



Instruction

Z-Wave Node Type Overview and Network Installation Guide

Document Part Number:	952006802
Document Revision:	1
Description:	This document describes how to setup a network with regard to inclusion of nodes in the network, operation and exclusion of nodes from the network.
Written By:	JFR;SML
Date:	2005-04-13
Reviewed By:	CMO;JFR;NTJ;PSH;RBO
Restrictions:	No restrictions

Approved by:

Date	CET	Initials	
2005-04-13	14:09:29	NTJ	

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 other than to conduct technical evaluation. 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	20050118	SML JFR	ALL	Revised document. Added SUC and SIS. Primary Controller does not use Node ID 0xEF anymore.

Table of Contents

1	ABBREVIATIONS.....	1
2	INTRODUCTION.....	2
2.1	Purpose.....	2
2.2	Backward compatibility.....	2
2.3	Audience and prerequisites.....	2
3	Z-WAVE BASICS.....	3
3.1	Network Nodes.....	3
3.1.1	Controller Nodes.....	3
3.1.1.1	Portable Controller.....	3
3.1.1.2	Installation Controller.....	3
3.1.1.3	Static Controller.....	3
3.1.1.4	Bridge Controller.....	4
3.1.2	Slave Nodes.....	4
3.1.2.1	Slave.....	4
3.1.2.2	Routing Slave.....	5
3.1.2.3	Enhanced Slave.....	5
3.2	Home ID.....	5
3.3	Node ID.....	5
3.4	Routing.....	7
3.5	Initiators.....	7
3.5.1	Slave Initiators.....	7
3.5.2	Controller Initiators.....	7
3.6	Node Information Frame.....	8
4	Z-WAVE NETWORK FEATURES.....	9
4.1	Include/exclude Process.....	9
4.1.1	Include Nodes.....	9
4.1.1.1	Include slaves.....	9
4.1.1.2	Include controllers.....	9
4.1.2	Exclude Nodes.....	9
4.2	Association of nodes.....	10
5	BASIC INSTALLATION EXAMPLE.....	11
5.1	Initial Setup.....	11
5.2	Associating Nodes.....	11
5.3	Adding Slave Nodes.....	12
5.4	Adding Controllers.....	13
5.5	Removing Slaves.....	14
5.6	Removing Secondary Controllers.....	14
5.7	Recover from Controller Node Error.....	14
6	STATIC CONTROLLER AND ROUTING SLAVE EXAMPLE.....	16
6.1	Including Routing Slaves.....	16
6.2	Including Static Controllers.....	18
7	STATIC UPDATE CONTROLLER (SUC) EXAMPLE.....	20
7.1	Including a SUC to the network.....	20
8	SUC ID SERVER (SIS) EXAMPLE.....	21
8.1	Including a SIS to the network.....	21

9 REFERENCES.....23

Table of Figures

Figure 1. Assigning Home ID and Node ID	6
Figure 2. Unique Home ID for two adjacent homes	6
Figure 3. Router frame flow	7
Figure 4. Including a Slave node	11
Figure 5. Associating a Slave Node with a Controller	11
Figure 6. Including Slave node 2	12
Figure 7. Associating Slave Node 2 with the Controller	12
Figure 8. Including a second Controller	13
Figure 9. Including and associating additional Slave nodes	14
Figure 10. Including a Routing Slave	16
Figure 11. Associating a "Simple" Slave with a Routing Slave	17
Figure 12. Controller with three associated Routing Slaves	18
Figure 13. Including a Static Controller	18
Figure 14. Associating a Static Controller with a Routing Slave	19
Figure 15. Including a SUC to the network	20
Figure 16. Including a SIS to the network	21
Figure 17. After SIS is included to the network	21
Figure 18. Including an Inclusion Controller to the network	22

1 ABBREVIATIONS

Abbreviation	Explanation
SIS	SUC ID Server
SUC	Static Update Controller

2 INTRODUCTION

2.1 Purpose

The purpose of this document is to provide guidelines for installation, maintenance and operation of a Z-Wave network consisting of controller and slave nodes. Refer to [1] regarding how the functionality is implemented in the described devices.

2.2 Backward compatibility

The latest Developer's Kit contains new features to improved installation flexibility of a Z-Wave network. Therefore is it important to understand these features in detail to insure backward compatibility with Z-Wave enabled products build on older Developer's Kit releases.

From Developer's Kit v3.3x is the Static Update Controller (SUC) introduced to allow slave and controller nodes to request network topology updates.

From Developer's Kit v3.4x can the SUC in addition also function as a node ID server (NIS) to allow other controllers to include/exclude nodes to/from the network. Furthermore is the unique node ID 0xEF for primary controllers obsolete.

2.3 Audience and prerequisites

The audience is considered to be OEM's implementing the Z-Wave technology into their products.

3 Z-WAVE BASICS

This chapter describes the basic building blocks of the Z-Wave technology.

3.1 Network Nodes

The Z-Wave network consists of two different types of network nodes; controllers and slaves. The controller nodes are able to calculate routes (and alternative routes). The second node type is the slave node, which generally acts as input and output units. Both types exist in different versions as described below. The Z-Wave protocol supports networks of up to 232 nodes, which can be freely shared between controller and slave nodes.

3.1.1 Controller Nodes

A controller in the Z-Wave terminology is defined as a unit that has the ability to host a routing table of the entire network and calculate routes on the basis thereof. Moreover, the controller has the ability to pass on routes to slave units, in order to enable them to transmit routed signals.

The first controller that is installed in the network is by default configured to act as Primary Controller with license to include/exclude nodes. The Primary Controller is used to include all subsequent nodes in the network.

The controller exists in three fundamentally different versions. These are Portable Controllers, Static Controllers and Installation Controllers. In the current form of the Z-Wave technology, a Z-Wave network needs at least one Portable Controller. More Portable Controllers as well as Static Controllers can be added as needed as the network grows.

A Static Controller can be enabled to become a Static Update Controller (SUC), which adds advanced self-organization functionalities to the network. A SUC can furthermore be enabled to become a SUC Id Server (SIS), which adds more flexibility to the installation process.

3.1.1.1 Portable Controller

The Portable Controller has the ability to discover its own position in the network, when it needs to communicate with other nodes. An example of a device using this type could be a remote control unit, e.g. for controlling light or HVAC systems. The Portable Controller is typically used to include/exclude nodes and maintaining the Z-Wave network.

3.1.1.2 Installation Controller

The Installation Controller is essentially a Portable Controller node, which incorporates extra functionality that can be used to implement controllers especially targeted towards professional installers which needs extended network diagnostics.

3.1.1.3 Static Controller

The Static Controller is required to remain in a fixed position in the network, meaning that it should not be physically moved when it has been included in the network. Moreover it is required that the Static Controller is always in "listening mode".

The "always listening" advantage of the Static Controller is that other nodes can transmit frames to it whenever needed, both for uploading purposes as well as for consulting purposes.

The Static Controller has two important network features used for more advanced installations. Both controllers are described more thoroughly in the installation example chapters (4, 5):

Static update Controller (SUC):

When a Static Controller is configured as a SUC the primary Portable or Installer controller automatic send network updates to the SUC e.g. when a new node is included to the network the node will automatically appear in the SUC's topology map. There can only be one SUC in each individual network. Other controller types in the network will individually request the SUC for network updates. If no SUC is present Primary Controller is responsible of updating all controller in the network, if necessary. The SUC is capable of creating a new Primary Controller in case the original Primary Controller is lost or malfunctioning.

There can only be one SUC in each individual network.

SUC ID Server (SIS):

When a SUC in addition is configured as a node ID server (SIS) it enables all controller types to include/exclude nodes. The SIS becomes automatically the Primary Controller in the network when enabled.

There can only be one SIS in each individual network.

3.1.1.4 Bridge Controller

The Bridge has capabilities to have devices from other network types like X10 and UPnP appear in a Z-Wave network as virtual nodes. This enables control of Z-Wave nodes from e.g. an X10 controller or vice versa.

3.1.2 Slave Nodes

The slave nodes are devices that do not contain routing tables. The so-called Routing Slave nodes and Enhanced Routing Slaves nodes however can contain a number of pre-configured routes (assigned to them by a controller).

Any slave node can act as repeater for frames going to other nodes. It is however the originating transmitter node that decides which nodes are actually used for repeating a particular frame.

The only requirement for being able to act as repeater is that the node is in listening state. This requires that the node is always on, and in order to save on battery consumption, this mean that only mains-powered nodes will act as repeaters in most practical installations.

Battery operated slaves that do not listen continually are disregarded by controllers when they calculate routes.

There are three types of slave nodes.

3.1.2.1 Slave

The slave node type is able to receive frames and reply if necessary. The slave node cannot host pre-configured routes to other nodes. The slave node is typically used for devices that only require input (and report status if polled) and do not generate frames unsolicited. An example of a device using this type could be a power outlet.

3.1.2.2 Routing Slave

The Routing Slave can host a number of routes to a slave or controller. Such routes are called "Controller assigned routes". Routing Slaves can use these routes to communicate with either controllers or other slave nodes. The Routing Slave can either be A/C powered or battery powered. If the Routing Slave is A/C powered it can be used as a repeater in the Z-Wave network, otherwise it will be disregarded when routes are calculated. The Routing Slave functionality is used for devices that need to report unsolicited status or alarms. Examples of devices using this type could be a movement sensor.

3.1.2.3 Enhanced Slave

The enhanced slave has the same basic functionality as routing slave node, but because of more features on the hardware more software components are available. Enhanced slave nodes have software support for External EEPROM and an RTC. An example of a device using this type of Slave nodes could be a Thermostat.

3.2 Home ID

The Z-Wave protocol uses a unique identifier called the Home ID to separate networks from each other. A 32 bits unique identifier is pre programmed in all controller nodes at manufacturing. This unique 32 bits identifier is automatically used as Home ID during the installation of the Z-Wave network. If more than one controller is used in the network it is the 32 bits unique identifier of the first controller node used that governs the Home ID. Additional controller nodes will be assigned that same Home ID when included into the network. This is an automatic process that leaves the initial controller as the "Primary Controller" and the other controller(s) as "Secondary Controller(s)".

All slave nodes in the network will initially have a Home ID that is set to zero. In order to communicate in the network the Primary Controller need to assign them with its Home ID. It is only a Primary Controller or an Inclusion Controller that can assign Home ID's to a node.

3.3 Node ID

Node ID's are used to address individual nodes in a network. A Node ID is an 8 bits value and it is assigned to slave and controller nodes by a Primary Controller or an Inclusion Controller. The Node ID assigned to a device is only unique within a network defined by the unique Home ID.

See figure 1 and 2 for a graphical representation of above.

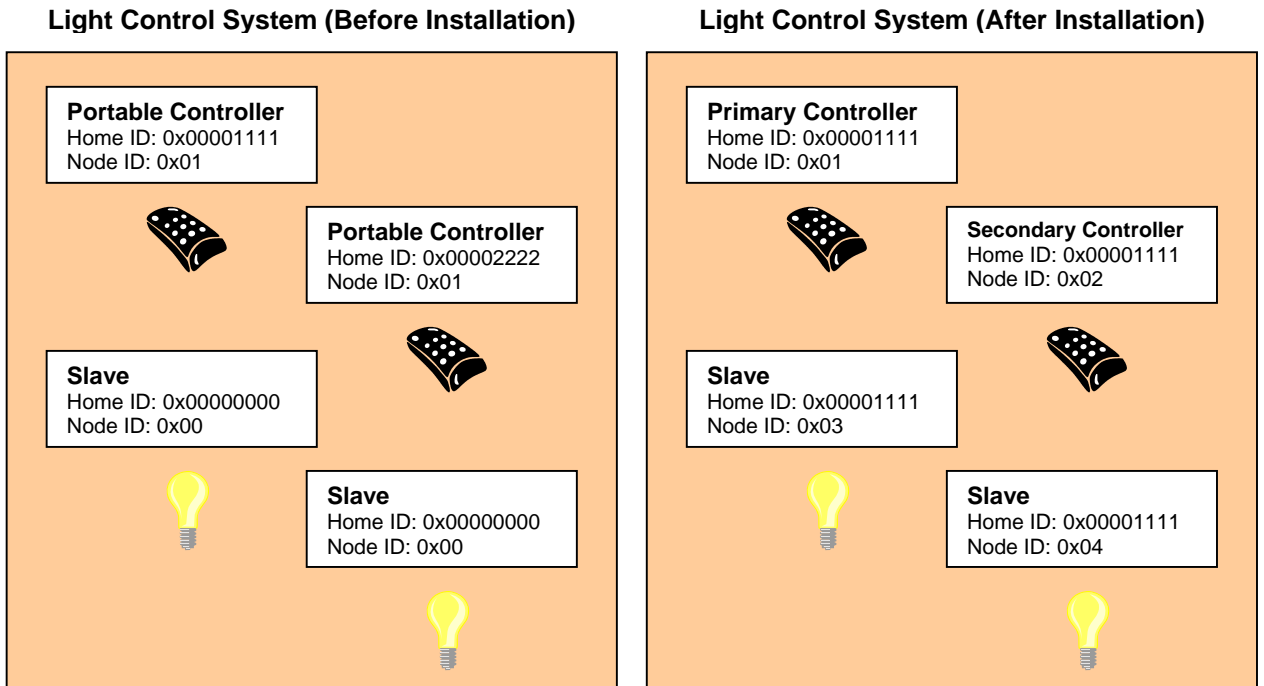


Figure 1. Assigning Home ID and Node ID

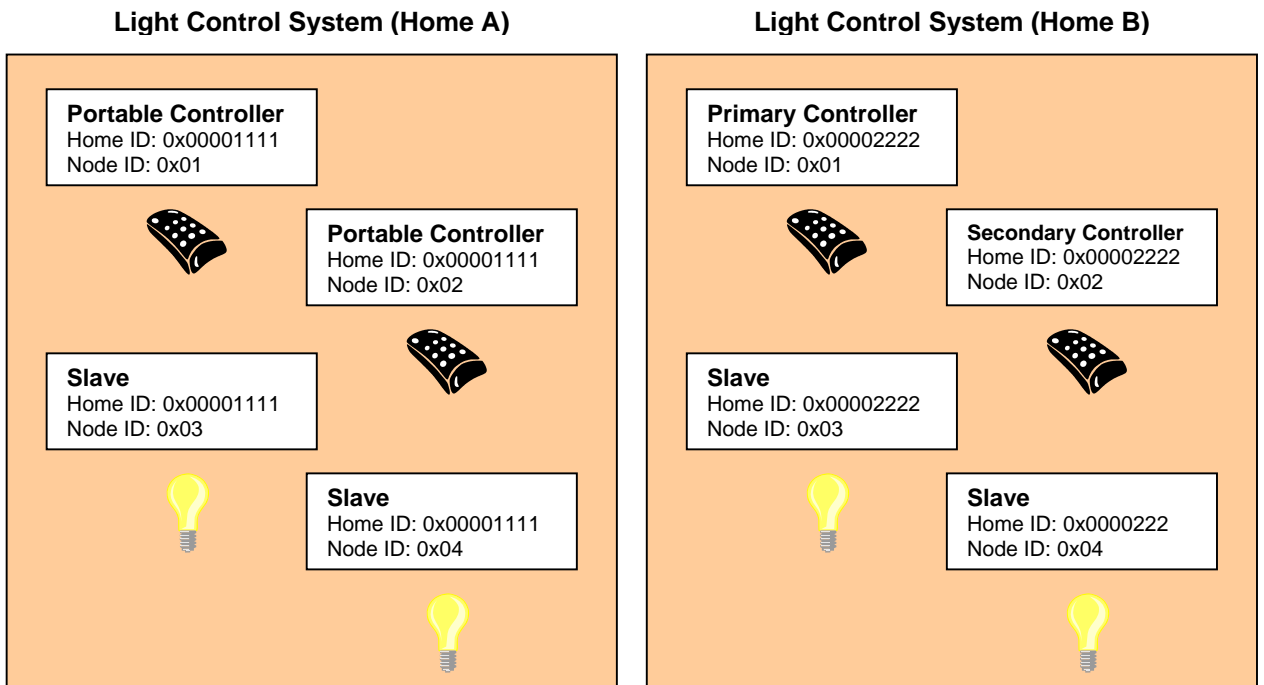


Figure 2. Unique Home ID for two adjacent homes

3.4 Routing

Any controller contains a routing table, which enables the controller to calculate routes in the Z-Wave network. Slave nodes will not have the ability to initiate transmission of routed frames, unless the controller has provided one or more controller assigned routes to the slave.

The figure below shows an example of the frame flow when a frame is sent from a controller, repeated through a slave, to a second slave. When the frame is transmitted (1) it is treated with normal retransmission and acknowledge (2), but the controller also waits for a route acknowledge indicating that the frame arrived to the final destination. The node that repeats the frame (3) in the network handles acknowledge (4) and retransmission. After sending the acknowledge frame (4), the slave will also send a router acknowledge frame (5) that is followed by an acknowledge frame (6) from Slave 1. Slave 1 repeats the router acknowledge frame (7) and receives an acknowledge frame from the controller (8).

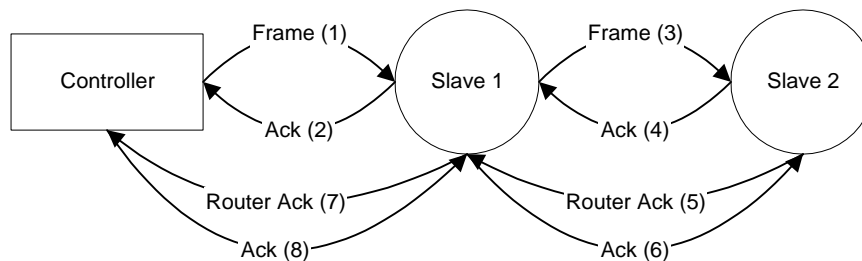


Figure 3. Router frame flow

3.5 Initiators

The Z-Wave system is relying on the ability to include, exclude and operate nodes in the network, as described below. This requires that each device have a number of "Initiators". In the rest of this document an initiator does not necessarily mean a physical button. An Initiator can either be a physical button, a special activation of a button (e.g. activating a button for 2 seconds), a combination of buttons (e.g. activating two buttons simultaneously) or an item in a menu system.

3.5.1 Slave Initiators

The slave nodes only need one Initiator since all the intelligence is placed in the controller nodes. The slave initiator will be used for inclusion/exclusion, association/disassociation and operation depending on whether the initiators on the controller are activated or not. If additional features need to be supported, additional initiators may be required.

3.5.2 Controller Initiators

The controller nodes need at least the following initiators:

Include Initiator:

The include initiator is used by Primary Controllers (when a SIS is not present) to include nodes in the network. When both the include initiator and a slave initiator have been activated simultaneously, the new node will be included to the network if the node was not included previously. Include initiator is activated on a Primary Controller or Inclusion Controller and on a controller that is un-initialized, this will result in the new controller is included to.

Exclude Initiator:

The exclude initiator is used by Primary Controllers to exclude nodes from the network. When the exclude initiator and a slave initiator are activated simultaneously, it will result in the slave being excluded from the network (and reset to Home ID zero). If the slave was not part of the network it will still be reset by this action.

Associate Initiator:

The associate initiator is used to associate nodes when both initiators on two nodes have been activated, the two nodes will be associated. This has the effect that when activating the operate initiator, all devices associated with this node is being operated.

Operate Initiator:

The operate initiator is used for controlling the functionality of the associated devices (e.g. on/off switch for power outlets).

Controller Assigned Route Initiator:

The controller assigned route initiator is only needed in systems where the Primary Controller is required to generate controller assigned routes for Routing Slaves. The initiator is used to associate a Routing Slave with Static Controllers and other slaves.

If additional features need to be supported, additional initiators may be required.

3.6 Node Information Frame

When the Operate Initiator in a node is activated it will issue a node information frame. This frame is part of the Z-Wave protocol, and specifies the capabilities of the node. These capabilities are the node type, whether the node is able to repeat frames, and other protocol relevant issues. The node information frame also contains the Home ID and the Node ID.

It is possible for the application to ask for the Node Information Frame from all nodes in the network and hence enabling any node to acquire information regarding any nodes features in the network at any given time.

4 Z-WAVE NETWORK FEATURES

This chapter contains information about functionalities in a Z-Wave Network.

4.1 Include/exclude Process

All nodes are required to take part in the include/exclude process. This process encompasses the ability to include and exclude nodes in the network. Only the Primary Controller and Inclusion Controllers in the network are able to include and exclude nodes. This strategy is chosen to ensure that the network stays consistent.

From the Installer viewpoint all Z-Wave nodes (regardless of their node type) will use the same installation process.

4.1.1 Include Nodes

When a node is to be included in the network, the Primary Controller or Inclusion Controller will grant the request if there is Node ID's available.

4.1.1.1 Include slaves

The Include process is initiated by both activating the include initiator on the controller and the slave initiator. When the node information frame is received from an un-initialized node the controller will assign Home ID and Node ID to the slave. If a slave has already been assigned a Home ID and Node ID, it will not accept the inclusion. The Node will have to be excluded (or reset) before the node can be included again.

4.1.1.2 Include controllers

The include process is initiated by activating the include initiator on the Primary Controller or Inclusion Controller and on the new controller that should be included into the network. The Primary Controller or Inclusion Controller will assign Home ID and Node ID to the new controller, which will then automatically become a Secondary or Inclusion Controller.

As part of the inclusion of additional controllers in the network, a replication of the topology information tables and optionally other information will automatically take place. This ensures that the new controller have the newest information available. At later stages, when the Primary Controller has been updated with new nodes (or nodes have been deleted), a new replication can be initiated by activating the include initiators on both controllers. This network information updating can also be done automatically using the Static Update Controller Functionality.

4.1.2 Exclude Nodes

The exclusion of Slave nodes happens when the both exclude initiator on the Primary Controller and the slave initiator are activated. When nodes are excluded it will result in the node being reset and the topology information in the Primary Controller or Inclusion Controller being updated to reflect the change.

When a slave node is reset, it will assume the Home ID zero again. When a controller is reset, it will assume the Home ID it was originally assigned during the manufacturing process.

The exclusion of a Controller happens when the both exclude initiator on the Primary and the Secondary Controllers are activated. When the Controller is reset, it will assume the Home ID it was originally assigned during manufacturing process.

4.2 Association of nodes

When two nodes have been assigned to the network it is possible to associate these with each other. The following describes the different possibilities for associating nodes.

Associating a Slave with a Controller

A slave node is associated with a controller by activating both the associate initiator on the controller and the initiator on the slave node. The controller will receive the node information from the slave node and add the slave node to the associated nodes list

Once a slave has been included to the network (see above) then it is possible to create associations with all Controllers.

Associating a Routing Slave with a (Routing) Slave

A Routing Slave is associated with another (Routing) Slave by using a Controller. The association of the nodes takes place by first activating the controller assigned route initiator on the Controller and the initiator on the destination Node (node that should be controlled). Secondly the controller assigned route initiator on the Controller and the initiator on the source node (node that should be controlling the association) are activated simultaneously. The Controller will then generate one or more routes and transmit these to the source node.

Associating a Static Controller with a Routing Slave

A Static Controller is associated with a Routing Slave by use of the Controller. The association of the nodes takes place by first activating the controller assigned route initiator on the Controller and the associate initiator on the Static Controller (node that should be controlled or informed about events). Secondly the controller assigned route initiator on the Controller and the initiator on the Routing Slave (node that should be controlling the association) are activated. The Controller will then generate one or more routes and send them to the source node.

5 BASIC INSTALLATION EXAMPLE

This chapter contains some examples of how to install elements of a Z-Wave system in a sequential order. It also describes how nodes are associated.

5.1 Initial Setup

The most basic setup is one controller node and one slave node, as depicted in figure 4.

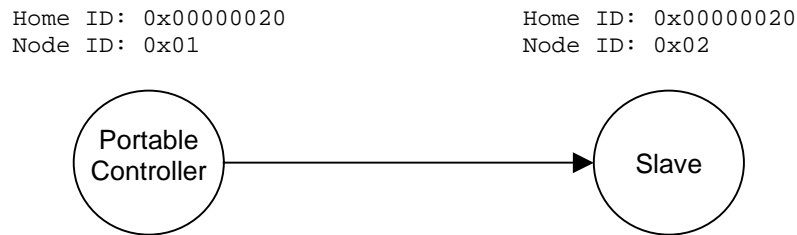


Figure 4. Including a Slave node.

Initially both the controller and the slave node are reset. The controller will have a predefined Home ID = 0x00000020) and a Node ID = 0x00. The slave have a predefined Home ID = 0x00000000 and Node ID = 0x00. Being the first controller in the system and therefore determining the Home ID, the controller defaults to become the Primary Controller and takes node ID = 0x01.

When the include initiator on the controller and the slave Initiator is activated simultaneously, the include procedure is initiated. The controller assigns the Home ID = 0x00000020 and the Node ID = 0x02 to the slave node. The slave is now part of the Z-Wave network, but no specific association has been made between the two nodes (the inclusion and the association process can be done as one interlinked action from the end-user's perspective).

5.2 Associating Nodes

The two nodes in figure 5 are included in the same Z-Wave network, but have not yet been associated.

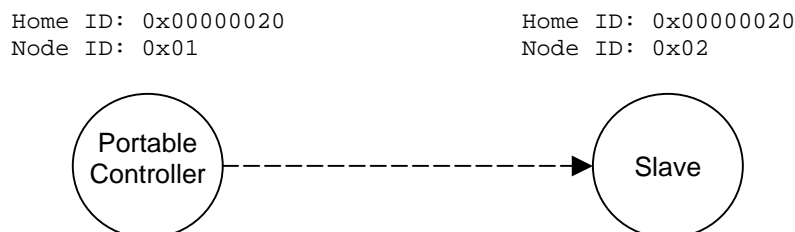


Figure 5. Associating a Slave Node with a Controller.

If the two nodes should be associated, this can happen by activating the associate initiator on the controller and the slave initiator simultaneously. This will cause the controller to listen for a node to associate. When the slave sends a node info frame (specified by the Z-Wave protocol and prompted by activating the slave initiator), the controller will include the slave in the associated slaves list.

Every time the operate initiator on the controller is activated, the controller will send a message to the slave node, which will cause the slave node to perform the specified action.

5.3 Adding Slave Nodes

New slave nodes can be included in the Z-Wave network as the network is gradually expanded. In figure 4, the example is extended to include another slave node. The inclusion happens the exact same way as for the first slave node. The second slave node will be assigned the Home ID = 0x00000020 and the Node ID = 0x03. The second slave node can also be associated with the controller by activating the associate initiator on the controller and the slave initiator simultaneously. The result is depicted in figure 6.

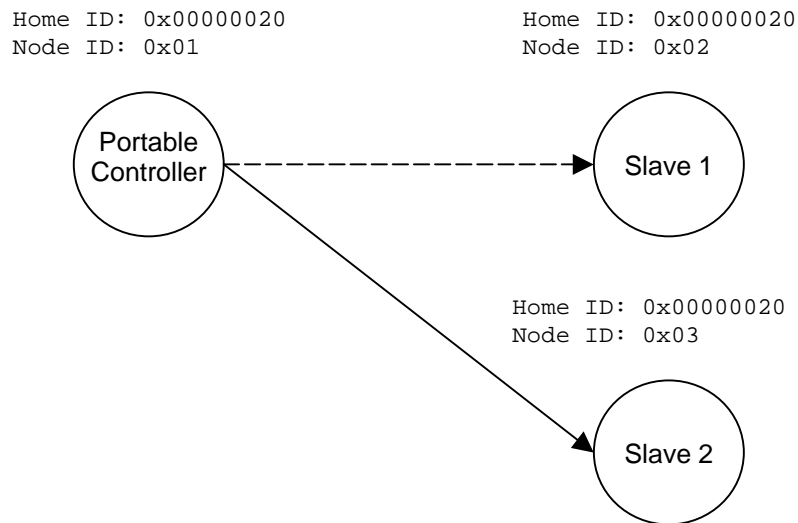


Figure 6. Including Slave node 2.

The second slave node can also be associated with the controller by activating the associate initiator on the controller and the slave initiator simultaneously as shown on figure 7.

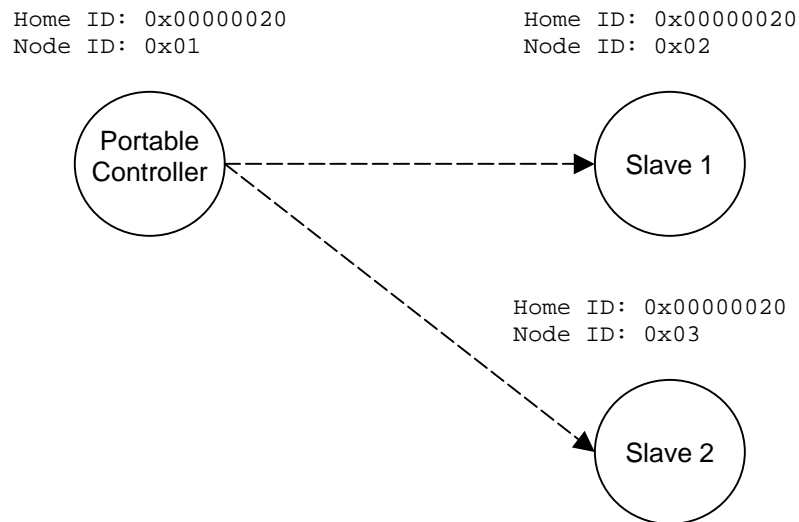


Figure 7. Associating Slave Node 2 with the Controller.

5.4 Adding Controllers

The Z-Wave network can also be extended to include multiple controllers. Figure 8 shows a system with 2 controllers.

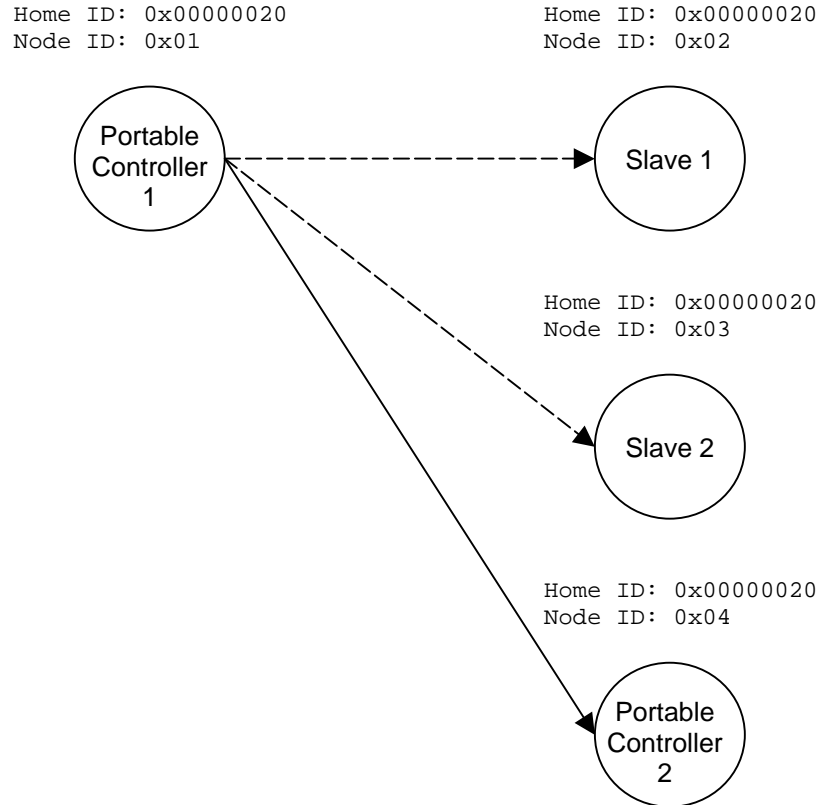


Figure 8. Including a second Controller.

The second controller to be added comes with another unique default Home ID, for example 0x00000030 and Node ID = 0x00. When the Include initiator is activated on both controllers, the include process is initiated. The first controller, which acts as Primary Controller for the Z-Wave network, will assign the Home ID = 0x00000020 and the Node ID = 0x04 to the second controller, which now assumes the position as a Secondary Controller.

Following the exchange of network identifiers a replication will take place. During the replication process, all routing information will be transferred from the Primary Controller to the Secondary Controller. The Secondary Controller is now able to perform all the same functionalities as the Primary Controller except from including/excluding nodes in the Z-Wave network.

Additional slave nodes can be assigned to the Z-Wave network by the Primary Controller and associated with either controller 1, controller 2 or both. In figure 7 below two slave nodes has been included with the Primary Controller and associated with controller 2.

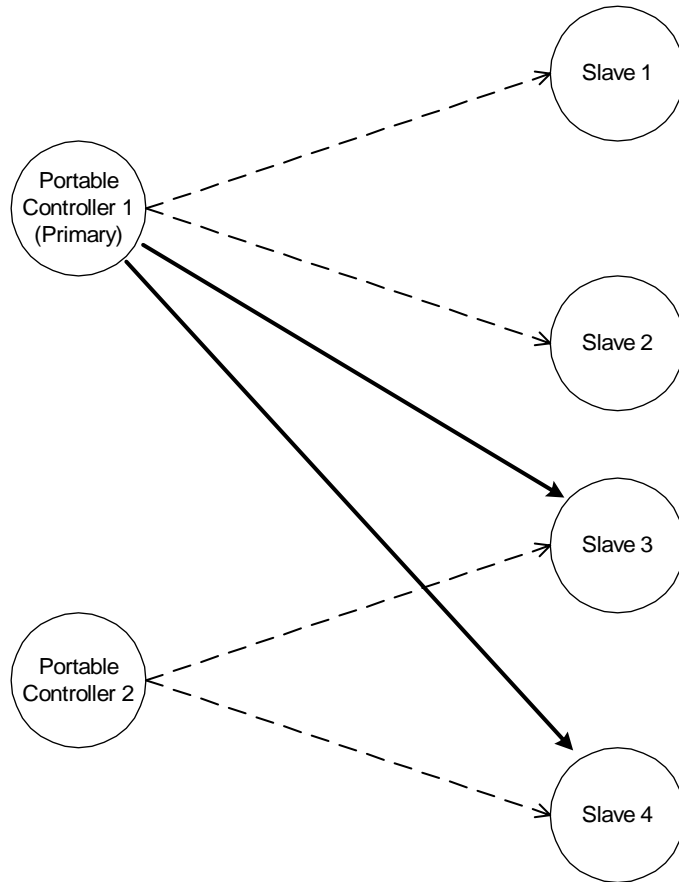


Figure 9. Including and associating additional Slave nodes.

The Portable Controller 2 can be associated with slave 3 and slave 4. The routing table of Portable Controller will not be updated before Portable Controller 1 performs a controller replication.

5.5 Removing Slaves

A slave node can be removed from the Z-Wave network by activating the exclude initiator on the Primary Controller and the slave initiator. This action will cause the controller to send Home ID = 0x00000000 and Node ID = 0x00 to the slave, thereby resetting the slave. The controller will also remove the routing information from the routing table.

5.6 Removing Secondary Controllers

A Secondary Controller can be removed by activating the exclusion initiator on both the Primary and the Secondary Controllers to be removed.

5.7 Recover from Controller Node Error

If a controller fails, recovery depends on whether it is a Primary Controller or a Secondary Controller. If it is a Secondary Controller it can just be removed, a new can be included and the routing information can be replicated.

If the failed controller is the Primary Controller the situation is different. Basically the entire network can be reset and then reconfigured with a new controller. This controller will then become the new Primary Controller. The unique 32 bits identifier of the new Primary Controller will now be the new Home ID for the entire Z-Wave network.

A more advanced method is to use the Network management features of the Z-Wave Network.

If a Static Controller is configured as a SUC it will keep an updated version of the routing table. All controllers in the system will request updates from the SUC. The primary controller is required to send updates to the SUC when new nodes have been included or old nodes have been reset. If the Primary Controller malfunctions, a secondary controller can be requested to be primary by making replication from the SUC.

6 STATIC CONTROLLER AND ROUTING SLAVE EXAMPLE

This chapter extends the previous example to also include Static Controllers and Routing Slaves. The starting point for this example is a network with a Portable Controller and two slave nodes.

6.1 Including Routing Slaves

The first step in this scenario is to include a Routing Slave, see figure 8.

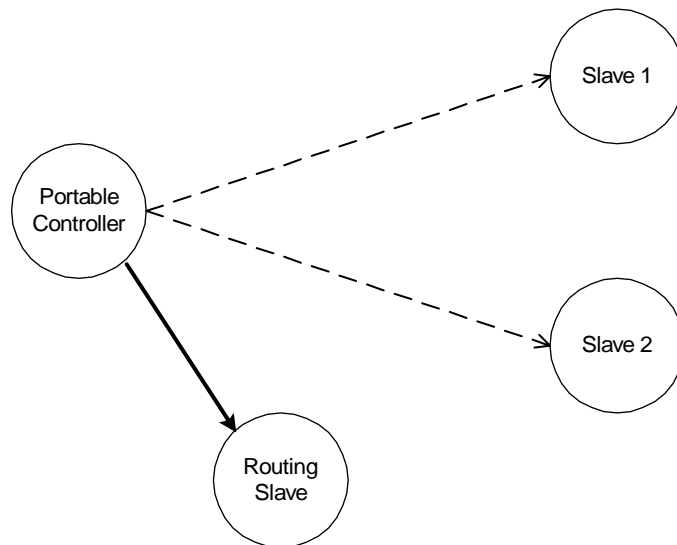


Figure 10. Including a Routing Slave.

The Routing Slave is included in the network in the exact same way as any other slave node. When the include initiator on the controller and the initiator on the Routing Slave are activated, the controller will assign Home ID and Node ID to the Routing Slave.

The Routing Slave can now be associated with a controller or another slave. The controller is an important part of facilitating the association process although it is not part of the final association.

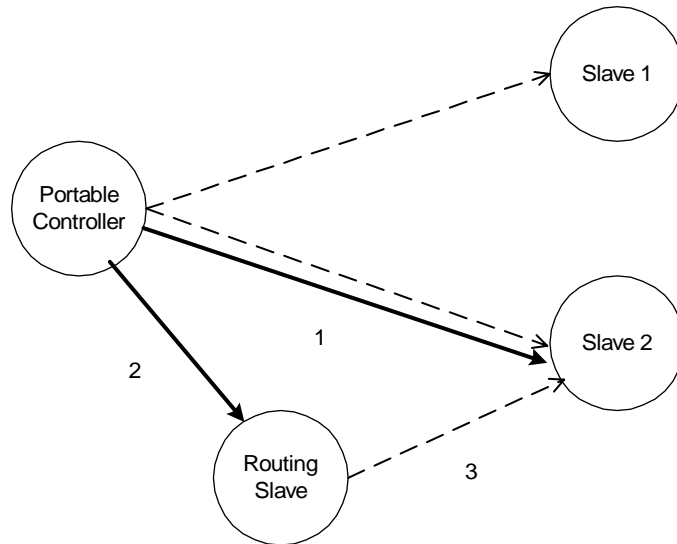


Figure 11. Associating a “Simple” Slave with a Routing Slave.

The controller needs an extra initiator compared to the previous example. This initiator is called a controller assigned route initiator. The controller assigned route initiator on the controller and the initiator on slave 2 (the device to be controlled) are activated. The procedure is repeated for the Routing Slave. Upon doing the second part of the association, the controller will calculate one or more routes between the Routing Slave and Slave 2. These routes will be sent to the Routing Slave, which will use the routes when communicating with Slave 2.

6.2 Including Static Controllers

This part of the example describes a situation where the network initially consists of one controller and three Routing Slaves. This network is depicted in figure 10.

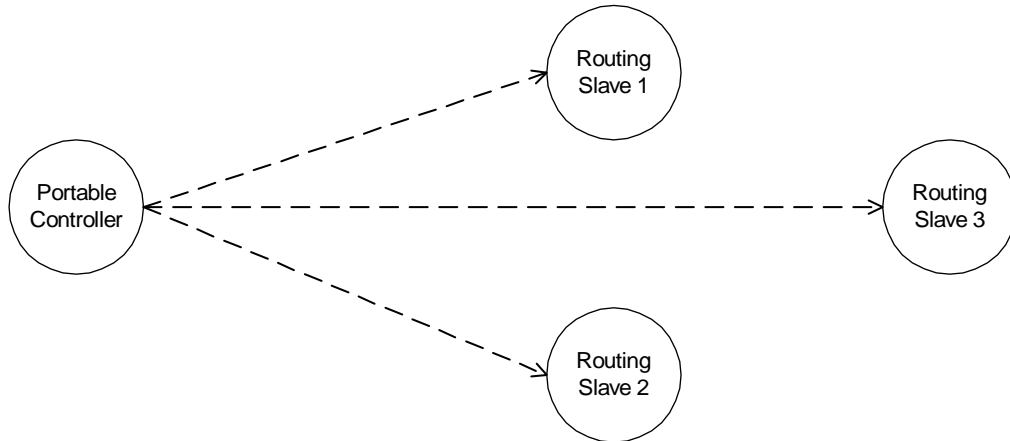


Figure 12. Controller with three associated Routing Slaves.

The following figure shows the previous example extended with a Static Controller. A Static Controller is included in the same way as a Portable Controller. The Static Controller will be assigned a Home ID and a Node ID, and the topology information is transferred. The main difference in the inclusion procedure is that the Static Controller automatically investigates which neighbours it is able to see during the include process. Figure 13 illustrates the network after the Static Controller has been included.

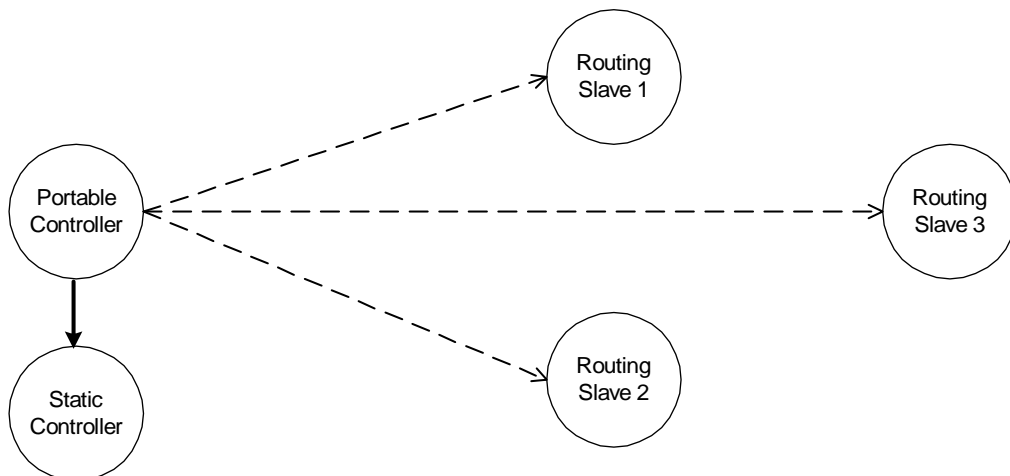


Figure 13. Including a Static Controller.

The Portable Controller now contains the full routing information, including the knowledge of which nodes that can be reached directly from the Static Controller.

In this scenario it is now possible to create an association that allows a Routing Slave to communicate with the Static Controller.

In the following example we want to associate the Static Controller with the Routing Slave 3. In the example it is assumed that the Static Controller can only reach Routing Slave 1 and Routing Slave 2

directly. Also for the Routing Slave 3 it is assumed that it is able to reach Routing Slave 1 and Routing Slave 2 directly, but not the Static Controller.

The Association is done as described in the previous example, by first moving the Portable Controller within range of the Static Controller, and then activating the static route initiator on the Portable Controller and the associate initiator on the Static Controller. Secondly the Portable Controller is moved closer to Routing Slave 3. When the associate initiator on the Portable Controller and the initiator on slave 3 have been activated, the Portable Controller will calculate routes from Routing Slave 3 to the Static Controller. In this example the Portable Controller will come up with two alternative routes. The first route is to reach the Static Controller through Routing Slave 1. The second route is to reach the Static Controller through Routing Slave 2.

Figure 14 shows the Z-Wave network after the Static Controller has been associated with Routing Slave 3.

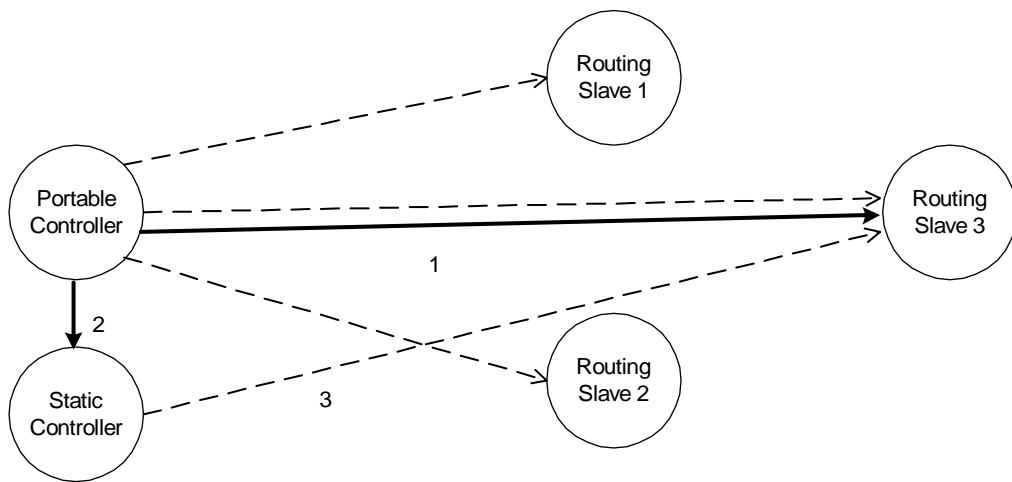


Figure 14. Associating a Static Controller with a Routing Slave.

7 STATIC UPDATE CONTROLLER (SUC) EXAMPLE

When a Static Controller is configured as a Static Update Controller (SUC) then it will automatically receive all changes in the network topology from the Primary Controller. Other nodes in the network can acquire this information.

Note! Only one SUC can be present in each individual network.

7.1 Including a SUC to the network

The following step describes the process of including a SUC to the network:

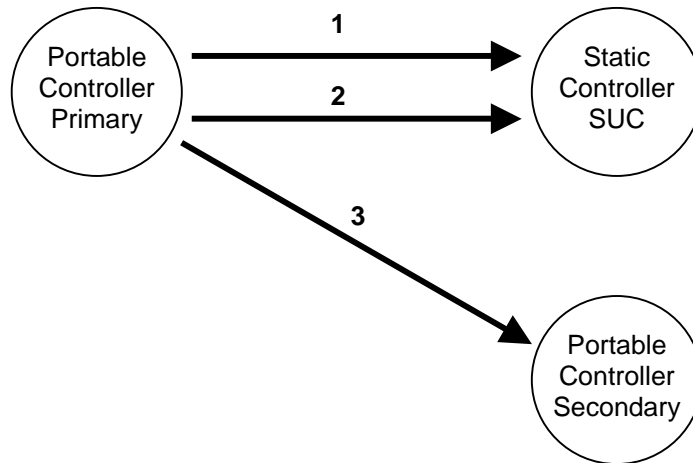


Figure 15. Including a SUC to the network.

- 1) Include the Static Controller as described in the previous chapter by replicating the information from the Primary Controller to the Static Controller. The Static Controller is now included to the network and is secondary. Afterwards will the application on the Primary Controller assign the Secondary Static Controller to become the SUC. It is recommended to include the SUC as the first node in the network because new nodes can then be advised about its presence when included.
- 2) The Primary Controller includes an additional Portable Controller node to the network in same way as described earlier in previous chapters.
- 3) The Primary Controller will automatically send the updated network topology for the included Secondary Portable Controller to the SUC. From now on every time a new node (controller or slave) is included to the network the SUC will get the updates automatically.

Association is done exactly as described earlier in previous chapter.

8 SUC ID SERVER (SIS) EXAMPLE

The SUC can in addition be configured with node ID server (SIS) functionality to enhance installation flexibility. The SIS enables other controllers to include/exclude nodes to/from the network. When the SIS is configured then it takes at the same time the role as the Primary Controller because it both have the latest network topology and allocate node IDs. All the other controllers are called Inclusion Controllers because they can include/exclude nodes to/from the network. Until the SIS is included to the network the installation process is as described in the previous installation chapters.

Note! Only one SIS can be present in each individual network.

8.1 Including a SIS to the network

The following step describes the process of including a SIS to the network.

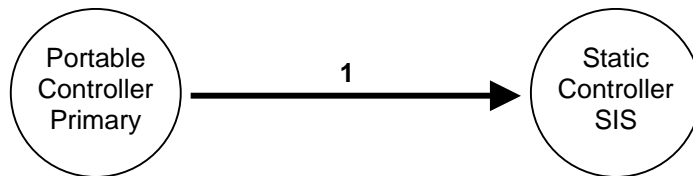


Figure 16. Including a SIS to the network.



Figure 17. After SIS is included to the network.

1. Include the Static Controller as described in the previous chapter by replicating the information from the Primary Controller to the Static Controller. The Static Controller is now included to the network and is secondary. Afterwards will the application on the Primary Controller assign the Secondary Static Controller to become the SIS. It is recommended to include the SIS as the first node in the network because new nodes can then be advised about its presence when included. The SIS becomes the Primary Controller after inclusion and the old Primary Controller becomes an Inclusion Controller.

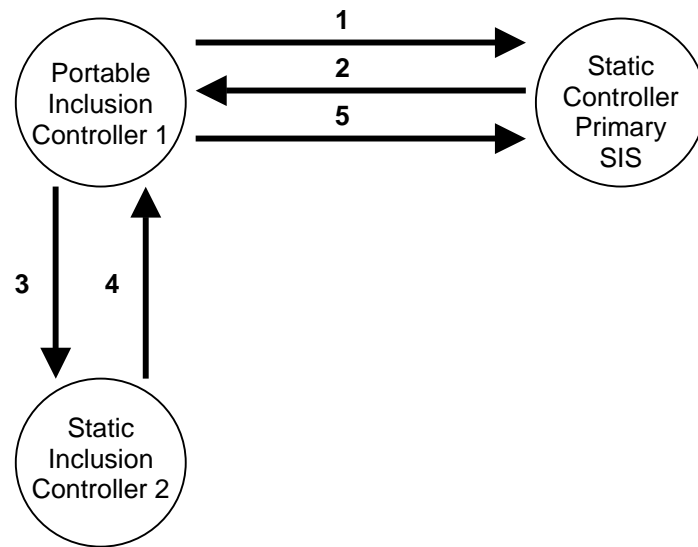


Figure 18. Including an Inclusion Controller to the network.

1. Inclusion Controller 1 requests an available node ID for the new Static Inclusion Controller 2 and the latest network topology from the SIS
2. The SIS provides a node ID and the latest network topology to the Inclusion Controller 1.
3. Inclusion Controller 1 now assigns the allocated node ID to Static Inclusion Controller 2. Based on the latest network topology the Static Inclusion Controller 2 check for neighbors within direct range. Check for neighbors within direct range is not done in case the Inclusion Controller 2 is portable
4. The Inclusion Controller 2 returns neighbor information to Inclusion Controller 1. In case the new Static Controller is based on an old library without Inclusion Controller capability then it becomes a Secondary Static Controller.
5. The Inclusion Controller 1 updates the SIS with the latest network topology due to the included Inclusion Controller 2.

9 REFERENCES

- [1] Zensys, 900206102, Z-Wave ZW0102/ZW0201 Application Programming Guide.