1.3.5.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Alarm Sensor Command Class

5.1.3.5.4 Recommended Command Classes to Control

No recommended command classes to control functionality in other devices.

5.1.3.6 Advanced Zensor Net Alarm Sensor Specific Device Class

An Advanced Zensor Net Alarm Sensor Specific Device Class inherits all the commands from the Alarm Sensor Generic Device Class.

The Advanced Zensor Net Alarm Sensor Specific Device Class supports battery operated Alarm Sensor devices in a Zensor Net. The device is also configured as Frequently Listening Routing Slaves (FLiRS) enabling communication with a classic Z-Wave network. Notice that the FLiRS functionality enables support of the node information frame. The device supports the following functionalities:

- Alarm Sensor (mandatory)
- Association (mandatory)
- Battery (mandatory)
- Manufacturer Specific (mandatory)
- Version (mandatory)

Alarm Sensor

The alarm sensor is used to signal an event in case the detector detects an alarm. The alarm is distributed to the other nodes in the Zensor Net using broadcast forwarding.

Association

The alarm sensor sends the detected alarm etc. to the defined associations. The alarm can alternatively be broadcasted in case no associations are defined. In both cases are the Alarm Sensor Command Class used for this purpose.

5.1.3.6.1 Mandatory Command Classes to Support

In addition to the inherited command classes from the generic device the following must also be supported:

- Association Command Class
- Alarm Sensor Command Class
- Battery Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.1.3.6.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set	=	Cannot be mapped because the mapped command class doesn't have a Set command. A device receiving this command must simply ignore it.
Basic Get	=	Alarm Sensor Get
Basic Report	=	Alarm Sensor Report (Sensor State field is mapped into the Basic Reports Value field)

5.1.3.6.2 Recommended Command Classes to Support

The Silence Alarm command class is recommended in case the sensor alarm needs to be silenced.

5.1.3.6.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Alarm Sensor Command Class

5.1.3.6.4 Recommended Command Classes to Control

No recommended command classes to control functionality in other devices.

5.1.3.7 Basic Routing Smoke Sensor Specific Device Class

A Basic Routing Smoke Sensor Specific Device Class inherits all the commands from the Alarm Sensor Generic Device Class.

The Basic Routing Smoke Sensor Specific Device Class supports simple battery operated smoke sensor devices based on typically a routing slave or static controller. The device supports the following functionalities:

- Alarm Sensor (mandatory)
- Association (mandatory)
- Manufacturer Specific (mandatory)
- Version (mandatory)

Alarm Sensor

The smoke sensor is used to signal an event in case the detector detects smoke. Only Sensor Type equal to 'Smoke Sensor=0x01' is allowed.

Association

The smoke sensor sends the detected alarm to the defined associations. The alarm can alternatively be broadcasted in case no associations are defined. In both cases are the Alarm Sensor Command Class used for this purpose.

5.1.3.7.1 Mandatory Command Classes to Support

In addition to the inherited command classes from the generic device the following must also be supported:

- Alarm Sensor Command Class
- Association Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.1.3.7.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set	=	Cannot be mapped because the mapped command class doesn't have a Set command. A device receiving this command must simply ignore it.
Basic Get	=	Alarm Sensor Get
Basic Report	=	Alarm Sensor Report (Sensor State field is mapped into the Basic Reports Value field)

5.1.3.7.2 Recommended Command Classes to Support

No recommended command classes supported by the device.

5.1.3.7.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Alarm Sensor Command Class

5.1.3.7.4 Recommended Command Classes to Control

No recommended command classes to control functionality in other devices.

5.1.3.8 Routing Smoke Sensor Specific Device Class

A Routing Smoke Sensor Specific Device Class inherits all the commands from the Alarm Sensor Generic Device Class.

The Simple Meter Specific Device Class supports simple battery operated smoke sensor devices based on typically a routing slave or static controller. The device supports the following functionalities:

- Alarm Sensor (mandatory)
- Association (mandatory)
- Battery (mandatory)
- Manufacturer Specific (mandatory)
- Version (mandatory)

Alarm Sensor

The smoke sensor is used to signal an event in case the detector detects smoke. Only Sensor Type equal to 'Smoke Sensor=0x01' is allowed.

Association

The smoke sensor sends the detected alarm to the defined associations. The alarm can alternatively be broadcasted in case no associations are defined. In both cases are the Alarm Sensor Command Class used for this purpose.

5.1.3.8.1 Mandatory Command Classes to Support

In addition to the inherited command classes from the generic device the following must also be supported:

- Alarm Sensor Command Class
- Association Command Class
- Battery Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.1.3.8.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set	=	Cannot be mapped because the mapped command class doesn't have a Set command. A device receiving this command must simply ignore it.
Basic Get	=	Alarm Sensor Get
Basic Report	=	Alarm Sensor Report (Sensor State field is mapped into the Basic Reports Value field)

5.1.3.8.2 Recommended Command Classes to Support

The Silence Alarm command class is recommended in case the sensor alarm needs to be silenced.

5.1.3.8.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Alarm Sensor Command Class

5.1.3.8.4 Recommended Command Classes to Control

No recommended command classes to control functionality in other devices.

5.1.3.9 Basic Zensor Net Smoke Sensor Specific Device Class

A Basic Zensor Net Smoke Sensor Specific Device Class inherits all the commands from the Alarm Sensor Generic Device Class.

The Basic Zensor Net Smoke Sensor Specific Device Class supports battery operated Smoke Sensor devices in a Zensor Net. The devices can not be configured as Frequently Listening Routing Slaves (FLIRS). Notice that Zensor Net nodes don't support the node information frame because the broadcast forwarding mechanism is the only transport mechanism available. The device supports the following functionalities:

- Alarm Sensor (mandatory)
- Manufacturer Specific (mandatory)
- Version (mandatory)

Alarm Sensor

The smoke sensor is used to signal an event in case the detector detects smoke. Only Sensor Type equal to 'Smoke Sensor=0x01' is allowed. The alarm is distributed to the other nodes in the Zensor Net using broadcast forwarding.

5.1.3.9.1 Mandatory Command Classes to Support

In addition to the inherited command classes from the generic device the following must also be supported:

- Alarm Sensor Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.1.3.9.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set	=	Cannot be mapped because the mapped command class doesn't have a Set command. A device receiving this command must simply ignore it.
Basic Get	=	Alarm Sensor Get
Basic Report	=	Alarm Sensor Report (Sensor State field is mapped into the Basic Reports Value field)

5.1.3.9.2 Recommended Command Classes to Support

The Silence Alarm command class is recommended in case the Alarm Sensor needs to be silenced.

5.1.3.9.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Alarm Sensor Command Class

5.1.3.9.4 Recommended Command Classes to Control

No recommended command classes to control functionality in other devices.

5.1.3.9.5 Recommended Command Classes to Control

No recommended command classes to control functionality in other devices.

5.1.3.10 Zensor Net Smoke Sensor Specific Device Class

A Zensor Net Smoke Sensor Specific Device Class inherits all the commands from the Alarm Sensor Generic Device Class.

The Zensor Net Smoke Sensor Specific Device Class supports battery operated Smoke Sensor devices in a Zensor Net. The devices can not be configured as Frequently Listening Routing Slaves (FLiRS). Notice that Zensor Net nodes don't support the node information frame because the broadcast forwarding mechanism is the only transport mechanism available. The device supports the following functionalities:

- Alarm Sensor (mandatory)
- Battery (mandatory)
- Manufacturer Specific (mandatory)
- Version (mandatory)

Alarm Sensor

The smoke sensor is used to signal an event in case the detector detects smoke. Only Sensor Type equal to 'Smoke Sensor=0x01' is allowed. The alarm is distributed to the other nodes in the Zensor Net using broadcast forwarding.

5.1.3.10.1 Mandatory Command Classes to Support

In addition to the inherited command classes from the generic device the following must also be supported:

- Alarm Sensor Command Class
- Battery Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.1.3.10.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set	=	Cannot be mapped because the mapped command class doesn't have a Set command. A device receiving this command must simply ignore it.
Basic Get	=	Alarm Sensor Get
Basic Report	=	Alarm Sensor Report (Sensor State field is mapped into the Basic Reports Value field)

5.1.3.10.2 Recommended Command Classes to Support

The Silence Alarm command class is recommended in case the sensor alarm needs to be silenced.

5.1.3.10.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Alarm Sensor Command Class

5.1.3.10.4 Recommended Command Classes to Control

No recommended command classes to control functionality in other devices.

5.1.3.11 Advanced Zensor Net Smoke Sensor Specific Device Class

An Advanced Zensor Net Smoke Sensor Specific Device Class inherits all the commands from the Alarm Sensor Generic Device Class.

The Advanced Zensor Net Smoke Sensor Specific Device Class supports battery operated Smoke Sensor devices in a Zensor Net. The device is also configured as Frequently Listening Routing Slaves (FLIRS) enabling communication with a classic Z-Wave network. Notice that the FLIRS functionality enables support of the node information frame. The device supports the following functionalities:

- Alarm Sensor (mandatory)
- Association (mandatory)
- Battery (mandatory)
- Manufacturer Specific (mandatory)
- Version (mandatory)

Alarm Sensor

The smoke sensor is used to signal an event in case the detector detects smoke. Only Sensor Type equal to 'Smoke Sensor=0x01' is allowed. The alarm is distributed to the other nodes in the Zensor Net using broadcast forwarding.

Association

The smoke sensor sends the detected alarm etc. to the defined associations. The alarm can alternatively be broadcasted in case no associations are defined. In both cases are the Alarm Sensor Command Class used for this purpose.

5.1.3.11.1 Mandatory Command Classes to Support

In addition to the inherited command classes from the generic device the following must also be supported:

- Association Command Class
- Alarm Sensor Command Class
- Battery Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.1.3.11.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set	=	Cannot be mapped because the mapped command class doesn't have a Set command. A device receiving this command must simply ignore it.
Basic Get	=	Alarm Sensor Get
Basic Report	=	Alarm Sensor Report (Sensor State field is mapped into the Basic Reports Value field)

5.1.3.11.2 Recommended Command Classes to Support

The Silence Alarm command class is recommended in case the sensor alarm needs to be silenced.

5.1.3.11.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Alarm Sensor Command Class

5.1.3.11.4 Recommended Command Classes to Control

No recommended command classes to control functionality in other devices.

5.2 Binary Switch Generic Device Class

The Binary Switch Generic device class supports all devices that need to switch between two states e.g. turn power on/off or other “simple” devices.

The Binary Switch Device Class identifier is equal to `GENERIC_TYPE_SWITCH_BINARY` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_SLAVE` or `BASIC_TYPE_ROUTING_SLAVE`.

The device supports the following functionalities:

- Binary Switch (mandatory)

The supported functionalities are described in the following:

Binary Switch

The Binary Switch shall be able to receive commands instructing the device to change state between power on/off or open/close or similar states.

5.2.1 Mandatory Command Classes to Support

The following command classes to support are defined on the generic device class level:

- Basic Command Class
- Binary Switch Command Class

5.2.2 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the generic device class level.

5.2.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Binary Switch Generic Device Class:

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Binary Power Switch	SPECIFIC_TYPE_POWER_SWITCH_BINARY
Binary Scene Switch	SPECIFIC_TYPE_SCENE_SWITCH_BINARY

Table 4. Specific Device Class identifiers for the Binary Switch Generic Device Class

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.2.3.1 No Specific Device Class defined

This option is selected in case the generic level is appropriate for defining the device.

5.2.3.1.1 Mandatory Command Classes to Support

No mandatory command classes to support are defined on the specific device class level.

5.2.3.1.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

- Basic Set = Binary Switch Set
- Basic Get = Binary Switch Get
- Basic Report = Binary Switch Report

5.2.3.1.2 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.2.3.2 Binary Power Switch Specific Device Class

A Binary Power Switch device inherits all the commands from the Binary Switch Device Class. Apart from these Command Classes it also supports the All Switch Command Class.

The Binary Power Switch Device Class supports devices that need to turn power on/off. The device supports the following functionalities:

- All switch (mandatory)
- Protection (recommended)
- Clock (recommended)

The supported functionalities are described in the following:

All Switch

The all switch functionality is mandatory. This functionality allows the switch to take part in all on/all off operations.

Protection

The Binary Power Switch may optionally support the Protection functionality. If the protection functionality is supported, then the device shall be able to handle protection related commands.

Clock

The Binary Power Switch may optionally support the clock functionality. If the clock functionality is supported, then the Binary Power Switch shall be able to handle clock related commands.

5.2.3.2.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- All Switch Command Class

5.2.3.2.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

- Basic Set = Binary Switch Set
- Basic Get = Binary Switch Get
- Basic Report = Binary Switch Report

5.2.3.2.2 Recommended Command Classes to Support

The following Command Classes can optionally be supported:

- Clock Command Class
- Protection command class

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities.

5.2.3.2.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.2.3.3 Binary Scene Switch Specific Device Class

A Binary Scene Switch device inherits all the mandatory command classes from the Binary Switch Generic Device Class.

The Binary Scene Switch Specific Device Class supports devices that need to turn power on/off and take part in scene operations. The device supports the following functionalities:

- Scene Activation (mandatory)
- Scene Actuator Configuration (mandatory)
- All Switch (mandatory)
- Manufacturer Specific (mandatory)

Some of the supported functionalities are described in the following:

Scene Activation

Used to control scenes in a binary scene switch.

Scene Actuator Configuration

Used to configure scenes in a binary scene switch.

All Switch

This functionality allows the switch to take part in all on/all off operations.

5.2.3.3.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Scene Activation Command Class
- Scene Actuator Configuration Command Class
- All Switch Command Class
- Manufacturer Specific Command Class

5.2.3.3.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

- Basic Set = Binary Switch Set
- Basic Get = Binary Switch Get
- Basic Report = Binary Switch Report

5.2.3.3.2 Recommended Command Classes to Support

The version command class can be used to retrieve the command class revisions etc. supported by the device.

No recommended command classes to support are defined on the specific device class level.

5.2.3.3.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.2.3.3.4 Recommended Command Classes to Control

No recommended command classes to control with are defined on the specific device class level.

5.3 Remote Controller Generic Device Class

The Remote Controller Generic Device Class is a device that can be used to control other nodes in a Z-Wave network. There are no restrictions on the placement of the controller i.e. it can be moved around.

Remote Controller Device Class identifier is equal to `GENERIC_TYPE_GENERIC_CONTROLLER` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_CONTROLLER`.

This Device Class can be used to realize remote controllers. Very simple remote controllers can be implemented using the Basic Command Class to control other devices. This Device Class is tailored to controllers that may be moved around but it should of course always remain within range of minimum one device in the Z-Wave network.

5.3.1 Mandatory Command Classes to Control

The following command classes to control with are defined on the generic device class level:

- Basic Command Class

In case a remote controller device is configured to control an unknown device it shall use basic command to control this device.

5.3.2 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Remote Controller Generic Device Class:

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Portable Remote Controller	SPECIFIC_TYPE_PORTABLE_REMOTE_CONTROLLER
Portable Scene Controller	SPECIFIC_TYPE_PORTABLE_SCENE_CONTROLLER
Portable Installer Tool	SPECIFIC_TYPE_PORTABLE_INSTALLER_TOOL

Table 5. Specific Device Class identifiers for the Remote Controller Generic Device Class

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.3.2.1 Portable Remote Controller Specific Device Class

The Portable Remote Controller Specific Device Class can be used to control other nodes in a Z-Wave network. There are no restrictions on the placement of the controller i.e. it can be moved around. The Portable Controller supports the following functionalities:

- Battery (recommended)
- Clock (recommended)

This Device Class can be used to realize e.g. remote controllers. This Device Class is tailored to controllers that may be moved around but it should of course always remain within range of minimum one device in the network.

The supported functionalities are described in the following:

Battery

The Portable Controller may optionally support the battery functionality. If the battery functionality is supported, then the Portable Controller shall be able to handle battery related commands.

Clock

The Portable Controller may optionally support the clock functionality. If the clock functionality is supported, then the Portable Controller shall be able to handle clock related commands.

5.3.2.1.1 Mandatory Command Classes to Support

No mandatory command classes to support are defined on the specific device class level.

5.3.2.1.2 Recommended Command Classes to Support

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities.

The following Command Classes can optionally be supported:

- Clock Command Class
- Battery Command Class.

5.3.2.1.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.3.2.1.4 Recommended Command Classes to Control

No recommended command classes to control with are defined on the specific device class level.

5.3.2.2 Portable Scene Controller Specific Device Class

The Portable Scene Controller Specific Device Class can be used to configure and control scenes in other nodes in a Z-Wave network. There are no restrictions on the placement of the Portable Scene Controller i.e. it can be moved around.

The Portable Scene Controller supports the following functionality

- Association (mandatory - support)
- Scene Activation (mandatory - control)
- Scene Controller Configuration (mandatory - support)
- Manufacturer Specific (mandatory - support)
- Basic (mandatory – control)
- Scene Actuator Configuration (optional - control)
- Controller replication of the application part (recommended)

Some of the supported functionalities are described in the following:

Associations

Associations is used to organize nodes in different groups allowing the device to identify the nodes by a group identifier. The groups can also be copied to other devices.

Scene Activation

Is used to activate scenes in other devices and indicate the active scene on the portable scene-controlling device. The Basic command class to ensure backwards compatibility must control nodes in a group not supporting scene activation.

Scene Controller Configuration

Scene controller configuration is used to bind a physical identifier e.g. assigning a push button to a group of associations and the wanted scene settings.

Controller replication of the application part

The Portable Scene Controller must respond with ZW_ReplicationReceiveComplete in case it receives the replication command class otherwise will the replication of protocol data fail. Optionally can the Portable Scene Controller interpret the data transferred by the replication command class or simply skip the data.

5.3.2.2.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Association Command Class
- Scene Controller Configuration Command Class
- Manufacturer Specific Command Class

5.3.2.2.2 Recommended Command Classes to Support

The version command class can be used to retrieve the command class revisions etc. supported by the device.

5.3.2.2.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Scene Activation Command Class

5.3.2.2.4 Recommended Command Classes to Control

A number of command classes are recommended to enable a portable scene controller to exchange configuration information. The following command classes can optionally be used to control other devices:

- Association Command Class
- Scene Controller Configuration Command Class
- Scene Actuator Configuration Command Class
- All Switch Command Class

5.3.2.3 Portable Installer Tool Specific Device Class

The Portable Installer Tool Specific Device Class could be used to realize a controller which is primarily used to set-up a network and configure all nodes in the network. The installer tool is not envisioned to be used as a day-to-day controller. The controller is portable it should therefore be noticed that the SUC/SIS functionality must not be enable for this device. The device must however be able to appoint the SUC/SIS role to another node in the network.

5.3.2.3.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Controller Replication Command Class
- Multi Command Command Class
- Version Command Class
- Manufacturer Specific Command Class

5.3.2.3.2 Recommended Command Classes to Support

The following command classes are recommended to be supported:

- Clock Command Class
- Battery Command Class

5.3.2.3.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Multi Channel Command Class - Version 2
- Controller Replication Command Class
- Association Command Class
- Multi Channel Association Command Class - Version 2
- Wake Up Command Class
- Configuration Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.3.2.3.4 Recommended Command Classes to Control

A number of command classes are recommended in order to retrieve information and configure advanced devices:

- Binary Switch Command Class
- Multilevel Switch Command Class
- Battery Command Class
- Scene Controller Configuration Command Class
- Scene Actuator Configuration Command Class
- All Switch Command Class
- Node Naming and Location Command Class
- Grouping Name Command Class
- Protection Command Class
- Powerlevel Command Class

5.4 Static Controller Generic Device Class

The Static Controller Generic Device Class contains much of the same functions as the Remote Control, but has some restrictions and some extra functionality. The main limitation is that the controller is expected to be stationary. Based on the routing table it is therefore able to tell slave nodes how it can be reached.

The Static Controller Device Class identifier is equal to `GENERIC_TYPE_STATIC_CONTROLLER` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_STATIC_CONTROLLER`.

This Device Class can be used to realize Static Controllers. This Device Class is tailored to controllers that are in a fixed position in the network. The Static Controller can serve as a receiver for e.g. sensors or other battery-operated devices that needs to send unsolicited reports to a controller. It can also be used in a system where the controller needs to know the status of each controlled device in the network. A system like this could be a residential gateway, which can be accessed remotely. By being able to learn and store the best route to all nodes in the system it can also significantly reduce the latency in larger systems.

5.4.1 Recommended Command Classes to Support

The following command classes can optionally be supported:

- Basic Command Class

5.4.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set and Basic Get = The Basic Set and basic Get can freely be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Set and Get must be mapped within the same command class. In case the manufacture deems that no relevant commands are available for mapping the command a device receiving these commands must ignore them.

Basic Report = The Basic Report can be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Report must be mapped within the same command class as the Basic Get and must be in accordance with the mapped Get. In case the manufacture deems that no relevant commands are available for mapping the Basic Get command a device is not allowed to respond with a Basic Report under any circumstances.

5.4.2 Mandatory Command Classes to Control

The following command classes to control with are defined on the generic device class level:

- Basic Command Class

In case a static controller device is configured to control an unknown device it shall use basic command to control this device.

5.4.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Static Controller Generic Device Class:

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
PC Controller	SPECIFIC_TYPE_PC_CONTROLLER
Scene Controller	SPECIFIC_TYPE_SCENE_CONTROLLER
Static Installer Tool	SPECIFIC_TYPE_STATIC_INSTALLER_TOOL

Table 6. Specific Device Class identifiers for the Static Controller Generic Device Class

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.4.3.1 PC Controller Specific Device Class

The PC Controller Specific Device Class can be used to control and monitor other nodes in a Z-Wave network. The PC controller is a controller based on a PC with a graphical user interface. The controller is a static controller.

The PC controller supports the following functionality

- Static Update Controller (recommended)
- Controller replication of the application part (recommended)
- Clock (recommended)

This Device Class is tailored to controllers that have a fixed position and is always powered can take advantage of the possibilities that a powerful PC platform offers in terms of processing power and graphics. Only the controller needs to be powered to fulfill the SUC role in the network, so the PC can be turned off when not used.

The supported functionalities are described in the following:

Static Update Controller (SUC)

The PC controller may optionally support the Static Update Controller functionality. The controller must be enabled to accept an assignment of the SUC role in the Z-Wave network. Finally If the SUC functionality is supported the controller must accept being assigned as the SUC of a Z-Wave network.

Controller replication of the application part

The PC Controller must respond with ZW_ReplicationReceiveComplete in case it receives the replication command class otherwise will the replication of protocol data fail. Optionally can the PC Controller interpret the data transferred by the replication command class or simply skip the data.

Clock

The PC Controller may optionally support the clock functionality.

5.4.3.1.1 Mandatory Command Classes to Support

No mandatory command classes to support are defined on the specific device class level.

5.4.3.1.2 Recommended Command Classes to Support

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities.

5.4.3.1.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.4.3.2 Scene Controller Specific Device Class

The Scene Controller Specific Device Class can be used to configure and control scenes in other nodes in a Z-Wave network.

The Scene controller supports the following functionality

- Association (mandatory - support)
- Scene Activation (mandatory - control)
- Scene Controller Configuration (mandatory - support)
- Manufacturer Specific (mandatory - support)
- Scene Actuator Configuration (optional - control)
- Static update controller (recommended)
- Controller replication of the application part (recommended)

Some of the supported functionalities are described in the following:

Associations

Is used to organize nodes in different groups allowing the device to identify the nodes by a group identifier. The groups can also be copied to other devices.

Scene Activation

Is used to activate scenes in other devices and indicate the active scene on the scene-controlling device. The Basic command class to ensure backwards compatibility must control nodes in a group not supporting scene activation.

Scene Controller Configuration

Is used to bind a physical identifier e.g. a push button to a group of associations and the wanted scene settings.

Static Update Controller (SUC) or SUC ID Server (SIS)

The Scene controller may optionally support the SUC/SIS functionality. In case this functionality is supported the controller must accept being assigned the role as SUC/SIS in the Z-Wave network.

Controller replication of the application part

The Scene Controller must respond with `ZW_ReplicationReceiveComplete` in case it receives the replication command class otherwise will the replication of protocol data fail. Optionally can the Scene Controller interpret the data transferred by the replication command class or simply skip the data.

Combo devices

A Scene Controller could optionally be extended with dimmer functionality. The combo device must then be able to differ between which functionalities (load or LED indicator) it should react on when receiving a Scene Activation Set command.

The combo device does this by checking its own configuration to interpret how the received command should be carried out. When a combo device receives Scene Activation Set command then it should check if any of the controller's buttons has the received scene ID associated with it. If NO, the command is clearly send for the dimmer/switch and should be implemented. If YES, the device should check the association list for the controller's button bind to the received scene. If the list has device's own ID in it then the scene command send for both identities of the device: dimmer and controller and so should be implemented. If the own ID is not in the association list for button then this scene ID was send just for display and dimmer/switch shouldn't respond to it.

5.4.3.2.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Association Command Class
- Scene Controller Configuration Command Class
- Basic Command Class
- Manufacturer Specific Command Class

5.4.3.2.2 Recommended Command Classes to Support

The version command class can be used to retrieve the command class revisions etc. supported by the device.

The scene activation command class can be used in case the scene controller which in addition to being scene controller also control a load locally.

5.4.3.2.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Scene Activation Command Class

5.4.3.2.4 Recommended Command Classes to Control

The following command classes can optionally be supported to allow scene controllers exchange configurations:

- Association Command Class
- Scene Controller Configuration Command Class
- Scene Actuator Configuration Command Class
- All Switch Command Class

5.4.3.3 Static Installer Tool Specific Device Class

The Static Installer Tool Specific Device Class is envisioned to be used to set-up and configure a Z-Wave network and for configuration of the various nodes in a network. The controller is a static controller but is not envisioned to be a part of the network after the initial installation and configuration is performed. For this reason it is not recommended to enable SUC /SIS functionality in the controller, but the controller must be able to appoint the SUC/SIS role to another node in the network.

5.4.3.3.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Controller Replication Command Class
- Multi Command Command Class
- Version Command Class
- Manufacturer Specific Command Class

5.4.3.3.2 Recommended Command Classes to Support

The following command classes are recommended to be supported:

- Clock Command Class

5.4.3.3.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Multi Channel Command Class - Version 2
- Controller Replication Command Class
- Association Command Class
- Multi Channel Association Command Class - Version 2
- Wake Up Command Class
- Configuration Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.4.3.3.4 Recommended Command Classes to Control

A number of command classes are recommended in order to retrieve information and configure more advanced devices:

- Binary Switch Command Class
- Multilevel Switch Command Class
- Battery Command Class
- Scene Controller Configuration Command Class
- Scene Actuator Configuration Command Class
- All Switch Command Class
- Node Naming and Location Command Class
- Grouping Name Command Class
- Protection Command Class
- Powerlevel Command Class

5.5 Repeater Slave Generic Device Class

A Repeater Slave device class is a node that doesn't have any application functionality that can be controlled, but only exist in the network to assist other nodes to reach each other.

The Repeater Slave Device Class identifier is equal to `GENERIC_TYPE_REPEATER_SLAVE` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_SLAVE` or `BASIC_TYPE_ROUTING_SLAVE`.

5.5.1 Mandatory Command Classes to Support

The following command classes to support are defined on the generic device class level:

- Basic Command Class

5.5.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set and Basic Get = The Basic Set and basic Get can freely be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Set and Get must be mapped within the same command class. In case the manufacture deems that no relevant commands are available for mapping the command a device receiving these commands must ignore them.

Basic Report = The Basic Report can be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Report must be mapped within the same command class as the Basic Get and must be in accordance with the mapped Get. In case the manufacture deems that no relevant commands are available for mapping the Basic Get command a device is not allowed to respond with a Basic Report under any circumstances.

5.5.2 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the generic device class level.

5.5.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Repeater Slave Generic Device Class:

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Basic Repeater Slave	SPECIFIC_TYPE_REPEATER_SLAVE

Table 7. Specific Device Class identifiers for the Repeater Slave Generic Device Class

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.5.3.1 Basic Repeater Slave Specific Device Class

The Basic Repeater Slave device can be used to repeat frames using several hops to extend range.

The Basic Repeater Slave supports the following functionality

- Powerlevel test (recommended)
- Association (recommended)

The supported functionalities are described in the following:

Powerlevel test

Used to perform a RF link test during installation.

Association

Can be used to configure a RF link test during installation.

5.5.3.1.1 Mandatory Command Classes to Support

No mandatory command classes to support are defined on the specific device class level.

5.5.3.1.2 Recommended Command Classes to Support

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities.

5.5.3.1.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.6 Multilevel Switch Generic Device Class

The Multilevel Switch Generic device class supports all devices that need to switch between several (more than two) states. This Device Class supports for example light dimmers.

The Multilevel Switch Device Class identifier is equal to `GENERIC_TYPE_SWITCH_MULTILEVEL` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_SLAVE` or `BASIC_TYPE_ROUTING_SLAVE`.

The device supports the following functionalities:

- Multilevel Switch (mandatory)

The supported functionalities are described in the following:

Multilevel Switch

The Multilevel Switch shall be able to receive commands instructing the device to change state between various levels, such as setting light dim levels or fan speeds.

5.6.1 Mandatory Command Classes to Support

The following command classes to support are defined on the generic device class level:

- Basic Command Class
- Multilevel Switch Command Class

5.6.2 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the generic device class level.

5.6.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Multilevel Switch Generic Device Class:

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Multilevel Power Switch	SPECIFIC_TYPE_POWER_SWITCH_MULTILEVEL
Multilevel Scene Switch	SPECIFIC_TYPE_SCENE_SWITCH_MULTILEVEL
Multiposition Motor	SPECIFIC_TYPE_MOTOR_MULTIPosition
Motor Control Class A	SPECIFIC_TYPE_CLASS_A_MOTOR_CONTROL
Motor Control Class B	SPECIFIC_TYPE_CLASS_B_MOTOR_CONTROL
Motor Control Class C	SPECIFIC_TYPE_CLASS_C_MOTOR_CONTROL

Table 8. Specific Device Class identifiers for Multilevel Switch Generic Device Class

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.6.3.1 No Specific Device Class defined

This option is selected in case the generic level is appropriate for defining the device.

5.6.3.1.1 Mandatory Command Classes to Support

No mandatory command classes to support are defined on the specific device class level.

5.6.3.1.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set	=	Multilevel Switch Set
Basic Get	=	Multilevel Switch Get
Basic Report	=	Multilevel Switch Report

5.6.3.2 Multilevel Power Switch Specific Device Class

A Multilevel Power Switch device inherits all the commands from the Multilevel Switch Device Class. Apart from these Command Classes it also supports the All Switch Command Class.

The Multilevel Power Switch Device Class supports devices that need to turn power on/off and also be able to dim. The device supports the following functionalities:

- All switch (mandatory)
- Protection (recommended)
- Clock (recommended)

The supported functionalities are described in the following:

All Switch

The All Switch functionality is mandatory. This functionality allows the switch to take part in all on/all off operations.

Protection

The Multilevel Power Switch may optionally support the Protection functionality. If the protection functionality is supported, then the device shall be able to handle protection related commands.

Clock

The Multilevel Power Switch may optionally support the clock functionality. If the clock functionality is supported, then the Multilevel Power Switch shall be able to handle clock related commands.

5.6.3.2.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- All Switch Command Class

5.6.3.2.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

- Basic Set = Multilevel Switch Set
- Basic Get = Multilevel Switch Get
- Basic Report = Multilevel Switch Report

5.6.3.2.2 Recommended Command Classes to Support

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities.

The following Command Classes can optionally be supported:

- Clock Command Class
- Protection command class

5.6.3.2.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.6.3.3 Multilevel Scene Switch Specific Device Class

A Multilevel Scene Switch device inherits all the mandatory command classes from the Multilevel Switch Generic Device Class.

The Multilevel Scene Switch Device Class supports devices that need to turn power on/off and also are able to dim. Furthermore the device can take part in scene operations. The device supports the following functionalities:

- Scene Activation (mandatory)
- Scene Actuator Configuration (mandatory)
- All Switch (mandatory)
- Manufacturer Specific (mandatory)

Some of the supported functionalities are described in the following:

Scene Activation

Is used to control scenes in the multilevel switch.

Scene Actuator Configuration

Is used to configure scenes in the multilevel switch.

All Switch

This functionality allows the switch to take part in all on/all off operations.

5.6.3.3.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Scene Activation Command Class
- Scene Actuator Configuration Command Class
- All Switch Command Class
- Manufacturer Specific Command Class

5.6.3.3.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set	=	Multilevel Switch Set
Basic Get	=	Multilevel Switch Get
Basic Report	=	Multilevel Switch Report

5.6.3.3.2 Recommended Command Classes to Support

The version command class can be used to retrieve the command class revisions etc. supported by the device.

5.6.3.3.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.6.3.3.4 Recommended Command Classes to Control

No recommended command classes to control with are defined on the specific device class level.

5.6.3.4 Multiposition Motor Specific Device Class (Not recommended)

It is not recommended to use the Multiposition Motor Specific Device Class for new products.

Instead it is recommended to base product development of this category on the Motor Control Class A/B/C Specific Device Class.

A Multiposition Motor device inherits all the mandatory command classes from the Multilevel Switch Generic Device Class.

The Multiposition Motor Device Class supports simple motorized devices such as blinds and shades and other multiposition/bidirectional motors. Enable the movement to absolute positions and to start / stop relative movements using the same controller keys as for dimmers. Expand the view from window coverings to other types of motorized devices that share the same requirements and capabilities. The device supports the following functionalities:

- Manufacturer Specific (mandatory)
- Version (mandatory)

5.6.3.4.1 Mandatory Command Classes to Support

In addition to the inherited command classes from the generic device the following must also be supported:

- Manufacturer Specific Command Class
- Version Command Class

No mandatory command classes to support are defined on the specific device class level.

5.6.3.4.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set = Multilevel Switch Set.

Basic Get = Multilevel Switch Get

Basic Report = Multilevel Switch Report

5.6.3.4.2 Recommended Command Classes to Support

No recommended command classes to support are defined on the specific device class level.

5.6.3.4.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.6.3.4.4 Recommended Command Classes to Control

No recommended command classes to control with are defined on the specific device class level.

5.6.3.5 Motor Control Class A Specific Device Class

A Motor Control Class A Specific Device Class inherits all the commands from the Multilevel Switch Generic Device Class.

The Motor Control Class A is used for applications with no indication of position or endpoints hence it is not possible for this type of device to report its position at endpoints or position between endpoints. It can be ordered to move in directions and to stop the movement. Typical application implementing Motor Control Class A is e.g. a ceiling fan.

5.6.3.5.1 Mandatory Command Classes to Support

The Motor Control Class A Specific Device Class must support and implement the below command classes accordingly.

- **Binary Switch Command Class**

The Binary Switch Command Class support simple controller devices requiring “stop-on-reverse” functionality.

Note! It is allowed for the Motor Control Class A Specific Device Class to report 0xFE as an unknown state/position.

Binary Switch Set (0x00) = If motor is stopped then start motion in direction A.
If motor is already running in direction A, then ignore.
If motor is running in direction B then stop motion.

Binary Switch Set (0x01 – 0xFE) = Must be ignored.

Binary Switch Set (0xFF) = If motor is stopped then start motion in direction B.
If motor is already running in direction B, then ignore.
If motor is running in direction A then stop motion.

Binary Switch Get = Respond with Binary Switch Report. Valid values:
0x00 = last moving direction A
0xFE = unknown
0xFF = last moving direction B

- **Multilevel Switch Command Class (Version 3)**

The Multilevel Switch Command Class support controller devices requiring absolute deterministic control of the Motor Control device.

Note! “dimming duration”, “start level” and “ignore start level” fields from the Multilevel Switch Set and Multilevel Switch Start Level Change commands must be ignored and in addition it is allowed for the Motor Control Class A Specific Device Class to report 0xFE as an unknown state/position.

Multilevel Switch Set (0x00) = Start motion in direction A.

Multilevel Switch Set (0x01-0x62 – 0x64-0xFE) = Must be ignored.

Multilevel Switch Set (0x63 or 0xFF) = Start motion in direction B.

Multilevel Switch Get	=	Respond with Multilevel Switch Report. Valid values: 0x00 = last moving direction A 0xFE = unknown 0x63/0xFF = last moving direction B
Multilevel Switch Start Level Change	=	Command parameter Up/Down controls motion in direction A/B respectively and parameters Increment/Decrement and Step Size controls the general increment and decrement functions. All other command parameters must be ignored.
Multilevel Switch Stop Level Change	=	Stop motion.

- **Manufacture Specific Command Class**

The Manufacture Specific Command Class will advertise manufacture specific information e.g. manufacture id, product id etc.

- **Version Command Class**

The Version Command Class is used to report the library type, protocol version and application version of the device.

5.6.3.5.2 Basic Command Class Implementation

The Basic Command Class must be implemented in the following way:

Basic Set	=	Multilevel Switch Set (see Mandatory Command Class to support)
Basic Get	=	Multilevel Switch Get (see Mandatory Command Class to support)
Basic Report	=	Multilevel Switch Report (see Mandatory Command Class to support)

5.6.3.5.3 Recommended Command Classes to Support

No recommended command classes to support.

5.6.3.5.4 Mandatory Command Classes to Control

No mandatory command classes to control.

5.6.3.6 Motor Control Class B Specific Device Class

A Motor Control Class B Specific Device Class inherits all the commands from the Multilevel Switch Generic Device Class.

The Motor Control Class B is used for applications with knowledge of endpoints but imprecise position. It can be instructed to go to endpoints and to stop motion. This motor class can report its position at endpoints and also an imprecise position between endpoints. Typical application implementing Motor Control Class B is e.g. up/down motion of window shades where the position is measured over time.

5.6.3.6.1 Mandatory Command Classes to Support

The Motor Control Class B Specific Device Class must support and implement the below command classes accordingly.

- **Binary Switch Command Class**

The Binary Switch Command Class support simple controller devices requiring “stop-on-reverse” functionality.

Note! It is allowed for the Motor Control Class B Specific Device Class to report 0xFE as an unknown state/position.

Binary Switch Set (0x00) = If motor is stopped then go to endpoint A.
If motor is already running towards endpoint A then ignore.
If motor is running towards endpoint B then stop motion.

Binary Switch Set (0x01 – 0xFE) = Must be ignored.

Binary Switch Set (0xFF) = If motor is stopped then go to endpoint B.
If motor is already running towards endpoint B then ignore.
If motor is running towards endpoint A then stop motion.

Binary Switch Get = Respond with Binary Switch Report. Valid values:
0x00 = at endpoint A
0xFE = unknown
0xFF = at endpoint B

- **Multilevel Switch Command Class (Version 3)**

The Multilevel Switch Command Class support controller devices requiring absolute deterministic control of the Motor Control device.

Note! “dimming duration”, “start level” and “ignore start level” fields from the Multilevel Switch Set and Multilevel Switch Start Level Change commands must be ignored and in addition it is allowed for the Motor Control Class B Specific Device Class to report 0xFE as an unknown state/position.

Multilevel Switch Set (0x00) = Go to endpoint A.

Multilevel Switch Set (0x01-0x62 – 0x64-0xFE) = Must be ignored.

Multilevel Switch Set (0x63 or 0xFF) = Go to endpoint B.

Multilevel Switch Get	=	Respond with Multilevel Switch Report. Valid values: 0x00 = at endpoint A 0x01-0x62 = approximate position between endpoints 0xFE = unknown 0x63/0xFF = at endpoint B
Multilevel Switch Start Level Change	=	Command parameter Up/Down controls motion in direction A/B respectively and parameters Increment/Decrement and Step Size controls the general increment and decrement functions. All other command parameters must be ignored.
Multilevel Switch Stop Level Change	=	Stop motion.

- **Manufacture Specific Command Class**

The Manufacture Specific Command Class will advertise manufacture specific information e.g. manufacture id, product id etc.

- **Version Command Class**

The Version Command Class is used to report the library type, protocol version and application version of the device.

5.6.3.6.2 Basic Command Class Implementation

The Basic Command Class must be implemented in the following way:

Basic Set	=	Multilevel Switch Set (see Mandatory Command Class to support)
Basic Get	=	Multilevel Switch Get (see Mandatory Command Class to support)
Basic Report	=	Multilevel Switch Report (see Mandatory Command Class to support)

5.6.3.6.3 Recommended Command Classes to Support

No recommended command class to support.

5.6.3.6.4 Mandatory Command Classes to Control

No mandatory command classes to control.

5.6.3.7 Motor Control Class C Specific Device Class

A Motor Control Class C Specific Device Class inherits all the commands from the Multilevel Switch Generic Device Class.

The Motor Control Class C is used for application with knowledge of precise position i.e. the motor component provides feedback. This class can be instructed to go to as well as report back its exact position and endpoints. Typical application implementing Motor Control Class C is motor controllers having feedback mechanism.

5.6.3.7.1 Mandatory Command Classes to Support

The Motor Control Class C Specific Device Class must support the below command classes.

- **Binary Switch Command Class**

The Binary Switch Command Class support simple controller devices requiring “stop-on-reverse” functionality.

Note! It is allowed for the Motor Control Class C Specific Device Class to report 0xFE as an unknown state/position.

Binary Switch Set (0x00) = If motor is stopped then go to endpoint A.
If motor is already running towards endpoint A, then ignore.
If motor is running towards endpoint B then stop motion.

Binary Switch Set (0x01 – 0xFE) = Must be ignored.

Binary Switch Set (0xFF) = If motor is stopped then go to endpoint B.
If motor is already running towards endpoint B, then ignore.
If motor is running towards endpoint A then stop motion.

Binary Switch Get = Respond with Binary Switch Report. Valid values:
0x00 = at endpoint A
0xFE = unknown
0xFF = at endpoint B

- **Multilevel Switch Command Class (Version 3)**

The Multilevel Switch Command Class support controller devices requiring absolute deterministic control of the Motor Control device.

Note! “dimming duration”, “start level” and “ignore start level” fields from the Multilevel Switch Set and Multilevel Switch Start Level Change commands must be ignored and in addition it is allowed for the Motor Control Class C Specific Device Class to report 0xFE as an unknown state/position.

Multilevel Switch Set (0x00) = Go to endpoint A.

Multilevel Switch Set (0x01-0x62) = Go to the exact instructed position.

Multilevel Switch Set (0x64-0xFE) = Must be ignored.

Multilevel Switch Set (0x63 or 0xFF)	=	Go to endpoint B.
Multilevel Switch Get	=	Respond with Multilevel Switch Report. Valid values: 0x00 = at endpoint A 0x01-0x62 = <u>exact</u> position between endpoints 0x63/0xFF = at endpoint B
Multilevel Switch Start Level Change	=	Command parameter Up/Down controls motion in direction A/B respectively and parameters Increment/Decrement and Step Size controls the general increment and decrement functions. All other command parameters must be ignored.
Multilevel Switch Stop Level Change	=	Stop motion.

- **Manufacture Specific Command Class**

The Manufacture Specific Command Class will advertise manufacture specific information e.g. manufacture id, product id etc.

- **Version Command Class**

The Version Command Class is used to report the library type, protocol version and application version of the device.

5.6.3.7.2 Basic Command Class Implementation

The Basic Command Class must be implemented in the following way:

Basic Set	=	Multilevel Switch Set (see Mandatory Command Class to support)
Basic Get	=	Multilevel Switch Get (see Mandatory Command Class to support)
Basic Report	=	Multilevel Switch Report (see Mandatory Command Class to support)

5.6.3.7.3 Recommended Command Classes to Support

No recommended command classes to support.

5.6.3.7.4 Mandatory Command Classes to Control

No mandatory command classes to control.

5.7 Toggle Switch Generic Device Class (Not recommended)

Not recommended to use this generic device class in new devices.

Instead, it is recommended to base devices of this category on the Binary / Multilevel Switch generic device class.

A Toggle Switch device class supports all one-button devices that need to toggle between two states e.g. turn power on/off. Further the device can be extended with dimming capabilities.

The Toggle Switch Device Class identifier is equal to `GENERIC_TYPE_SWITCH_TOGGLE` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_STATIC_CONTROLLER`, `BASIC_TYPE_SLAVE` or `BASIC_TYPE_ROUTING_SLAVE`.

5.7.1 Mandatory Command Classes to Support

The following command classes to support are defined on the generic device class level:

- Basic Command Class

5.7.1.1 Basic Command Class Implementation

Refer to the specific device classes.

5.7.2 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the generic device class level.

5.7.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Toggle Switch Generic Device Class:

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Binary Toggle Switch	SPECIFIC_TYPE_SWITCH_TOGGLE_BINARY
Multilevel Toggle Switch	SPECIFIC_TYPE_SWITCH_TOGGLE_MULTILEVEL

Table 9. Specific Device Class identifiers for the Toggle Switch Generic Device Class

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.7.3.1 Binary Toggle Switch Specific Device Class (Not recommended)

A Binary Toggle Switch device supports all one-button devices that need to toggle between two states e.g. turn power on/off.

5.7.3.1.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Binary Switch Command Class
- Binary Toggle Switch Command Class

5.7.3.1.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set = Binary Toggle Switch Set

Basic Get = Binary Toggle Switch Get

Basic Report = Binary Toggle Switch Report

5.7.3.1.2 Recommended Command Classes to Support

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities.

No recommended command classes to support are defined on the specific device class level.

5.7.3.1.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.7.3.2 Multilevel Toggle Switch Specific Device Class (Not recommended)

A Multilevel Toggle Switch device supports all one-button devices that both need to toggle between two states and have dimmable capabilities.

5.7.3.2.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Multilevel Switch Command Class
- Multilevel Toggle Switch Command Class

5.7.3.2.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set = Multilevel Toggle Switch Set

Basic Get = Multilevel Toggle Switch Get

Basic Report = Multilevel Toggle Switch Report

5.7.3.2.2 Recommended Command Classes to Support

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities.

5.7.3.2.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.8 Remote Switch Generic Device Class

A Remote Switch device class provides a user interface to control the operation of one or more devices via the Z-Wave network.

The Remote Switch Device Class identifier is equal to `GENERIC_TYPE_SWITCH_REMOTE` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_STATIC_CONTROLLER` or `BASIC_TYPE_ROUTING_SLAVE`.

5.8.1 Mandatory Command Classes to Support

No mandatory command classes to support are defined on the generic device class level.

5.8.2 Mandatory Command Classes to Control

The following command classes to control with are defined on the generic device class level:

- Basic Command Class

In case a remote switch device is configured to control an unknown device it shall use basic command to control this device.

5.8.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Remote Switch Generic Device Class:

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Binary Remote Switch	SPECIFIC_TYPE_SWITCH_REMOTE_BINARY
Multilevel Remote Switch	SPECIFIC_TYPE_SWITCH_REMOTE_MULTILEVEL
Binary Toggle Remote Switch	SPECIFIC_TYPE_SWITCH_REMOTE_TOGGLE_BINARY
Multilevel Toggle Remote Switch	SPECIFIC_TYPE_SWITCH_REMOTE_TOGGLE_MULTILEVEL

Table 10. Specific Device Class identifiers for the Remote Switch Generic Device Class

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.8.3.1 Binary Remote Switch Specific Device Class

A Binary Remote Switch device supports for example a switch there can control a Binary Switch Generic Device Class via the Z-Wave network.

5.8.3.1.1 Mandatory Command Classes to Support

No mandatory command classes to support are defined on the specific device class level.

5.8.3.1.2 Recommended Command Classes to Support

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities..

5.8.3.1.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Binary Switch Command Class

5.8.3.2 Multilevel Remote Switch Specific Device Class

A Multilevel Remote Switch device supports for example a switch there can control a Multilevel Switch Generic Device Class via the Z-Wave network.

5.8.3.2.1 Mandatory Command Classes to Support

No mandatory command classes to support are defined on the specific device class level.

5.8.3.2.2 Recommended Command Classes to Support

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities.

5.8.3.2.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Multilevel Switch Command Class

5.8.3.3 Binary Toggle Remote Switch Specific Device Class (Not recommended)

It is not recommended to use this specific device class for new devices.

Instead it is recommended to base devices of this category on the Binary Remote Switch specific device class.

A Binary Toggle Remote Switch device supports for example a one-button switch there can control a Binary Toggle Switch Generic Device Class via the Z-Wave network.

5.8.3.3.1 Mandatory Command Classes to Support

No mandatory command classes to support are defined on the specific device class level.

5.8.3.3.2 Recommended Command Classes to Support

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities.

5.8.3.3.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Binary Toggle Switch Command Class

5.8.3.4 Multilevel Toggle Remote Switch Specific Device Class (Not recommended)

It is not recommended to use this specific device class for new devices.

Instead it is recommended to base devices of this category on the Multilevel Remote Switch specific device class.

A Multilevel Toggle Remote Switch device supports for example a one-button switch there can control a Multilevel Toggle Switch Generic Device Class via the Z-Wave network.

5.8.3.4.1 Mandatory Command Classes to Support

No mandatory command classes to support are defined on the specific device class level.

5.8.3.4.2 Recommended Command Classes to Support

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities.

5.8.3.4.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Multilevel Toggle Switch Command Class

5.9 Binary Sensor Generic Device Class

The Binary Sensor device class is used to realize various sensors.

The Binary Sensor Device Class identifier is equal to `GENERIC_TYPE_SENSOR_BINARY` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_SLAVE` or `BASIC_TYPE_ROUTING_SLAVE`.

The device supports the following functionalities:

- Binary Sensor (mandatory)

The supported functionalities are described in the following:

Binary Sensor

The Binary Sensor must support the Binary Sensor functionality. The Binary Sensor functionality can be used by event driven sensors, i.e. sensors that only send reports when a given action occurs. This is e.g. a movement sensor.

5.9.1 Mandatory Command Classes to Support

The following command classes to support are defined on the generic device class level:

- Basic Command Class
- Binary Sensor Command Class

5.9.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set = Cannot be mapped because the mapped command class doesn't have a Set command. A device receiving this command must simply ignore it.

Basic Get = Binary Sensor Get

Basic Report = Binary Sensor Report

5.9.2 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the generic device class level.

5.9.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Binary Sensor Generic Device Class:

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Routing Binary Sensor	SPECIFIC_TYPE_ROUTING_SENSOR_BINARY

Table 11. Specific Device Class identifiers for the Binary Sensor Generic Device Class

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.9.3.1 Routing Binary Sensor Specific Device Class

The Routing Binary Sensor device is used to realize various sensors. The device supports the following functionalities:

- Wakeup (recommended)
- Association (recommended)
- Clock (recommended)
- Battery (recommended)

The supported functionalities are described in the following:

Wakeup

The Routing Binary Sensor may optionally support the wakeup functionality. If the wakeup functionality is supported, then the Routing Binary Sensor shall be able to handle wakeup related commands.

Association

The Routing Binary Sensor may optionally support the association functionality. If the association functionality is supported, then the Routing Binary Sensor shall be able to handle association related commands.

Clock

The Routing Binary Sensor may optionally support the clock functionality. If the clock functionality is supported, then the Routing Binary Sensor shall be able to handle clock related commands.

Battery

The Routing Binary Sensor may optionally support the battery functionality. If the battery functionality is supported, then the Routing Binary Sensor shall be able to handle battery related commands.

5.9.3.1.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Binary Sensor Command Class

5.9.3.1.2 Recommended Command Classes to Support

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities.

The following Command Classes can optionally be supported:

- Wakeup Command Class
- Association Command Class
- Clock Command Class
- Battery Command Class.

5.9.3.1.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.10 Multilevel Sensor Generic Device Class

The Multilevel Sensor device class is used to realize various sensors.

The Multilevel Sensor Device Class identifier is equal to `GENERIC_TYPE_SENSOR_MULTILEVEL` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_SLAVE` or `BASIC_TYPE_ROUTING_SLAVE`.

The device supports the following functionalities:

- Multilevel Sensor (mandatory)

The supported functionalities are described in the following:

Multilevel Sensor

The device must support the Multilevel Sensor functionality. This functionality allows the Multilevel Sensor to report various values back to the controller.

5.10.1 Mandatory Command Classes to Support

The following command classes to support are defined on the generic device class level:

- Basic Command Class
- Multilevel Sensor Command Class

5.10.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set = Cannot be mapped because the mapped command class doesn't have a Set command. A device receiving this command must simply ignore it.

Basic Get = Multilevel Sensor Get

Basic Report = Multilevel Sensor Report

5.10.2 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the generic device class level.

5.10.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Multilevel Sensor Generic Device Class:

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Routing Multilevel Sensor	SPECIFIC_TYPE_ROUTING_SENSOR_MULTILEVEL

Table 12. Specific Device Class identifiers for the Multilevel Sensor Generic Device Class

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.10.3.1 Routing Multilevel Sensor Specific Device Class

The Routing Multilevel Sensor device is used to realize various sensors. The device supports the following functionalities:

- Wakeup (recommended)
- Association (recommended)
- Clock (recommended)
- Battery (recommended)

The supported functionalities are described in the following:

Wakeup

The Routing Multilevel Sensor may optionally support the wakeup functionality. If the wakeup functionality is supported, then the Routing Binary Sensor shall be able to handle wakeup related commands.

Association

The Routing Multilevel Sensor may optionally support the association functionality. If the association functionality is supported, then the Routing Binary Sensor shall be able to handle association related commands.

Clock

The sensor may optionally support the clock functionality. If the clock functionality is supported, then the sensor shall be able to handle clock related commands.

Battery

The sensor may optionally support the battery functionality. If the battery functionality is supported, then the sensor shall be able to handle battery related commands.

5.10.3.1.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Multilevel Sensor Command Class

5.10.3.1.2 Recommended Command Classes to Support

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities.

The following Command Classes can optionally be supported:

- Wakeup Command Class
- Association Command Class
- Clock Command Class
- Battery Command Class

5.10.3.1.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.11 Pulse Meter Generic Device Class

The Pulse Meter device class is used to realize various meters, such as gas, water and electricity meters. A meter typically reports a number of pulses, which have a specific meaning for a given meter.

The Pulse Meter Device Class identifier is equal to `GENERIC_TYPE_METER_PULSE` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_SLAVE` or `BASIC_TYPE_ROUTING_SLAVE`.

The device supports the following functionalities:

- Pulse Meter (mandatory)

The supported functionalities are described in the following:

Pulse Meter

The device must support the Pulse Meter functionality.

5.11.1 Mandatory Command Classes to Support

The following command classes to support are defined on the generic device class level:

- Basic Command Class
- Pulse Meter Command Class

5.11.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set = Cannot be mapped because the mapped command class doesn't have a Set command. A device receiving this command must simply ignore it.

Basic Get = Pulse Meter Get

Basic Report = Pulse Meter Report

5.11.2 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the generic device class level.

5.11.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Pulse Meter Generic Device Class:

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED

Table 13. Specific Device Class identifiers for the Pulse Meter Generic Device Class

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.12 Display Generic Device Class

The Display Generic Device Class supports a range of displays showing various data such as the current power consumption (electricity), or the number of emails in your in-box, or the outside temperature, or the weather forecast etc.

The Display Generic Device Class identifier is equal to `GENERIC_TYPE_DISPLAY` and the Basic Device Class identifier is typically `BASIC_TYPE_SLAVE`.

5.12.1 Mandatory Command Classes to Support

The generic device must support the following command classes:

- Basic Command Class

5.12.1.1 Basic Command Class Implementation

Refer to the specific device classes.

5.12.2 Mandatory Command Classes to Control

No defined mandatory command classes to control on the generic device class level.

5.12.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Display Generic Device Class:

Table 2, Specific Device Class identifiers for the Display Generic Device Class

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Simple Display	SPECIFIC_TYPE_SIMPLE_DISPLAY

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.12.3.1 No Specific Device Class defined

Select this option in case the generic level is appropriate for defining the device.

5.12.3.1.1 Mandatory Command Classes to Support

No mandatory command classes specified to access functionality in the specific device itself.

5.12.3.1.1.1 *Basic Command Class Implementation*

The basic command class must be implemented in the following way:

Basic Set and Basic Get = The Basic Set and basic Get can freely be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Set and Get must be mapped within the same command class. In case the manufacture deems that no relevant commands are available for mapping the command a device receiving these commands must ignore them.

Basic Report = The Basic Report can be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Report must be mapped within the same command class as the Basic Get and must be in accordance with the mapped Get. In case the manufacture deems that no relevant commands are available for mapping the Basic Get command a device is not allowed to respond with a Basic Report under any circumstances.

5.12.3.2 Simple Display Specific Device Class

The Simple Display Specific Device Class supports a range of general-purpose displays showing text based information of various kinds. Typically, a Simple Display Specific Device Class devices use a slave. The device supports the following functionalities:

- Screen Attributes (mandatory)
- Screen Meta Data (mandatory)
- Manufacturer Specific (mandatory)
- Version (mandatory)

A Simple Display Specific Device Class inherits all the commands from the Display Generic Device Class.

Screen Attributes

Make it possible to obtain display parameters such as number of display lines, number of characters per line etc.

Screen Meta Data

Enables streaming of text based information to the device hosting the display.

5.12.3.2.1 Mandatory Command Classes to Support

In addition to the inherited command classes from the generic device the device itself must also support the following:

- Screen Attributes Command Class
- Screen Meta Data Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.12.3.2.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set and Basic Get = The Basic Set and basic Get can freely be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Set and Get must be mapped within the same command class. In case the manufacture deems that no relevant commands are available for mapping the command a device receiving these commands must ignore them.

Basic Report = The Basic Report can be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Report must be mapped within the same command class as the Basic Get and must be in accordance with the mapped Get. In case the manufacture deems that no relevant commands are available for mapping the Basic Get command a device is not allowed to respond with a Basic Report under any circumstances.

5.12.3.2.2 Recommended Command Classes to Support

Nothing recommended with respect to command classes supported by the device itself.

5.12.3.2.3 Mandatory Command Classes to Control

Nothing mandatory with respect to command classes the device can control in other devices.

5.12.3.2.4 Recommended Command Classes to Control

Nothing recommended with respect to command classes the device can control in other devices.

5.13 Entry Control Generic Device Class

The Entry Control Generic Device Class is used to implement various forms of entry control devices such as door locks.

The Entry Control Generic Device Class identifier is equal to `GENERIC_TYPE_ENTRY_CONTROL` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_ROUTING_SLAVE`.

It is recommended to secure the communication for this type of device in order to provide message integrity and confidentiality with regards to access control.

5.13.1 Mandatory Command Classes to Support

The generic device must support the following Command Classes:

- Basic Command Class

5.13.1.1 Basic Command Class Implementation

Refer to the specific devices for a description.

5.13.2 Recommended Command Classes to Support

No recommended command classes to support are defined on the generic device class level.

5.13.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the generic device class level.

5.13.4 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Entry Control Generic Device Class:

Specific Device Class	ZW_classcmd.h
Specific Device Class Not Used	SPECIFIC_TYPE_NOT_USED
Door Lock	SPECIFIC_TYPE_DOOR_LOCK
Advanced Door Lock	SPECIFIC_TYPE_ADVANCED_DOOR_LOCK
Secure Keypad Door Lock	SPECIFIC_TYPE_SECURE_KEYPAD_DOOR_LOCK

Table 14. Specific Device Class identifiers for the Entry Control Generic Device Class

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.13.4.1 Specific Device Class Not Used

This option is selected in case the generic level is appropriate for defining the device.

5.13.4.1.1 Mandatory Command Classes to Support

No mandatory command classes to support are defined on the specific device class level..

5.13.4.1.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

- | | |
|---------------------------|---|
| Basic Set and Basic Get = | The Basic Set and basic Get can freely be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Set and Get must be mapped within the same command class. In case the manufacture deems that no relevant commands are available for mapping the command a device receiving these commands must ignore them. |
| Basic Report = | The Basic Report can be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Report must be mapped within the same command class as the Basic Get and must be in accordance with the mapped Get. In case the manufacture deems that no relevant commands are available for mapping the Basic Get command a device is not allowed to respond with a Basic Report under any circumstances. |

5.13.4.2 Door Lock Specific Device Class

An Door Lock Specific Device Class inherits all the commands from the Entry Control Generic Device Class.

The Door Lock Device Class supports devices that can be used to lock or unlock a door. The device supports the following functionalities:

5.13.4.2.1 Mandatory Command Classes to Support

The Door Lock Specific Device must support the following Command Classes:

- Lock Command Class
Simple Command Class to control the Door Lock

5.13.4.2.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set = Must be ignored to avoid unintentional operation. Cannot be mapped to another command class.

Basic Get = Lock Get

Basic Report = Lock Report

5.13.4.2.2 Recommended Command Classes to Support

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities

- Manufacture Specific Command Class
Command Class to advertise manufacture specific information
- Version Command Class
The Version Command Class can be used to report the library type, protocol version and application version from a device

5.13.4.2.3 Mandatory Command Classes to Control

No mandatory command classes to control functionality in other devices from the specific device.

5.13.4.3 Advanced Door Lock Specific Device Class

An Advanced Door Lock Specific Device Class inherits all the commands from the Entry Control Generic Device Class.

The Advanced Door Lock Device Class supports devices that can be used to lock or unlock a door. The device supports the following functionalities:

5.13.4.3.1 Mandatory Command Classes to Support

The Door Lock Command Class is used to control the Advanced Door Lock Device and to monitor the status.

- Door Lock Command Class
Command Class to control the Door Lock
- Manufacture Specific Command Class
Command Class to advertise manufacture specific information
- Version Command Class
The Version Command Class can be used to report the library type, protocol version and application version from a device

5.13.4.3.1.1 Basic Command Class Implementation

The Basic Command Class must be implemented in the following way:

Basic Set	=	Must be ignored to avoid unintentional operation. Cannot be mapped to another command class.
Basic Get	=	Door Lock Operation Get
Basic Report	=	Door Lock Operation Report

5.13.4.3.2 Recommended Command Classes to Support

The following Command Classes can optionally be supported:

- User Code Command Class
The User Code Command Class can be used by the application to configure the user codes that can unlock the door.
- Alarm Command Class
Command Class to announce alarm condition e.g. wrong code entered or door is open but lock status is closed.
- Protection Command Class
The Door Lock may optionally support the Protection functionality. If the protection functionality is supported, the device shall be able to handle protection related commands
- Battery Command Class
The Battery Command Class can report to the controller about the battery status and level.

5.13.4.3.3 Mandatory Command Classes to Control

No mandatory command classes to control functionality in other devices from the specific device.

5.13.4.4 Secure Keypad Door Lock Specific Device Class

The Secure Keypad Door Lock Specific Device Class inherits all the commands from the Entry Control Generic Device Class.

The Secure Keypad Door Lock Specific Device Class supports devices that can be used to lock or unlock a door both remotely and locally using keypad.

Please note that Security Command class is required for these devices, and that is mandatory to support some of the command classes only encapsulated into the Security Command Class.

5.13.4.4.1 Mandatory Command Classes to Support

The Secure Keypad Door Lock Specific Device must support the following Command Classes:

- Security Command Class
Command Class to encrypt messages using AES-128
- Basic Command Class (secured)
The basic command class can be used to obtain the operation mode of the lock, the mapping is given in Section 5.13.4.4.1.1.
- Door Lock Command Class (secured)
Command Class to control the Door Lock
- User Code Command Class (secured)
Command Class to set and retrieve user codes
- Manufacture Specific Command Class (unsecure)
Command Class to advertise manufacture specific information
- Version Command Class (unsecure)
The Version Command Class can be used to report the library type, protocol version and application version from a device

5.13.4.4.1.1 Basic Command Class Implementation

The Basic Command Class must be implemented in the following way:

Basic Set	=	Must be ignored to avoid unintentional operation. Cannot be mapped to another command class.
Basic Get	=	Door Lock Operation Get
Basic Report	=	Door Lock Operation Report

Due to Security concerns non-secure Basic Command Class commands will be ignored by device.

5.13.4.4.2 Recommended Command Classes to Support

The following Command Classes can optionally be supported:

- Alarm Command Class (secure or not secure)
Command Class to announce alarm condition e.g. wrong code entered or door is open but lock status is closed.
- Protection Command Class (secure or not secure)

The Door Lock may optionally support the Protection functionality. If the protection functionality is supported, the device shall be able to handle protection related commands

- Battery Command Class (secure or not secure)
The Battery Command Class can report to the controller about the battery status and level.

5.13.4.4.3 Mandatory Command Classes to Control

No mandatory command classes to control functionality in other devices from the specific device class.

5.14 Semi Interoperable Generic Device Class

The Semi Interoperable Generic Device Class is used to realize devices comprising additional functionality compared to the typically home automation application.

The Semi Interoperable Device Class identifier is equal to `GENERIC_TYPE_SEMI_INTEROPERABLE` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_SLAVE` or `BASIC_TYPE_ROUTING_SLAVE`.

The Semi Interoperable Device supports the following functionalities:

- Device Information (mandatory)
- Proprietary Support (mandatory)

The supported functionalities are described in the following:

Device Information

The device must support various command classes to obtain detailed information regarding manufacturer, versions etc. This information is important to transfer the vendor specific parameters correct.

Proprietary Support

The device must be able to transfer the vendor specific parameters using a command class without pre-defined contents of the data fields.

5.14.1 Mandatory Command Classes to Support

The following command classes to support are defined on the generic device class level:

- Basic Command Class
- Manufacturer Specific Command Class
- Version Command Class
- Proprietary Command Class

The use of proprietary commands is always subject to approval.

Support and/or control of other command classes maybe made a condition when approving the use of proprietary commands.

5.14.1.1 Basic Command Class Implementation

Refer to the specific device classes.

5.14.2 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the generic device class level,

5.14.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Semi Interoperable Generic Device Class:

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Energy Production	SPECIFIC_TYPE_ENERGY_PRODUCTION

Table 15. Specific Device Class identifiers for the Semi Interoperable Generic Device Class

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.14.3.1 Energy Production Specific Device Class

An Energy Production device supports various equipment for energy production. The device supports the following functionalities:

- Energy Production (mandatory)

The supported functionalities are described in the following:

Energy Production

The energy production command class is used to read various production parameters.

5.14.3.1.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Energy Production Command Class

5.14.3.1.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set and Basic Get = The Basic Set and basic Get can freely be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Set and Get must be mapped within the same command class. In case the manufacture deems that no relevant commands are available for mapping the command a device receiving these commands must ignore them.

Basic Report = The Basic Report can be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Report must be mapped within the same command class as the Basic Get and must be in accordance with the mapped Get. In case the manufacture deems that no relevant commands are available for mapping the Basic Get command a device is not allowed to respond with a Basic Report under any circumstances.

5.14.3.1.2 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.15 Window Covering Generic Device Class (Not recommended)

It is not recommended to use this generic device class for new devices.

Instead, it is recommended to base devices of this category on the Multi-level switch generic device class with the Multi-position Motor specific device class.

The Window Covering device is used to control motorized drapes, shades and blinds etc.

Window Covering Device Class identifier is equal to `GENERIC_TYPE_WINDOW_COVERING` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_STATIC_CONTROLLER` or `BASIC_TYPE_ROUTING_SLAVE`.

5.15.1 Mandatory Command Classes to Support

No mandatory command classes to support are defined on the generic device class level.

5.15.1.1 Basic Command Class Implementation

Refer to the specific devices for a description

5.15.2 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the generic device class level.

5.15.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Window Covering Generic Device Class:

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	<code>SPECIFIC_TYPE_NOT_USED</code>
Simple Window Covering Control	<code>SPECIFIC_TYPE_SIMPLE_WINDOW_COVERING</code>

Table 16. Specific Device Class identifiers for the Window Covering Generic Device Class

Refer to `ZW_classcmd.h` source code file for the assigned Specific Device Class identifiers.

5.15.3.1 Simple Window Covering Control Specific Device Class (Not recommended)

The Simple Window Covering Control device can be used to control motorized drapes, shades, blinds etc. The simple control comprises of commands to start open/close and stop window coverings.

5.15.3.1.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Basic Window Covering

5.15.3.1.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set (Value = 0x00) = Basic Window Covering Start Level Change (Open).

Basic Set (Value = 0xFF) = Basic Window Covering Start Level Change (Close).

Basic Get and Report = Cannot be mapped because the mapped command class doesn't have a Get command. A device receiving this command must simply ignore it.

Basic Report = Cannot be mapped because the mapped command class doesn't have a Report command. A device is not allowed to respond with a Basic Report under any circumstances.

If the shade is stopped, a Basic Set command will start motion in the specified direction. If the shade is moving, a Basic Set command in the same direction will be ignored. If the shade is moving, a Basic Set command in the opposite direction will stop motion.

5.15.3.1.2 Recommended Command Classes to Support

The manufacturer specific and version command class is highly recommended because they can be used to identify the device and its capabilities. The multi instance command class can be used for window coverings using two motion controls. To access various settings use the configuration command class.

5.15.3.1.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.16 Thermostat Generic Device Class

A Thermostat device that enables the user to set a comfort setpoint temperature is maintained.

The Thermostat Device Class identifier is equal to GENERIC_TYPE_THERMOSTAT and will typically be based on the Basic Device Class identifier BASIC_TYPE_STATIC_CONTROLLER or BASIC_TYPE_ROUTING_SLAVE.

5.16.1 Mandatory Command Classes to Support

The basic command class must be implemented for all permanently listening devices. Battery-operated devices may implement the basic command class.

5.16.1.1 Basic Command Class Implementation

Refer to the specific devices for a description

5.16.2 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the generic device class level.

5.16.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Thermostat Generic Device Class:

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Thermostat Heating	SPECIFIC_TYPE_THERMOSTAT_HEATING
Thermostat General	SPECIFIC_TYPE_THERMOSTAT_GENERAL
Thermostat General V2	SPECIFIC_TYPE_THERMOSTAT_GENERAL_V2
Setback Schedule Thermostat	SPECIFIC_TYPE_SETBACK_SCHEDULE_THERMOSTAT
Setback Thermostat	SPECIFIC_TYPE_SETBACK_THERMOSTAT
Setpoint Thermostat	SPECIFIC_TYPE_SETPOINT_THERMOSTAT

Table 17. Specific Device Class identifiers for the Thermostat Generic Device Class

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.16.3.1 Thermostat Heating Specific Device Class (Not permitted)

It is not permitted to use this specific device class for new devices.

Instead it is recommended to base devices of this category on the General Thermostat V2 specific device class or other specific devices classes defined.

The Thermostat Heating device is defined in [5].

5.16.3.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

- Basic Set and Basic Get = The Basic Set and basic Get can freely be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Set and Get must be mapped within the same command class. In case the manufacture deems that no relevant commands are available for mapping the command a device receiving these commands must ignore them.
- Basic Report = The Basic Report can be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Report must be mapped within the same command class as the Basic Get and must be in accordance with the mapped Get. In case the manufacture deems that no relevant commands are available for mapping the Basic Get command a device is not allowed to respond with a Basic Report under any circumstances.

5.16.3.2 Thermostat General Specific Device Class (Not permitted)

It is not permitted to use this specific device class for new devices.

Instead it is recommended to base devices of this category on the General Thermostat V2 specific device class or other specific devices classes defined.

5.16.3.2.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Manufacturer Specific Command Class
- Thermostat Mode Command Class
- Thermostat Setpoint Command Class

5.16.3.2.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set = Not supported to avoid potential liability issues where a user inadvertently turned his heating off. A device receiving this command must simply ignore it.

Basic Get = Thermostat Mode Get

Basic Report = Thermostat Mode Report

5.16.3.2.2 Recommended Command Classes to Support

The version command class is highly recommended because they can be used to retrieve versions of the command classes etc. supported by the device.

The multilevel sensor command class can be used to read the current temperature. Remember also to consider the relevance of the thermostat operating state, thermostat fan mode and thermostat fan state command classes when designing a thermostat. Finally the multi instance command class can be used in case a secondary temperature sensor (outside or water temperature) is present.

5.16.3.2.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.16.3.3 Thermostat General V2 Specific Device Class

A Thermostat General device supports a general thermostat.

5.16.3.3.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Basic Command Class
- Manufacturer Specific Command Class
- Version Command Class
- Thermostat Mode Command Class
- Thermostat Setpoint Command Class

5.16.3.3.1.1 Basic Command Class Implementation

The basic command class shall be implemented in the following way:

Basic Set (Value = 0x00) =	Set Energy Saving Mode
Basic Set (Value = 0xFF) =	Set Comfort Mode
Basic Get =	Get report
Basic Report (Value = 0x00) =	Report Energy Saving Mode
Basic Report (Value = 0xFF) =	Report Comfort Mode

Note: The implementation of Energy Saving Mode is manufacturer specific, and must be documented in the User's Manual.

5.16.3.3.2 Recommended Command Classes to Support

The multilevel sensor command class can be used to read the current temperature. Remember also to consider the relevance of the thermostat operating state, thermostat fan mode and thermostat fan state command classes when designing a thermostat. Finally the multi instance command class can be used in case a secondary temperature sensor (outside or water temperature) is present.

5.16.3.3.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.16.3.4 Setback Schedule Thermostat Specific Device Class

Setback Schedule Thermostats control the temperature in one or multiple rooms. The thermostat has a setpoint, which is used for normal operation. But to save additional energy the thermostat has the ability to go into energy saving mode controlled by a schedule.

The schedule is downloaded to the thermostat from the controller once the system is set up and every time the user changes the schedule in the controller. The thermostat has a schedule for one week. After one week the sequence in the schedule is repeated.

The schedule can be overridden if the homeowner returns to his home earlier, is away on vacation for longer time or is taking a day off at home.

Note: A device must both support and control the Multi Command Command Class including relevant encapsulated command classes to be able to operate a Setback Schedule Thermostat Specific Device.

5.16.3.4.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Climate Control Schedule Command Class
- Manufacturer Specific Command Class
- Multi Command Command Class
- Version Command Class

A battery-operated Setback Schedule Thermostat device must additionally implement as supported the following command classes:

- Battery Command Class
- Wake Up Command Class Version 2

The Setback Schedule Thermostat shall understand supported commands that are encapsulated by a Multi Command.

Note: In order to prolong the battery-life of battery-operated devices it is recommended to always send as many commands as possible encapsulated in a Multi Command.

5.16.3.4.1.1 Basic Command Class Implementation

The basic command class shall be implemented in the following way:

Basic Set (Value = 0x00) =	Schedule Override Set (Temporary Override, Energy Saving Mode)
Basic Set (Value = 0xFF) =	Schedule Override Set (Temporary Override, 0 degree setback)
Basic Get =	Schedule Override Get
Basic Report (Value = 0x00) =	Schedule Override Report (Temporary Override, Energy Saving Mode)
Basic Report (Value = 0xFF) =	Schedule Override Report (Temporary Override, 0 degree setback) including the remaining options

Note: The implementation of Energy Saving Mode is manufacturer specific.

5.16.3.4.2 Recommended Command Classes to Support

The battery command class is recommended in case the Setback Schedule Thermostat device is battery-operated.

5.16.3.4.3 Mandatory Command Classes to Control

The following command classes to control with are defined on the specific device class level:

- Climate Control Schedule Command Class
- Multi Command Command Class
- Clock Command Class

Note: In order to prolong the battery-life of battery-operated devices it is recommended to always send as many commands as possible encapsulated in a Multi Command.

5.16.3.4.4 Mandatory Operation

After Inclusion:

A Setback Schedule Thermostat has to know the Node ID of the device which contains the schedules.

A Setback Schedule Thermostat shall perform the following steps after inclusion:

1. Wait for a WAKE_UP_INTERVAL_SET
2. Wait for a WAKE_UP_NO_MORE_INFORMATION

The Setback Schedule Thermostat must be able to accept other supported commands while performing these steps.

The Node which has the schedule ('The Controller') shall perform the following steps after it has been enabled to accept new Setback Schedule Thermostats.

1. Wait for a Node Information Frame
2. Send the relevant WAKE_UP_INTERVAL_SET

The Controller must be able to accept other supported commands while performing these steps.

Battery-operated Setback Schedule Thermostats devices may implement a timeout to allow the device to enter a special mode which preserves battery-life in the case that the wake up functionality is not configured immediately after inclusion. It must require a physical activation of the device to exit the special mode.

Note: The implementation of exiting the special mode is manufacturer specific.

When exiting the special mode the following steps must be performed:

1. Send a Node Information Frame
2. Perform the steps from the previous list for the Setback Schedule Thermostat

If timeout occurs again, the device may re-enter the special mode.

Wake up:

A battery-operated Setback Schedule Thermostat shall perform the following steps on wake up:

1. Send a BATTERY_REPORT
2. Send a SCHEDULE_OVERRIDE_REPORT
3. Send a CLOCK_GET
4. Send a SCHEDULE_CHANGED_GET
5. Send a WAKE_UP_NOTIFICATION

Step 1 - 3 has special considerations:

- Step 1 must be performed on first wake up and may be left out if no new battery measurements were made.
- Step 2 must be performed if the override state of the device has changed and otherwise it may be left out.
- Step 3 must be performed on first wake up and may be left out if an update of time is not needed. It is recommended to synchronize time depending on the needed accuracy.

Note: In order to prolong the battery-life of battery-operated devices it is recommended to always send as many commands as possible encapsulated in a Multi Command. At wake up is it allowed to send commands encapsulated in a Multi Command without checking that the destination supports relevant command classes.

The operation is visualized using Multi Commands in the following sequence diagram:

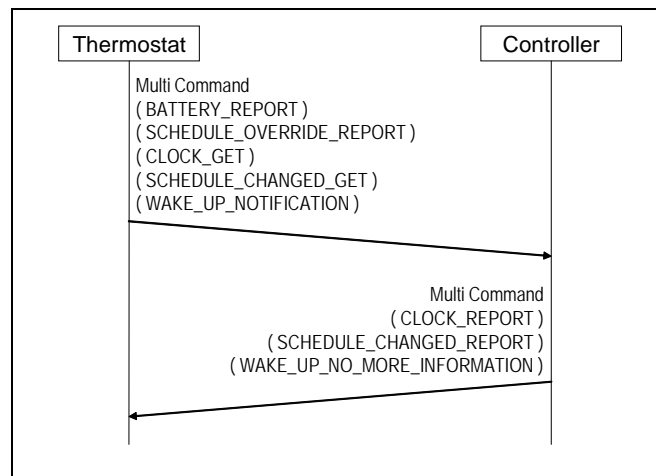


Figure 7. Sequence diagram of operation after wake up

Note: If either device, when using Multi Commands, has the need to send more commands they should try to integrate them into the two Multi Commands exchanged after a wake up as visualized in the above diagram.

Example: A user has pressed a button on the thermostat which makes it wake up. The thermostat then performs the wake up operation but also adds an unsolicited report informing of the user interaction in a Multi Command.

Schedule is not up-to-date:

A battery-operated Setback Schedule Thermostat shall perform the following steps when its schedule is not up-to-date. **These steps should not be performed using Multi Commands as there can only be one schedule in the Multi Command report:**

1. Send a SCHEDULE_GET (Monday)
2. Wait for the SCHEDULE_REPORT (Monday)
3. Send a SCHEDULE_GET (Tuesday)
4. Wait for the SCHEDULE_REPORT (Tuesday)
5. Send a SCHEDULE_GET (Wednesday)
6. Wait for the SCHEDULE_REPORT (Wednesday)
7. Send a SCHEDULE_GET (Thursday)
8. Wait for the SCHEDULE_REPORT (Thursday)
9. Send a SCHEDULE_GET (Friday)
10. Wait for the SCHEDULE_REPORT (Friday)
11. Send a SCHEDULE_GET (Saturday)
12. Wait for the SCHEDULE_REPORT (Saturday)
13. Send a SCHEDULE_GET (Sunday)
14. Wait for the SCHEDULE_REPORT (Sunday)

The operation is visualized in the following sequence diagram:

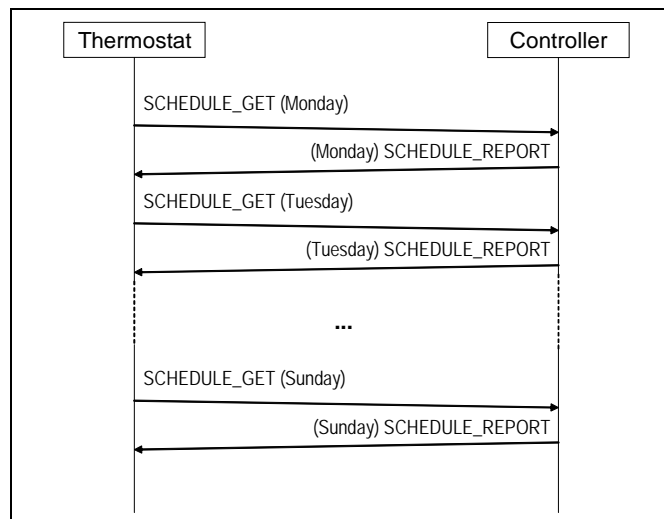


Figure 8. Sequence diagram of operation for getting schedules

5.16.3.5 Setback Thermostat Specific Device Class

The Setback Thermostat device controls the temperature in a room. The thermostat has a setpoint, which is used for normal operation. To save energy it supports the ability to go into setback mode.

The Setback Thermostat device does not support a schedule; a state change must be initiated by the user or another device.

5.16.3.5.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Thermostat Setpoint Command Class
- Thermostat Setback Command Class
- Thermostat Mode Command Class
- Manufacturer Specific Command Class
- Version Command Class

A battery-operated Setback Thermostat device must additionally implement as supported the following command classes:

- Battery Command Class
- Wake Up Command Class Version 2
- Multi Command Command Class

Note: In order to prolong the battery-life of battery-operated devices it is recommended to always send as many commands as possible encapsulated in a Multi Command.

5.16.3.5.1.1 Basic Command Class Implementation

The basic command class shall be implemented in the following way:

Basic Set (Value = 0x00) =	Set Energy Saving Mode
Basic Set (Value = 0xFF) =	Set Comfort Mode
Basic Get =	Get report
Basic Report (Value = 0x00) =	Report Energy Saving Mode
Basic Report (Value = 0xFF) =	Report Comfort Mode

Note: The implementation of Energy Saving Mode is manufacturer specific, and must be documented in the User's Manual. Performing a set using basic commands is always a temporary override; refer to description under Thermostat Setback Command Class.

5.16.3.5.2 Recommended Command Classes to Support

The following command classes are recommended to be supported by the Setback Thermostat:

- Multilevel Sensor (Temperature)
- Thermostat Operation State
- Thermostat Fan
- Thermostat Fan State
- Indicator
- Protection
- Application State
- Association

5.16.3.5.3 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the specific device class level.

5.16.3.6 Setpoint Thermostat Specific Device Class

Setpoint Thermostats control the temperature in one or multiple rooms. It uses a setpoint which defines the desired temperature.

5.16.3.6.1 Mandatory Command Classes to Support

The Setpoint Thermostat device shall support the following command classes:

- Manufacturer Specific Command Class
- Multi Command Command Class
- Thermostat Setpoint
- Version Command Class

A battery-operated Setpoint Thermostat device shall additionally support the following command classes:

- Battery Command Class
- Wake Up Command Class, Version 2

The Setpoint Thermostat shall understand supported commands that are encapsulated by a Multi Command.

Note: In order to prolong the battery-life of battery-operated devices it is recommended to always send as many commands as possible encapsulated in a Multi Command.

Note: The Wake Up Command Class Version 2 allows a controller to learn what wake up intervals a battery-operated device supports.

5.16.3.6.1.1 Basic Command Class Implementation

The basic command class shall be implemented for all permanently listening devices.

The basic command class shall be implemented in the following way:

Basic Set	=	Thermostat Setpoint Set
Basic Get	=	Thermostat Setpoint Get
Basic Report	=	Thermostat Setpoint Report

5.16.3.6.2 Recommended Command Classes to Support

It is recommended for a Setpoint Thermostat device to support the following command classes:

- Association
- Multilevel Sensor

5.16.3.6.3 Mandatory Command Classes to Control

The Setpoint Thermostat shall control the following command classes:

- Multi Command Command Class
- Thermostat Setpoint Command Class

Note: In order to prolong the battery-life of battery-operated devices it is recommended to always send as many commands as possible encapsulated in a Multi Command.

5.16.3.6.4 Mandatory Operation

After Inclusion:

A Setpoint Thermostat has to know the Node ID of the device which contains the schedules.

A Setpoint Thermostat shall perform the following steps after inclusion:

3. Wait for a WAKE_UP_INTERVAL_SET
4. Wait for a WAKE_UP_NO_MORE_INFORMATION

The Setpoint Thermostat must be able to accept other supported commands while performing these steps.

Battery-operated devices shall implement a timeout to allow the device to enter a special mode which preserves battery-life. It must be possible to exit the special mode.

Note: The implementation of exiting the special mode is manufacturer specific.

When exiting the special mode the following steps must be performed:

3. Send a Node Information Frame
4. Perform the steps from the previous list

If timeout occurs again, the device must enter the special mode.

Wake up:

A battery-operated Setpoint Thermostat shall perform the following steps on wake up:

6. Send a BATTERY_REPORT
7. Send a MULTI_LEVEL_SENSOR_REPORT (if implemented)
8. Send a THERMOSTAT_SETPOINT_REPORT
9. Send a WAKE_UP_NOTIFICATION

Step 1 has special considerations:

- Step 1 must be performed on first wake up and may be left out if no new battery measurements were made.

Note: In order to prolong the battery-life of battery-operated devices it is recommended to always send as many commands as possible encapsulated in a Multi Command.

The operation is visualized using Multi Commands in the following sequence diagram:

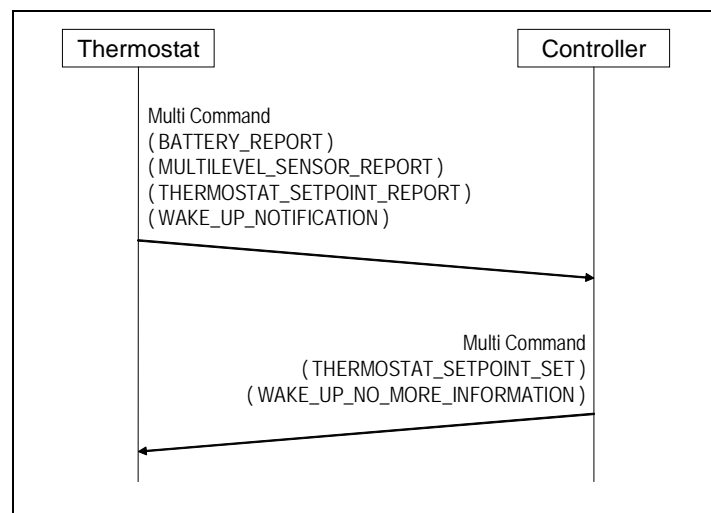


Figure 9. Sequence diagram of operation after wake up

Note: If either device, when using Multi Commands, has the need to send more commands they should try to integrate them into the two Multi Commands exchanged after a wake up as visualized in the above diagram.

Example: A user has pressed a button on the thermostat which makes it wake up. The thermostat then performs the wake up operation but also adds an unsolicited report informing of the user interaction to the Multi Command the thermostat sends to the controller.

5.17 AV Control Point Generic Device Class

The AV Control Point device class is used by AV equipment such as TVs, stereo systems etc.

The AV Control Point Device Class identifier is equal to `GENERIC_TYPE_AV_CONTROL_POINT`.

5.17.1 Mandatory Command Classes to Support

The following command classes to support are defined on the generic device class level:

- Basic Command Class

5.17.1.1 Basic Command Class Implementation

Refer to the specific devices for a description

5.17.2 Mandatory Command Classes to Control

No mandatory command classes to control with are defined on the generic device class level.

5.17.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the AV Control Point Generic Device Class:

Table 3. Specific Device Class identifiers for the AV Control Point Generic Device Class.

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Satellite Receiver	SPECIFIC_TYPE_SATELLITE_RECEIVER
Satellite Receiver V2	SPECIFIC_TYPE_SATELLITE_RECEIVER_V2
Doorbell	SPECIFIC_TYPE_DOORBELL

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.17.3.1 No Specific Device Class defined

This option is selected in case the generic level is appropriate for defining the device.

5.17.3.1.1 Mandatory Command Classes to Support

No mandatory command classes to support are defined on the specific device class level.

5.17.3.1.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set and Basic Get = The Basic Set and basic Get can freely be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Set and Get must be mapped within the same command class. In case the manufacture deems that no relevant commands are available for mapping the command a device receiving these commands must ignore them.

Basic Report = The Basic Report can be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Report must be mapped within the same command class as the Basic Get and must be in accordance with the mapped Get. In case the manufacture deems that no relevant commands are available for mapping the Basic Get command a device is not allowed to respond with a Basic Report under any circumstances.

5.17.3.2 Satellite Receiver Specific Device Class (Not recommended)

It is not recommended to use this specific device class for new devices.

Instead it is recommended to base devices of this category on the Satellite Receiver V2 specific device class.

The Satellite Receiver device can be used as a satellite receiver.

5.17.3.2.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Simple AV Control Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.17.3.2.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set and Basic Get = The Basic Set and basic Get can freely be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Set and Get must be mapped within the same command class. In case the manufacture deems that no relevant commands are available for mapping the command a device receiving these commands must ignore them.

Basic Report = The Basic Report can be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Report must be mapped within the same command class as the Basic Get and must be in accordance with the mapped Get. In case the manufacture deems that no relevant commands are available for mapping the Basic Get command a device is not allowed to respond with a Basic Report under any circumstances.

5.17.3.2.2 Recommended Command Classes to Support

The Screen Meta Data command class can be used in case the device is equipped with a display.

5.17.3.2.3 Mandatory Command Classes to Control

No recommended command classes to control with are defined on the specific device class level.

5.17.3.3 Satellite Receiver V2 Specific Device Class

The Satellite Receiver device can be used as a satellite receiver.

5.17.3.3.1 Mandatory Command Classes to Support

The following command classes to support are defined on the specific device class level:

- Simple AV Control Command Class
- Basic Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.17.3.3.1.1 Basic Command Class Implementation

The basic command class shall be implemented in the following way:

Basic Set (Value = 0x00) =	Device = Off (standby)
Basic Set (Value = 0xFF) =	Device = On / On last volume
Basic Get =	Get report
Basic Report (Value = 0x00) =	Device = Off (standby)
Basic Report (Value = 0xFF) =	Device = On / On last volume

Note: Optional: Volume control can be implemented using Basic Set / Get values 1...99.

5.17.3.3.2 Recommended Command Classes to Support

The Screen Meta Data command class can be used in case the device is equipped with a display.

5.17.3.3.3 Mandatory Command Classes to Control

No recommended command classes to control with are defined on the specific device class level.

5.17.3.4 Doorbell Specific Device Class

A Doorbell Specific Device Class inherits all the commands from the AV Control Point Generic Device Class. The doorbell application accepts a control command that triggers the doorbell to play a sound or any multimedia function that might be attached to the doorbell.

5.17.3.4.1 Mandatory Command Classes to Support

The Doorbell Specific Device Class must support the below command classes.

- **Binary Sensor Command Class**
The Binary Sensor Command Class is used by the doorbell to report an event when the bell has been triggered. A Binary Sensor Report (0xFF) will report a detected event for an application specific period of timeout. Once this timeout has expired and no further event has been detected the Binary Sensor Report must contain (0x00) to indicate an idle state.
- **Association Command Class**
The Association Command Class is be used by the doorbell to configure event subscriptions e.g. unsolicited Binary Sensor Reports to be transmitted when doorbell has been triggered.
- **Manufacture Specific Command Class**
The Manufacture Specific Command Class will advertise manufacture specific information e.g. manufacture id, product id etc.
- **Version Command Class**
The Version Command Class is used to report the library type, protocol version and application version of the device.

5.17.3.4.2 Basic Command Class Implementation

The Basic Command Class must be implemented in the following way:

Basic Set (0x00)	=	Basic Set (0x00). If the doorbell is replaying a ringing tone or application specific sound while receiving a Basic Set (0x00), the replay will be interrupted and stopped instantly. If the doorbell is idle i.e. no replay is active, while receiving Basic Set (0x00) the command must be ignored.
Basic Set (0xFF)	=	Basic Set (0xFF). If the doorbell is idle while receiving Basic Set (0xFF), the doorbell will replay a ringing tone or application specific sound. If replaying is active while receiving Basic Set (0xFF) the doorbell must interrupt the current replay and restart the replay of the ringing tone or application specific sound from start.
Basic Get	=	Binary Sensor Get
Basic Report	=	Binary Sensor Report

5.17.3.4.3 Recommended Command Classes to Support

The following Command Classes can optionally be supported:

- **Alarm Command Class**
Command Class to announce alarm condition.
- **Protection Command Class**
The Doorbell may optionally support the Protection functionality. If the protection functionality is supported, the device shall be able to handle protection related commands.
- **Battery Command Class**
The Battery Command Class can report to the controller about the battery status and level.

5.17.3.4.4 Mandatory Command Classes to Control

No mandatory command classes to control functionality in other devices from the specific device.

5.18 Meter Generic Device Class

The Meter Generic Device Class supports all water meters or energy metering devices (gas, electric etc.) which enable readout of accumulated values and thereby realizing automatic meter reading capabilities.

Automatic meter reading (AMR), is the technology of automatically collecting data from water meter or energy metering devices and transferring that data to a central database for billing and/or analyzing.

The Meter Generic Device Class identifier is equal to `GENERIC_TYPE_METER` and will typically be based on the Basic Device Class identifier `BASIC_TYPE_SLAVE` or `BASIC_TYPE_ROUTING_SLAVE`.

5.18.1 Mandatory Command Classes to Support

The generic device must support the following command classes:

- Basic Command Class

5.18.1.1 Basic Command Class Implementation

Refer to the specific device classes.

5.18.2 Mandatory Command Classes to Control

No mandatory command classes to control functionality in other devices from the generic device.

5.18.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Meter Generic Device Class:

Table 4, Specific Device Class identifiers for the Meter Generic Device Class

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Simple Meter	SPECIFIC_TYPE_SIMPLE_METER

Refer to ZW_classcmd.h source code file for the assigned Specific Device Class identifiers.

5.18.3.1 No Specific Device Class defined

This option is selected in case the generic level is appropriate for defining the device.

5.18.3.1.1 Mandatory Command Classes to Support

No mandatory command classes are specified to access functionality in the specific device itself.

5.18.3.1.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set and Basic Get = The Basic Set and basic Get can freely be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Set and Get must be mapped within the same command class. In case the manufacture deems that no relevant commands are available for mapping the command a device receiving these commands must ignore them.

Basic Report = The Basic Report can be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Report must be mapped within the same command class as the Basic Get and must be in accordance with the mapped Get. In case the manufacture deems that no relevant commands are available for mapping the Basic Get command a device is not allowed to respond with a Basic Report under any circumstances.

5.18.3.2 Simple Meter Specific Device Class

A Simple Meter Specific Device Class inherits all the commands from the Meter Generic Device Class.

The Simple Meter Specific Device Class supports readout of accumulated values in physical units from water meters and energy metering devices. The device supports the following functionalities:

- Meter (mandatory)
- Manufacturer Specific (mandatory)
- Version (mandatory)

5.18.3.2.1 Mandatory Command Classes to Support

In addition to the inherited command classes from the generic device the following must also be supported:

- Meter Command Class
- Manufacturer Specific Command Class
- Version Command Class

5.18.3.2.1.1 Basic Command Class Implementation

The basic command class must be implemented in the following way:

Basic Set = Cannot be mapped because the mapped command class doesn't have a Set command. A device receiving this command must simply ignore it.

Basic Get = Meter Get

Basic Report = Meter Report

5.18.3.2.2 Recommended Command Classes to Support

No recommended command classes supported by the device.

5.18.3.2.3 Mandatory Command Classes to Control

No mandatory command classes to control functionality in other devices from the generic device.

5.18.3.2.4 Recommended Command Classes to Control

No recommended command classes to control functionality in other devices.

5.19 Ventilation Generic Device Class

The Ventilation Generic Device Class should be used for Z-Wave Enabled ventilation systems; it will provide specific device classes to further differentiate between types of ventilation systems.

The ventilation generic device class is equal to `GENERIC_TYPE_VENTILATION`.

5.19.1 Mandatory Command Classes to Support

The following command classes to support are defined on the generic device class level:

- Basic Command Class

5.19.2 Mandatory Command Classes to Control

No mandatory command classes to control are defined on the generic device class level.

5.19.3 Specific Device Classes

The table below shows the current list of defined specific device classes for the Ventilation Generic Device Class:

Table 5 - Specific Device Class identifiers for the Ventilation Generic Device Class.

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	<code>SPECIFIC_TYPE_NOT_USED</code>
Residential Heat Recovery Ventilation	<code>SPECIFIC_TYPE_RESIDENTIAL_HRV</code>

5.19.3.1 No Specific Device Class defined

This option is selected in case the generic level is appropriate for defining the device.

5.19.3.1.1 Mandatory Command Classes to Support

No mandatory command classes to support are defined on the specific device class level.

5.19.3.1.1.1 *Basic Command Class implementation*

The basic command class must be implemented in the following way:

- | | |
|-------------------------|---|
| Basic Set and Basic Get | = The Basic Set and basic Get can freely be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Set and Get must be mapped within the same command class. In case the manufacture deems that no relevant commands are available for mapping the command a device receiving these commands must ignore them. |
| Basic Report | = The Basic Report can be mapped to another command class supported by the device. The mapping must be documented in the User's Manual. The Basic Report must be mapped within the same command class as the Basic Get and must be in accordance with the mapped Get. In case the manufacture deems that no relevant commands are available for mapping the Basic Get command a device is not allowed to respond with a Basic Report under any circumstances. |

5.19.3.2 Residential Heat Recovery Ventilation Specific Device Class

The Residential Heat Recovery Ventilation Specific Device Class should be used for residential ventilation systems that include a heat exchanger.

The device class will mandate the command classes needed to control the ventilation system, and to read out parameters related to the ventilation system.

5.19.3.2.1 Mandatory Command Classes to Support

The following command classes are mandatory to be supported by a residential heat recovery ventilation system.

- HRV Status Command Class, Version 1
 - The HRV Status Command Class allows the ventilation system to report system parameters.
- HRV Control Command Class, Version 1
 - The HRV Control Command Class Version 1 introduces commands for controlling ventilation systems.
- Version Command Class
- Manufacturer Specific Command Class

5.19.3.2.1.1 Basic Command Class implementation

The basic command class must be implemented in the following way:

Basic Set (0x00)	=	Activate energy savings mode
Basic Set (0xFF)	=	Deactivate energy savings mode
Basic Get	=	Get report
Basic Report (0x00)	=	Energy savings mode activated
Basic Report (0xFF)	=	Energy savings mode deactivated

Note: The implementation of the energy saving mode and demand / automatic mode corresponds to the implementations in the HRV Mode Command Class.

5.19.3.2.2 Mandatory Command Classes to Control

No mandatory command classes to control functionality in other devices from the specific device.

5.20 Z/IP Gateway Generic Device Class

The Z/IP Gateway Generic Device Class provides the structural framework for the specific device classes defined in the following sections.

Table 6, Identifier for the Z/IP Gateway Generic Device Class

Generic Device Class	ZW_classcmd.h
Z/IP Gateway	GENERIC_TYPE_ZIP_GATEWAY

5.20.1 Mandatory Command Classes to support

The generic device must support the following command classes:

- Basic Command Class

5.20.1.1 Basic Command Class implementation

The basic command class must be implemented in the following way:

Basic Set = Forwarded to IP host if forwarding of unsolicited frames is enabled.

Basic Get = Forwarded to IP host if forwarding of unsolicited frames is enabled.

Basic Report = Forwarded from IP host in response to Basic Get.

Note that the Z/IP gateway will never respond directly to `Basic Get` commands.

If forwarding of unsolicited frames is enabled, a `Basic Get` command will be forwarded to the IP host specified for unsolicited frames. It is then up to a receiving IP application to respond accordingly.

5.20.2 Mandatory Command Classes to control

No mandatory command classes to control functionality in other devices from the generic device.

5.20.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Z/IP Gateway Generic Device Class:

Table 7, Specific Device Class identifiers for the Z/IP Gateway Generic Device Class

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Z/IP Tunneling Gateway	SPECIFIC_TYPE_ZIP_TUN_GATEWAY
Advanced Z/IP Gateway	SPECIFIC_TYPE_ZIP_ADV_GATEWAY

5.20.3.1 Z/IP Tunneling Gateway Specific Device Class

The Z/IP Tunneling Gateway Specific Device Class inherits all the commands from the Z/IP Gateway Generic Device Class.

The Z/IP Tunneling Gateway Specific Device Class supports IP integration of classic Z-Wave nodes. In addition, the Z/IP Gateway Specific Device Class supports tunneling services for networked applications in a Z/IP node.

The Z/IP Tunneling Gateway will typically be based on the Basic Device Class identifier BASIC_TYPE_STATIC_CONTROLLER.

The “Z/IP Gateway” device class supports

- **IP application control of Z-Wave devices**
 - An IP application may send Z-Wave commands in UDP packets
- **Forwarding of Z-Wave commands to LAN IP applications**
 - The Z/IP gateway emulates a Z-Wave device, so a Z-Wave device may send Z-Wave commands to an IP application by sending Z-Wave commands to the Z/IP gateway.
- **Forwarding of Z-Wave commands to Internet IP applications**
 - The Z/IP gateway emulates a Z-Wave device, so a Z-Wave device may send Z-Wave commands to an IP application by sending Z-Wave commands to the Z/IP gateway via HTTP tunneling.
- **Datagram tunneling to LAN IP applications**
 - A Z-Wave node may create a tunnel from the node to the Z/IP gateway. From the IP side of the Z/IP gateway, datagrams may be sent to an IP application, encapsulated in UDP Z/IP packets.
- **Datagram tunneling to Internet IP applications**
 - A Z-Wave node may create a tunnel from the node to the Z/IP gateway. From the IP side of the Z/IP gateway, datagrams may be sent to an IP application, encapsulated in Z/IP packets in HTTP requests.

5.20.3.1.1 Mandatory Command Classes to support

In addition to the inherited command classes from the generic device the following must also be supported:

- Manufacturer Specific Command Class
- Version Command Class
- Z/IP Tunneling Services Command Class
- Z/IP Tunneling Server Command Class

5.20.3.1.2 Recommended Command Classes to support

No recommended command classes to support.

5.20.3.1.3 Mandatory Command Classes to control

No mandatory command classes to control functionality in other devices.

5.20.3.1.4 Recommended Command Classes to control

No recommended command classes to control functionality in other devices.

5.20.3.2 Advanced Z/IP Gateway Specific Device Class

The Advanced Z/IP Gateway Specific Device Class inherits all the commands from the Z/IP Gateway Generic Device Class.

The advanced Z/IP Gateway Specific Device Class supports IP integration of classic Z-Wave nodes. In addition, the advanced Z/IP Gateway Specific Device Class supports tunneling services for networked applications in a Z/IP node.

Also, Z/IP nodes may exchange IP datagrams with IP hosts outside the Z/IP PAN via normal IP routing.

The Z/IP Gateway will typically be based on the Basic Device Class identifier BASIC_TYPE_STATIC_CONTROLLER.

The “Advanced Z/IP Gateway” device class supports

- **IP communication directly to nodes in the Z-Wave network**
 - The Z/IP Gateway implements true IP routing to a Z/IP node in the Z-Wave network.
- **IP communication directly between nodes in the Z-Wave network**
 - Z/IP nodes may communicate directly within the Z-Wave subnet via Z-Wave mesh routing.
- **IP communication directly from nodes in the Z-Wave network**
 - Z/IP nodes may communicate directly to IP applications in the LAN or in the Internet.
- **IP address assignment**
 - Z/IP nodes may use DHCP to request a LAN IP address, just like Wi-Fi nodes directly becoming members of the LAN IP subnet

5.20.3.2.1 Mandatory Command Classes to support

In addition to the inherited command classes from the generic device the following must also be supported:

- Manufacturer Specific Command Class
- Version Command Class
- Z/IP Tunneling Services Command Class
- Z/IP Tunneling Server Command Class
- Advanced Z/IP Services Command Class
- Advanced Z/IP Server Command Class

5.20.3.2.2 Recommended Command Classes to support

No recommended command classes to support.

5.20.3.2.3 Mandatory Command Classes to control

No mandatory command classes to control functionality in other devices.

5.20.3.2.4 Recommended Command Classes to control

No recommended command classes to control functionality in other devices.

5.21 Z/IP Node Generic Device Class

The Z/IP Node Generic Device Class provides the structural framework for the specific device classes defined in the following sections.

Table 8, Identifier for the Z/IP Node Generic Device Class

Generic Device Class	ZW_classcmd.h
Z/IP Node	GENERIC_TYPE_ZIP_NODE

While the Z/IP Node Generic Device Class is defined as all other Z-Wave generic device classes, it is special in that it should never be signaled as the main property of a Z-Wave node.

The device type of a Z-Wave node signaled in the Node Information frame must reflect the primary function of the node. Thus, the device type signaled may be `GENERIC_TYPE_METER_PULSE`, `GENERIC_TYPE_ENTRY_CONTROL` or other device classes representing the purpose of the node.

A Z/IP node must signal its support for the `GENERIC_TYPE_ZIP_NODE` and specific device classes by listing all the supported command classes in the command class list of the Node Information frame.

One exception exists. A device may signal the `GENERIC_TYPE_ZIP_NODE` device class if the node provides no functionality to other Z-Wave devices but only uses the Z-Wave infrastructure to access external IP resources.

An example may be a stock ticker which every 5 minutes wakes up, requests updated stock quotes and goes back to sleep.

5.21.1 Mandatory Command Classes to support

No mandatory command classes to support.

5.21.2 Mandatory Command Classes to control

No mandatory command classes to control functionality in other devices from the generic device.

5.21.3 Specific Device Classes

The table below shows the current list of defined Specific Device Classes for the Z/IP Node Generic Device Class:

Table 9, Specific Device Class identifiers for the Z/IP Node Generic Device Class

Specific Device Class	ZW_classcmd.h
Specific Device Class not used	SPECIFIC_TYPE_NOT_USED
Z/IP Tunneling Node	SPECIFIC_TYPE_ZIP_TUN_NODE
Advanced Z/IP Node	SPECIFIC_TYPE_ZIP_ADV_NODE

While the Z/IP Tunneling Node Specific Device Classes are defined as all other Z-Wave specific device classes, they are special in that they should never be signaled as the main property of a Z-Wave node.

The device type of a Z-Wave node signaled in the Node Information frame must reflect the primary function of the node. Thus, the device type signaled may be `GENERIC_TYPE_METER_PULSE`, `GENERIC_TYPE_ENTRY_CONTROL` or other device classes representing the purpose of the node.

A Z/IP Tunneling Node should signal its support for the `SPECIFIC_TYPE_ZIP_TUN_NODE` or the `SPECIFIC_TYPE_ZIP_ADV_NODE` class by listing all the supported command classes in the command class list of the Node Information frame.

One exception exists. A device may signal the `GENERIC_TYPE_ZIP_NODE` and `SPECIFIC_TYPE_ZIP_*_NODE` device classes if the node provides no functionality to other Z-Wave devices but only uses the Z-Wave infrastructure to access external IP resources. An example may be a stock ticker which every 5 minutes wakes up, requests updated stock quotes and goes back to sleep.

5.21.3.1 Z/IP Tunneling Node Specific Device Class

The Z/IP Tunneling Node Specific Device Class inherits all the commands from the Z/IP Node Generic Device Class.

The Z/IP Tunneling Node supports tunneling services for networked applications.

The “Z/IP Tunneling Node” device class supports

- **Datagram tunneling to LAN IP applications**
 - A Z/IP tunneling node may create a tunnel from the node to the Z/IP gateway. From the IP side of the Z/IP gateway, datagram's may be sent to an IP application, encapsulated in UDP Z/IP packets.
- **Datagram tunneling to Internet IP applications**
 - A Z/IP tunneling node may create a tunnel from the node to the Z/IP gateway. From the IP side of the Z/IP gateway, datagram's may be sent to an IP application, encapsulated in Z/IP packets in HTTP requests.

5.21.3.1.1 Mandatory Command Classes to support

In addition to the inherited command classes from the generic device the following must also be supported:

- Manufacturer Specific Command Class
- Version Command Class
- Z/IP Tunneling Services Command Class
- Z/IP Tunneling Client Command Class

5.21.3.1.2 Recommended Command Classes to support

No recommended command classes to support.

5.21.3.1.3 Mandatory Command Classes to control

No mandatory command classes to control functionality in other devices.

5.21.3.1.4 Recommended Command Classes to control

No recommended command classes to control functionality in other devices.

5.21.3.2 Advanced Z/IP Node Specific Device Class

The Advanced Z/IP Node Specific Device Class inherits all the commands from the Z/IP Node Generic Device Class.

The advanced Z/IP Gateway Specific Device Class supports tunneling services for networked applications.

Also, Z/IP nodes may exchange IP datagrams with other Z/IP nodes as well as IP hosts outside the Z/IP PAN via normal IP routing.

The “Advanced Z/IP Node” device class supports

- **IP communication directly between nodes in the Z-Wave network**
 - Z/IP nodes may communicate directly within the Z-Wave subnet via Z-Wave mesh routing.
- **IP communication directly from nodes in the Z-Wave network**
 - Z/IP nodes may communicate directly to IP applications in the LAN or in the Internet.
- **IP address assignment**
 - Z/IP nodes may use DHCP to request a LAN IP address, just like Wi-Fi nodes directly becoming members of the LAN IP subnet

5.21.3.2.1 Mandatory Command Classes to support

In addition to the inherited command classes from the generic device the following must also be supported:

- Manufacturer Specific Command Class
- Version Command Class
- Z/IP Tunneling Services Command Class
- Z/IP Tunneling Client Command Class
- Advanced Z/IP Services Command Class
- Advanced Z/IP Client Command Class

5.21.3.2.2 Recommended Command Classes to support

No recommended command classes to support.

5.21.3.2.3 Mandatory Command Classes to control

No mandatory command classes to control functionality in other devices.

5.21.3.2.4 Recommended Command Classes to control

No recommended command classes to control functionality in other devices.

6 COMMAND CLASSES

For a detailed description of the command classes and associated commands refer to [8].

7 DEVICE EXAMPLES

The following sections describe application examples using the Z-Wave device and command classes.

7.1 Outlet adapter with dimming capability

This example deals with an outlet adapter with dimming capability as shown on the figure.



Figure 10. Outlet adapter with dimming capabilities

A slave library is selected as the basic device class because initiating transmission to other devices is not necessary. The listening flag is set because the slave is mains powered allowing it to respond immediately on requests from other devices in the Z-Wave network. To allow the device to turn power on/off and dim it must comply with the Multilevel Power Switch Device Class that are a Specific Device Class created on the Generic Multilevel Switch Device Class. The selected device class supports the mandatory command classes Multilevel Switch and All Switch. The Version command class is also mandatory because Version 2 of the Multilevel Switch command class is used to allow dimming commands specifying the dimming duration/rate. No recommended command classes are selected.

7.1.1 Node Information

The controller will assign a Node ID to the device during the registration phase. When the node is being told to register it will issue a Node Information frame. The registration is part of the protocol and will be handled automatically. The Node Information frame that is sent will have the layout shown below.

7	6	5	4	3	2	1	0
Listening = YES	Protocol Specific Part						
Protocol Specific Part							
Protocol Specific Part							
Basic Device Class = BASIC_TYPE_SLAVE (Protocol Specific Part)							
Multilevel Switch Device Class = GENERIC_TYPE_SWITCH_MULTILEVEL							
Multilevel Power Switch Device Class = SPECIFIC_TYPE_POWER_SWITCH_MULTILEVEL							
Multilevel Switch Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
All Switch Command Class = COMMAND_CLASS_SWITCH_ALL							
Version Command Class = COMMAND_CLASS_VERSION							

7.1.2 Switch Functionality

When the device is registered to the network, the controller can begin to control the switch. The controller will use the Multilevel Switch commands and the All Switch commands to control the device. During the registration to the network the node will be assigned a Node ID. The controller uses this Node ID when it wants to address the node.

7.1.2.1 Switching On and Off

The following describes how the controller can turn the switch on and off. The controller uses the Multilevel Switch commands.

Switching on

The controller can switch the device on by using the Multilevel Switch Set command with the following layout.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_SET							
Value = SWITCHED_ON							

Switching off

The controller can switch the device off by using the Multilevel Switch Set command with the following layout.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_SET							
Value = SWITCHED_OFF							

Checking on and off

If the controller requests the switch whether it is turned on or off, it can use the Multilevel Switch Get command with the following layout.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_GET							

The switch device will then answer with a Multilevel Switch Report command with the following layout (assuming the device is switched off).

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_REPORT							
Value = SWITCHED_OFF							

7.1.2.2 Dimming

The following describes how the controller can set a specific dim level or dim up and down. The controller uses the Multilevel Switch commands.

Setting dim level

The controller can set a specific dim level on the device by using the Multilevel Switch Set command with the following layout.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_SET							
Value (0x24)							
Dimming Duration (0x02)							

In the example above the dim level is set to 36 (0x24) and it takes 2 seconds (0x02) before the specified level is reached.

Checking dim level

The controller can request the dim level from the switch by using the Multilevel Switch Get command with the following layout.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_GET							

The switch device will then answer with a Multilevel Switch Report command with the following layout (assuming the device is at dim level 78 (0x4E)).

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_REPORT							
Value (0x4E)							

Dimming

The controller can tell the switch to start dimming the power down by using the Multilevel Switch Start Level Change command with the following layout.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_START_LEVEL_CHANGE							
Re- ser- ved (0b)	Up/ Down (0b)	Ignore Start Level (1b)	Reserved (00000b)				
Start Level (0x00)							
Dimming Duration (0x05)							

The switch device will then start decreasing the dim level starting from the actual level. This will continue until the controller sends a command that tells the device to stop or until the device reaches the level 0. The dimming duration specify that it will take 5 seconds (0x05) to dim from level 0 to 99. The dimming rate is therefore equal to $99 \text{ levels} / 5 \text{ seconds} = 19.8 \text{ levels per second}$.

The controller stops dimming by the Multilevel Switch Stop Level Change command with the following layout.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_STOP_LEVEL_CHANGE							

7.1.2.3 All On/All Off

The switch can participate in all on or all off actions.

The controller can request all switches to be turned on by using the All Switch On command with the following layout.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_ALL							
Command = SWITCH_ALL_ON							

The controller can also request all switches to be turned off by using the All Switch Off command with the following layout.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_ALL							
Command = SWITCH_ALL_OFF							

A switch can be excluded from the all on/all off functionality by using the All Switch Set command with the following layout.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_ALL							
Command = SWITCH_ALL_SET							
Mode = ALL_SWITCH_EXCLUDE_ON_OFF							

7.2 Battery-powered rocker switch

This example illustrates a battery-powered rocker switch for lighting control as shown on the figure.



Figure 11. Battery-powered rocker switch

The battery-powered rocker switch can be placed directly on the wall without any additional wiring. A routing slave library is selected as the basic device class because the device must be capable of initiating transmission to a limited number of other devices. The battery-powered rocker switch must have assigned return routes to all the controllable destinations to obtain reliable RF communication. The listening flag is cleared because the slave is battery-powered. To allow the device to turn power on/off and dim light in other devices it must comply with the Multilevel Remote Switch Device Class that are a Specific Device Class created on the Remote Switch Generic Device Class. When the ON button is pressed on the switch it will turn the light on in the devices it has been associated with and off when the OFF button is pressed. When the ON button is held down on the switch it will start dimming up and the OFF button result in dimming down. The selected device class has the mandatory command class Multilevel Switch it can control in other devices. In addition are the Association, Battery and Wake Up command classes selected. The Association command class is used to specify the devices the rocker switch must be capable of initiating transmission to. The Battery command class is used to sent unsolicited frames in case a battery low is detected. The Wake Up command class allows the rocker switch to wake up occasionally and notify another device with respect to pending information.

7.2.1 Node Information

The controller will assign a Node ID to the device during the registration phase. When the node is being told to register it will issue a Node Information frame. The registration is part of the protocol and will be handled automatically. The Node Information frame that is sent will have the layout shown below.

7	6	5	4	3	2	1	0
Listening = NO	Protocol Specific Part						
Protocol Specific Part							
Protocol Specific Part							
Basic Device Class = BASIC_TYPE_ROUTING_SLAVE (Protocol Specific Part)							
Remote Switch Device Class = GENERIC_TYPE_SWITCH_REMOTE							
Multilevel Remote Switch Device Class = SPECIFIC_TYPE_SWITCH_REMOTE_MULTILEVEL							
Association Command Class = COMMAND_CLASS_ASSOCIATION							
Wake Up Command Class = COMMAND_CLASS_WAKE_UP							
Battery Command Class = COMMAND_CLASS_BATTERY							
Support/Control Mark = COMMAND_CLASS_MARK							
Multilevel Switch Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							

Notice the COMMAND_CLASS_MARK that separate command classes supported by the device itself and command classes it can control in other devices.

7.2.2 Rocker Switch Functionality

In the following sections are it shown how the command classes notified by the node information frame are used and the functionality they provide.

7.2.2.1 Association

The associations are typically created and maintained by a portable remote. The following describes how the associations are created and maintained.

Groups Supported

The rocker switch can only control other nodes via the Multilevel Switch command class and therefore is only one group adequate. Groups supported are requested as follows:

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_GROUPINGS_GET							

The rocker switch will respond as follows:

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_GROUPINGS_REPORT							
Supported Groupings (0x01)							

Nodes in a Group Supported

Now can the number of nodes in a given group be determined. The rocker switch supports only one group so the number of nodes must be requested from grouping identifier one:

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_GET							
Grouping Identifier (0x01)							

The rocker switch will respond as follows:

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_REPORT							
Grouping Identifier (0x01)							
Max Nodes Supported (0x05)							
Reports to Follow (0x00)							

The rocker switch supports up to five destinations because it is not possible to assign return routes to more destinations in a routing slave library. Notice that no node ID's are returned because the associations have not yet been created at this stage.

Configuring Associations

The rocker switch with node ID equal to 0x08 must control five outlet dimmers with the node ID's 0x02, 0x03, 0x04, 0x05 and 0x06 assigned to group one. This is done by the Association Set command as shown below:

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_SET							
Grouping identifier (0x01)							
Node ID1 (0x02)							
Node ID2 (0x03)							
Node ID3 (0x04)							
Node ID4 (0x05)							
Node ID5 (0x06)							

In addition to the above command must the following API calls be executed to assign the return routes for the five outlet dimmers:

```
ZW_DeleteReturnRoute( 0x08, callback_func );
```

```
ZW_AssignReturnRoute( 0x08, 0x02, callback_func );
```

```
ZW_AssignReturnRoute( 0x08, 0x03, callback_func );
```

```
ZW_AssignReturnRoute( 0x08, 0x04, callback_func );
```

```
ZW_AssignReturnRoute( 0x08, 0x05, callback_func );
```

```
ZW_AssignReturnRoute( 0x08, 0x06, callback_func );
```

Managing Associations

In case the outlet dimmer with node ID equal to 0x05 is to be replaced by a new with node ID equal to 0x09. Then it is necessary to be sure that 0x05 is among the current associations by sending the command:

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_GET							
Grouping Identifier (0x01)							

The rocker switch will respond as follows:

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_REPORT							
Grouping Identifier (0x01)							
Max Nodes Supported (0x05)							
Reports to Follow (0x00)							
Node ID1 (0x02)							
Node ID2 (0x03)							
Node ID3 (0x04)							
Node ID4 (0x05)							
Node ID5 (0x06)							

It is also necessary to know all the associations before the return routes can be assigned.

Now is the old outlet dimmer remove by the command:

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_REMOVE							
Grouping Identifier (0x01)							
Node ID4 (0x05)							

Then is the new outlet dimmer added by the command:

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_SET							
Grouping identifier (0x01)							
Node ID1 (0x09)							

and finally assignment of the return routes for the five outlet dimmers:

```
ZW_DeleteReturnRoute( 0x08, callback_func );
```

```
ZW_AssignReturnRoute( 0x08, 0x02, callback_func );
```

```
ZW_AssignReturnRoute( 0x08, 0x03, callback_func );
```

```
ZW_AssignReturnRoute( 0x08, 0x04, callback_func );
```

```
ZW_AssignReturnRoute( 0x08, 0x06, callback_func );
```

```
ZW_AssignReturnRoute( 0x08, 0x09, callback_func );
```

7.2.2.2 Wake Up

The battery-operated device is typically asleep unable to receive commands to minimize battery consumption. In the following it is shown how a battery-operated device can wake up occasionally to obtain information from another device.

Wake Up Configuration

A device can configure the battery-operated device with wake up interval and node ID of the device to receive I'm awake notifications by using the Wake Up Interval Set command. In case the device must wake up every 6 hours and notify another device with node ID equal to 0x07 then the layout is as follows.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WAKE_UP							
Command = WAKE_UP_INTERVAL_SET							
Seconds3 (MSB) = 0x00							
Seconds2 = 0x54							
Seconds1 (LSB) = 0x60							
Node ID = 0x07							

It is also possible to request the configuration parameters by using the Wake Up Interval Get command and the battery-operated device will then return a Wake Up Interval Report command.

Wake Up Sequence

The battery-operated device uses the Wake Up Notification command to notify another device that it is awake. When the Wake Up Notification command is received then the two devices are ready to exchange information.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WAKE_UP							
Command = WAKE_UP_NOTIFICATION							

The rocker switch can broadcast the wake up configuration until it gets the device destination via the Wake Up Interval Set command.

The notified device uses the Wake Up No More Information command to tell the battery-operated device to go back to sleep to minimize battery consumption.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WAKE_UP							
Command = WAKE_UP_NO_MORE_INFORMATION							

7.2.2.3 Battery

The wake up notified device can request the battery level from the rocker switch when awake by using the Battery Level Get command with the following layout.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BATTERY							
Command = BATTERY_GET							

The rocker switch device will then answer with a Battery Level Report command with the following layout (assuming the battery level is 95% (0x5F) of the full battery).

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BATTERY							
Command = BATTERY_REPORT							
Value (0x5F)							

The rocker switch can also send an unsolicited battery low using the following format:

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BATTERY							
Command = BATTERY_REPORT							
Value (0xFF)							

This allows the battery operated rocker switch to notify about the critical battery level immediately and not wait until the next Wake Up notification.

7.2.2.4 Switching On/Off and Dimming

Refer to the previously example regarding the outlet adapter with dimming capability with respect to the functionality available in the Multilevel Switch command class. In this example does the rocker switch not support this command class, but have the capability to control the Multilevel Switch command class in other devices.

REFERENCES

- [1] Zensys, INS10638, Z-Wave Certification Overview.
- [2] Z-Wave Alliance, ZAD10485, Device/Command Class Development Process.
- [3] Zensys, INS10247, Z-Wave ZW0102/ZW0201/ZW0301 Application Programming Guide.
- [4] Zensys, SDS11059, Chimney Fan Device Class.
- [5] Zensys, SDS11058, Thermostat Heating Device Class.
- [6] Zensys, INS10244, Z-Wave Node Type Overview and Network Installation Guide.
- [7] Zensys, APL10720, Programming the ZW0201 Flash from Internal MCU.
- [8] Zensys, SDS11060, Z-Wave Command Class Specification.

INDEX

A

Accumulated values.....	136
Advanced Door Lock Specific Device Class	104
Advanced Z/IP Gateway Specific Device Class	147
Advanced Z/IP Node Specific Device Class.....	151
Advanced Zensor Net Alarm Sensor Specific Device Class	31
Advanced Zensor Net Smoke Sensor Specific Device Class	41
Alarm Sensor Generic Device Class	20
All Switch Off command	158
All Switch On command	158
All Switch Set command.....	158
Application example.....	153
Automatic meter reading	136
AV Control Point Generic Device Class	129

B

Basic Command Class	8
Basic Device Classes	7, 10
Basic Repeater Slave Specific Device Class	64
Basic Routing Alarm Sensor Specific Device Class.....	23
Basic Routing Smoke Sensor Specific Device Class	33
Basic Zensor Net Alarm Sensor Specific Device Class	27
Basic Zensor Net Smoke Sensor Specific Device Class.....	37
Battery Level Get command	165
Battery Level Report command.....	165
Battery-powered rocker switch	159
Binary Power Switch Specific Device Class	46
Binary Remote Switch Specific Device Class	85
Binary Scene Switch Specific Device Class	48
Binary Sensor Generic Device Class	87
Binary Switch Generic Device Class	43, 85
Binary Toggle Remote Switch Specific Device Class	86
Binary Toggle Switch Generic Device Class	86
Binary Toggle Switch Specific Device Class	81
Blinds	111

C

Certification program	2
Command Class field.....	8
Command Classes	6

D

Device Classes	5
Device/command class development process	2
Display Generic Device Class	95
Door Lock Specific Device Class.....	103
Doorbell Specific Device Class.....	134
Drapes	111

E

Energy metering devices	136
-------------------------------	-----

Energy Production Specific Device Class	110
Enhanced Slave.....	14
Entry Control Generic Device Class	100

G

Generic Device Class field	8
Generic Device Classes	15

I

Inclusion controller.....	13, 14
Interoperability	2

L

Listening flag.....	8
---------------------	---

M

Mark Command Class	8
Meter Generic Device Class	136
Metering devices.....	136
Motor Control Class A Specific Device Class.....	73
Motor Control Class B Specific Device Class.....	75
Motor Control Class C Specific Device Class	77
Multilevel Power Switch Specific Device Class	69, 153
Multilevel Remote Switch Specific Device Class.....	85, 159
Multilevel Scene Switch Specific Device Class	70
Multilevel Sensor Generic Device Class.....	90
Multilevel Switch Generic Device Class	66, 85, 153
Multilevel Switch Get command	155, 156
Multilevel Switch Report command	155, 156
Multilevel Switch Set command.....	154, 155
Multilevel Switch Start Level Change command	157
Multilevel Switch Stop Level Change command	157
Multilevel Toggle Remote Switch Specific Device Class.....	86
Multilevel Toggle Switch Generic Device Class	86
Multilevel Toggle Switch Specific Device Class	82
Multiposition Motor Specific Device Class.....	72

N

Node Information Frame.....	7, 154, 160
-----------------------------	-------------

O

Optional Functionality flag	8
Outlet adapter with dimming capability.....	153

P

PC Controller Specific Device Class	58
Portable controller.....	11
Portable Installer Tool Specific Device Class.....	55
Portable Remote Controller Specific Device Class	52
Portable Scene Controller Specific Device Class.....	53
Primary controller.....	11, 12, 13, 14
Pulse Meter Generic Device Class.....	93

R

Remote Controller Generic Device Class.....	50
Remote Switch Generic Device Class.....	83, 159
Repeater Slave Generic Device Class	62
Residential Heat Recovery Ventilation Specific Device Class	142
Router	12, 13, 14
Routing Alarm Sensor Specific Device Class.....	25
Routing Binary Sensor Specific Device Class.....	89
Routing Multilevel Sensor Specific Device Class	92
Routing Slave	14
Routing Smoke Sensor Specific Device Class.....	35

S

Satellite Receiver Specific Device Class	132, 133
Scene Controller Specific Device Class	59
Secondary controller.....	11, 12
Secure Keypad Door Lock Specific Device Class.....	105
Semi Interoperable Generic Device Class	107
Setback Thermostat Specific Device Class.....	124
Shades.....	111
Simple Display Specific Device Class	98
Simple Meter Specific Device Class.....	139
Simple Window Covering Control Specific Device Class	112
Slave	13
Specific Device Class field	8
Specific Device Classes	15
Static controller	12
Static Controller Generic Device Class	56
Static Installer Tool Specific Device Class	61

T

Thermostat General Specific Device Class.....	116
Thermostat Generic Device Class.....	113
Toggle Switch Generic Device Class	79

V

Ventilation Generic Device Class	140
--	-----

W

Wake Up Interval Set command.....	164
Wake Up No More Information command	165
Wake Up Notification command	165
Water meters	136
Window Covering Generic Device Class.....	111

Z

Z/IP Gateway Generic Device Class	143
Z/IP Node Generic Device Class.....	148
Z/IP Tunneling Gateway Specific Device Class	145
Z/IP Tunneling Node Specific Device Class.....	150
Zensor Net Alarm Sensor Specific Device Class	29
Zensor Net Smoke Sensor Specific Device Class	39
ZW_classcmd.h	2
Z-Wave logo	2

