Errata Note

Known Test Observations SDK v6.81.06

Document No.:	ERN13927
Version:	11
Description:	-
Written By:	JFR;COLSEN;PSH;JSI;SSE;EFH;NTJ;BBR
Date:	2019-07-19
Reviewed By:	JFR;BBR;CRASMUSSEN;LTHOMSEN;NTJ;PSH;JKA;HAKRONER;ABXAVIER;JOPEDERSEN;SSE
Restrictions:	None

Approved by:

Date	CET	Initials	Name	Justification
2019-07-19	03:32:51	JFR	Jorgen Franck	on behalf of NTJ

REVISION RECORD

Doc. Rev	Date	By	Pages affected	Brief description of changes
1	20170720	JFR	All	Initial draft based on ERN13182 – Known Test Observations SDK v6.71.01
1	20170725	JFR	All	Updated to SDK v6.80.00
1	20170727	PSH	5	Added description of ZSP-208 bug
2	20170927	JFR	1 & 2	Updated to SDK v6.81.00
3	20180111	JFR	4 & 5	Updated wrt. known bugs
4	20180227	JFR	1 & 2 4 & 5 Appendix F	Updated to SDK v6.81.01 Updated wrt. bugs Bugs fixed in SDK v6.81.00 moved to appendix
5	20180326	BBR	All	Added Silicon Labs template
6	20180531	JFR	1 & 2 4 & 5 Appendix E	Updated to SDK v6.81.02 Updated wrt. bugs Bugs fixed in SDK v6.81.01 moved to appendix
7	20180831	JFR	4 & 5	Updated wrt. bugs (ZSP-328 & ZF-345)
8	20181008	JFR	1 & 2 4 & 5 Appendix D	Updated to SDK v6.81.03 Updated wrt. bugs (Fixed ZSP-328, ZSP-329 & ZF-491) Bugs fixed in SDK v6.81.02 moved to appendix
9	20190123	JFR	1 & 2 4 5 Appendix C	Updated to SDK v6.81.04 Updated wrt. bugs fixed: ZSP-345, ZF-504, ZF-464, ZF-462, ZF-459 & ZF-451 Added known bugs: ZF-469, ZF-508, ZF-523, ZF-524, ZF-525, ZF-526, ZF-527, ZF-528, ZF-529 & ZF-530. Bugs fixed in SDK v6.81.03 moved to appendix
10	20190412	JFR	1 & 2 4 & 5 6.3 Appendix B	Updated to SDK v6.81.05 Updated wrt. bugs Added Z-Wave NVM Converter Bugs fixed in SDK v6.81.04 moved to appendix
10	20190717	JFR	1 & 2 4 Appendix A	Updated to SDK v6.81.06 Updated wrt. fixed bug SWPROT-2863 Bugs fixed in SDK v6.81.05 moved to appendix

Table of Contents

1 INTRODUCTION

The document describes the bugs that exist on the Z-Wave 500 Series Software Developer's Kit (SDK) v6.81.06, which is a mature release enabling support of Smart Start, etc. This release is intended for Z-Wave certified 500 Series based products entering volume production. All the Z-Wave Plus Applications are Z-Wave certified. Chapter 2 lists all programs and version numbers included on this SDK. For details regarding the SDK features and functionality, refer to [1].

1.1 Abbreviations

Abbreviation	Explanation
ACK	Acknowledge
API	Application Programming Interface
C	Command
CC	Command Class
DUT	Device Under Test
FLiRS	Frequently Listening Routing Slave. Communication to a FLiRS node can be established by a wakeup beam
ID	Identifier
NIF	Node Information Frame
NWI	Network Wide Inclusion (add node out of direct range)
NWE	Network Wide Exclusion (remove node out of direct range)
OTA	Over The Air (e.g. making a firmware update wireless)
OTW	Over The Wire (e.g. making a firmware update via the serial API interface)
S0	Security 0 Command Class
S2	Security 2 Command Class
SDK	Software Development Kit
TO	Test Observation (bug)

2 RELEASED VERSIONS

2.1 SDK 6.81.06

```
Z-Wave Framework and Certified Applications.............. v4_03_00
Z-Wave Protocol and Serial API Applications.............. v6_07_00
Z-Wave Serial API Application Interface..................... v8

Tools
=====
IMA Tool..................................................... v0_99
NVM Converter................................................ v0_07
uVision Project Generator.................................... v1_16
```

3 KEIL PK51

Be aware that Z-Wave SDK 6.81.00+ requires Keil PK51 v9.54A due to bugs in previously used versions. Here is a short description of the bugs triggering the version shift:

1. A bug in the linker LX51 (PK51 v9.51a) can result in colliding segments. Linker returns WARNING L30: MEMORY SPACE OVERLAP when bug occurs.
2. A bug introduced in the linker LX51 (PK51 v9.52) result in wrong interpretation of directives for a code bank. For details refer to http://www.keil.com/support/docs/3647.htm
3. The OHX51 program in the Keil PK51 v9.53 tool chain can generate redundant hex data in output in certain situations. The srec_cat program detects that redundant code segments are present in the case where COMMON faces code space shortage. This generates a FATAL ERROR telling the user that COMMON block overflow occurred. The problem is solved by moving modules from COMMON to BANK 1 or 3.
4. Do not use Keil PK51 v9.56 due to reintroduction of a former bug. The code bank end markers (?CO?ZW_FIRMWARE_BANK3_MARK,…) are located in wrong locations, and the tool chain sometimes does not give any warning or error message despite the fact that the newly built firmware does not work.

4 Z-WAVE PROTOCOL

Following sections describe fixed and known bugs in the Z-Wave protocol libraries.

4.1 Fixed Bugs

Bug#/Summary:	SWPROT-2863 – Filtering of incoming frames
Library:	All
ASIC:	500 series
Detailed Description:	Incoming frames with a correct checksum may contain erroneous information that initiates an operation in the device in question.
Consequence:	Increased latency handling frames containing erroneous information.
Workaround:	None.

4.2 Known Bugs

Bug#/Summary:	ZSP-59 – After requesting Node Neighbor Update to a node, the controller clears LWR to that node.
Library:	All controllers
ASIC:	500 series
Consequence:	Controller must calculate a route based on the topology map; alternatively issue an explore frame.
Workaround:	None.

Bug#/Summary:	ZSP-72 – KEX_FAIL_DECRYPT status is not always used.
Library:	All slaves
ASIC:	500 Series
Detailed Description:	When sending KEX Report echo encrypted with a wrong network key, S2 does not respond with KEX Fail with Type set to KEX_FAIL_DECRYPT = 05 and encrypted with the temporary key [009F.01.00.11.077], instead it continues sending KEX_SET well after the inclusion has concluded.
Consequence:	None.
Workaround:	None.

Bug#/Summary:	ZSP-80 – The API call ZW_ReplaceFailedNode does not update SIS, when executed on an inclusion controller.
Library:	All controllers
ASIC:	500 series
Consequence:	The inclusion controller does not update SIS in case NIF have changed.
Workaround:	None.

Bug#/Summary:	ZSP-81 – During inclusion mode a controller ignores unsolicited data, but answers explorer with Explore Search Result.
Library:	All controllers
ASIC:	500 series
Detailed Description:	As the frame is not answered, the transmitting node will retry and ultimately send an explorer frame, which the controller does answer with an Explorer Search Result. This will trigger the process to start again.
Consequence:	Traffic overhead during inclusion
Workaround:	Do not communicate with a controller before it is included.

Bug#/Summary:	ZSP-83 – Excluding original primary controller from network may create multiple networks with identical home ID.
Library:	All controllers
ASIC:	500 series
Detailed Description:	When a primary controller promotes a secondary controller to primary, and the secondary controller then excludes the original primary, which is now secondary from the network, the excluded controller keeps its home ID. This results in two controllers, the promoted one and the excluded one, having the same home ID.
Consequence:	Multiple controllers with same home ID may be created.
Workaround:	Do a ZW_SetDefault() after being excluded from a network.

Bug#/Summary:	ZSP-87 – A controller calling RequestNodeNeighborUpdate may not get an updated 'Node Range Info' in the case where the requested controller already handles a similar request from another node.
Library:	Controllers
ASIC:	500 series
Detailed Description:	The returned 'Node Range Info' contains empty mask bytes indicating requested controller has no neighbors.
Consequence:	The 'Node Range Info' containing empty mask bytes will be discarded by the receiving controller as faulty. The receiving controller will not be updated.
Workaround:	None.

Bug#/Summary:	ZSP-167 – Returns wrong failed node remove status
Library:	All
ASIC:	500 Series
Detailed Description:	The controller tries to remove a node on its list of failing nodes. However, when failed node remove is issued, the node responds to the NOP frame (checking if it is alive). The expected status should then be FAILED_NODE_REMOVE_FAIL = 0x02 instead of FAILED_NODE_NOT_FOUND = 0x00 because the node is no longer failing.
Consequence:	No impact on application
Workaround:	Use FAILED_NODE_NOT_FOUND

Bug#/Summary:	ZSP-366 – Toolchain does not build sample apps identical to included hex files.
Library:	All
ASIC:	500 Series
Detailed Description:	Version Z-Wave Software Report Command also contains build numbers that are not added to the source code afterwards. Instead the default build numbers are used from the mk.bat file.
Consequence:	Results in different hex files because different build numbers are used.
Workaround:	Modify the following line in mk.bat file situated in the app's directories set BUILD_NUMBER=52445 to set BUILD_NUMBER=168

5 Z-WAVE FRAMEWORK AND EMBEDDED APPLICATIONS

The following sections describe fixed and known bugs in the Z-Wave Framework and embedded applications.

5.1 Framework (Application Utilities & Command Handlers)

5.1.1 Fixed Bugs

None.

5.1.2 Known Bugs

- ZF-345 – Sample app does not answer unsecure Manufacturer Specific Get Command, when granted S0 key. Workaround: Make the following change in the ZW_cmd_class_list.c by adding the following code to function

```
CmdClassSupported
 if ((SECURITY_KEY_NONE != device_higest_secure_level) &&   /* Securely included */
     (SECURITY_KEY_NONE == eKey) &&                         /* non-secure input */
     (SECURITY_KEY_S0 == device_higest_secure_level) &&     /* Security S0 */
     (COMMAND_CLASS_MANUFACTURER_SPECIFIC == cmdClass) &&
     NON_NULL( pSecurelist ))                               /* Securely included */
 {
   return TRUE; /*cmd is supported!*/
 }
```

  in front of

```
 if ((SECURITY_KEY_NONE == device_higest_secure_level) &&
     (cmdClass == COMMAND_CLASS_BASIC))
 {
   /* Non-secure node always support CC Basic. */
   return TRUE;
 }
```

- ZF-452 – Multi Channel devices clear all association groups on all end points and root device when receiving an Association Remove on a single end point. Notice that some devices have been certified with this behavior.
- ZF-508 - Application framework implements support for Firmware Update Meta Data CC, version 4. A controller supporting version 1 and version 2 must be able to firmware update a version 4 node, under the condition that the same checksum calculation is used. However, version 1 and 2 do not support fragment size in Firmware Update Meta Data Request Get and version 4 forgets to initialize it.
- ZF-553 – OTA firmware update does not work in case one is only updating an application patch version. It's required to update application, major and/or minor version, to be able to make an OTA firmware update.

5.2 Certified Applications

This section describes fixed and known bugs in the certified applications: Door Lock Key Pad, Power Strip, Sensor PIR, Switch On/Off and Wall Controller. This section also includes My Product Plus, Production Test Generator and Production Test DUT.

5.2.1 Fixed Bugs

None.

5.2.2 Known Bugs

- ZF-381 – Sensor PIR does not use Supervision encapsulation on Notification Reports. Using Supervision encapsulation provides an “application layer ack” and allows the device to go to sleep as soon as the supervision report is received. Workaround: Make the following change in `CommandClassNotification.c`:

```
if (ZW_TX_IN_PROGRESS != ZW_TransportMulticast_SendRequest(
      (BYTE *)pTxBuf,
      (sizeof(ZW_NOTIFICATION_REPORT_1BYTE_V4_FRAME) - sizeof(BYTE) +
        pTxBuf->ZW_NotificationReport1byteV4Frame.properties1) - sizeof(BYTE),
      FALSE, // No Supervision
      pTxOptionsEx,
      ZCB_RequestJobStatus))
```

to

```
if (ZW_TX_IN_PROGRESS != ZW_TransportMulticast_SendRequest(
      (BYTE *)pTxBuf,
      (sizeof(ZW_NOTIFICATION_REPORT_1BYTE_V4_FRAME) - sizeof(BYTE) +
        pTxBuf->ZW_NotificationReport1byteV4Frame.properties1) - sizeof(BYTE),
      TRUE, // Use Supervision encapsulation
      pTxOptionsEx,
      ZCB_RequestJobStatus))
```

5.3 Serial API Plus Applications

This section describes fixed and known bugs in the Serial API applications: Controller Bridge, Controller Portable, Controller Static, Controller Static Single, Slave Enhanced 232 and Slave Routing.

5.3.1 Fixed Bugs

None.

5.3.2 Known Bugs

None.

6 TOOLS

The following sections describe fixed and known bugs in the development tools.

The SDK contains various tools for helping SW developers writing and debugging code.

NOTICE: Some of the tools such as the Z-Wave Zniffer, PC Programmer, PC Controller, etc. are not bundled together with the SDK anymore but are available on www.silabs.com as individual programs.

6.1 IMA Tool Box

6.1.1 Known Bugs

- TO #05391 – Can show more than 100% neighbor connectivity after a Network Health Measurement.
- TO #06691 – IMA Tool crashes when using a static controller without RSSI functionality.

6.2 uVision4 Project Generator

6.2.1 Fixed Bugs

- UPGOR186-27 – Some uVision projects could not build and some were not equal to the hex file command line made generated ones.

6.2.2 Known Bugs

None

6.3 Z-Wave NVM Converter

6.3.1 Known Bugs

None

APPENDIX A FIXED BUGS IN 6.81.05

Appendix A.1 Z-Wave Protocol

None.

Appendix A.2 Framework

- ZF-490 – Z-Wave Application Framework and Certified Application versions are not aligned. Certified Application versions are now set to the correct values in app_version.h file situated in the app's directories.

Appendix A.3 Certified Applications

- ZF-469 - Sensor PIR sends Wakeup Notification after a requested Version Report. When PC Controller sends Wakeup No More Information Sensor PIR is already sleeping.
- ZF-523 – Door Lock Key Pad doesn't respond to all the non-secure command classes encrypted with any network key. This is a negative test checking if a failure condition is handled correctly.
- ZF-524 – Sensor PIR receives a Supervision encapsulated Wake Up Interval with an unallowed Wake Up Interval (out of min/max range). The returned Supervision Report announces "success" instead of "fail". The same applies if a Wake Up Interval with an intermediate step is received. This is a negative test checking if a failure condition is handled correctly.
- ZF-525 – Wall Controller. End Points are not allowed to send the Notifications. If an Endpoint association is configured in the Lifeline, then the End Points are sending Central Scene Notification from their End Point IDs. However, the Central Scene CC is ONLY advertised on the root device but NOT on the End Points as supported.
- ZF-526 – Door Lock Key Pad. A node included via SmartStart inclusion MUST NOT auto-reset because it is granted fewer keys or no keys at all by the controller during S2 bootstrapping. This issue doesn't occur on the Switch On/Off, Sensor PIR, and Power Strip sample application. This is a negative test checking if a failure condition is handled correctly.
- ZF-527 - Door Lock Key Pad. Door Lock Command Class, Version 2. The Door Lock Configuration Report returns 0xFF for the Door Handles Mode (0x0F for inside, 0x0F for outside). However, the Inside Door Handle Mode is not implemented. With the Door Lock Configuration Set command values for the Inside Door Handles Mode can be set.

- ZF-528 - Door Lock Key Pad. Association Command Class, Version 2. The Door Lock Operation Report is neither sent to the IDs in the Lifeline nor to the IDs in Association Group 2. The report is to be triggered upon door lock operations.
- ZF-529 - Door Lock Key Pad. Firmware Update Meta Data Command Class, Version 4. When a firmware update is initiated after an aborted firmware update. A Get command for 1st firmware fragment of target 0x00 is expected. After that there is no Status Report frame. When a firmware update is initiated after an invalid checksum and fragment. After sending a firmware fragment with a valid fragment number and invalid checksum, a Get command for 1st firmware fragment (Report Number = 1) again is expected but the frame is missing.
- ZF-530 - Door Lock Key Pad. A Door Lock Operation Report is not triggered upon a change in door lock operation.

Appendix A.4 SerialAPI Plus Applications

None.

APPENDIX B FIXED BUGS IN 6.81.04

Appendix B.1 Z-Wave Protocol

None.

Appendix B.2 Framework

- ZF-451 – Firmware Update MD CC blocking app from operating normally while in progress.
  Workaround: Remove the following line from `ZCB_OTAStart()`:
  `ZCB_eventSchedulerEventAdd((EVENT_APP) EVENT_SYSTEM_OTA_START);`
  This applies to all applications.
- ZF-459 – Response buffer not freed in Association Group List Get Command in case size is zero.
  Workaround: Make the following change in `CommandClassAssociationGroupInfo.c`:

```
case ASSOCIATION_GROUP_COMMAND_LIST_GET:
      pTxBuf = GetResponseBuffer();
      if (IS_NULL(pTxBuf))
      {
        // The buffer is not free :(
        return RECEIVED_FRAME_STATUS_FAIL;
      }

      groupID = pCmd->ZW_AssociationGroupCommandListGetFrame.groupingIdentifier;

      ZAF_CC_AGI_CorrectGroupIdIfInvalid(rxOpt->destNode.endpoint, &groupID);

      length = GetApplGroupCommandListSize(groupID, rxOpt->destNode.endpoint);
      if (0 == length)
      {
        return RECEIVED_FRAME_STATUS_FAIL;
      }
```

  to

```
case ASSOCIATION_GROUP_COMMAND_LIST_GET:
      pTxBuf = GetResponseBuffer();
      if (IS_NULL(pTxBuf))
      {
        // The buffer is not free :(
        return RECEIVED_FRAME_STATUS_FAIL;
      }

      groupID = pCmd->ZW_AssociationGroupCommandListGetFrame.groupingIdentifier;

      ZAF_CC_AGI_CorrectGroupIdIfInvalid(rxOpt->destNode.endpoint, &groupID);

      length = GetApplGroupCommandListSize(groupID, rxOpt->destNode.endpoint);
      if (0 == length)
      {
        FreeResponseBuffer()
        return RECEIVED_FRAME_STATUS_FAIL;
      }
```

- ZF-462 – Compile bug when expanding Switch On/Off to support multiple endpoints NUMBER_OF_ENDPOINTS_NVM_MAX.

- ZF-464 – Send/receive ID mismatch for supervision session ID. Workaround: Make the following change in `CommandClassSupervision.c`:
  `if ((m_sessionId - 1) == pCmd->ZW_SupervisionReportFrame.properties1)`
  to
  `if ((m_sessionId) == pCmd->ZW_SupervisionReportFrame.properties1)`
- ZF-504 – COMMON BANK overflow in Switch On/Off app with Debug enabled and including Triac Controller API with one or more interrupts enabled.

Appendix B.3 Certified Applications

None.

Appendix B.4 SerialAPI Plus Applications

None.

APPENDIX C FIXED BUGS IN 6.81.03

Appendix C.1 Z-Wave Protocol

Bug#/Summary: ZSP-328 – Multicast FLiRS node leaves unintentionally broadcast group

Library: All

ASIC: 500 Series

Detailed Description: The FLiRS node saves a broadcast bit after the first FLIRS broadcast, multicast, and finally singlecast (set broadcast bit) follow up. This bit ensures that all FLiRS nodes attached to a group wakes up when issuing a FLiRS broadcast. Afterwards the multicast addresses the specific group user wants to operate. The problem is that the broadcast bit can apparently be cleared for some reason causing the popcorn effect when operating these devices.

Consequence: Popcorn effect

Workaround: None.

Bug#/Summary: ZSP-329 – otacompress.exe fails on 32-bit computers

Library: All

ASIC: 500 Series

Detailed Description: Distributed otacompress.exe supports only 64-bit computers.

Consequence: Getting an otacompress.exe error 216 when compiling the project on a 32-bit computer.

Workaround: Copy the otacompress.exe file from …\SDK_v6_81_00\Z-Wave\lib\otacompress directory to the …\SDK_v6_81_02\Z-Wave\lib\otacompress.

Appendix C.2 Certified Applications

None.

Appendix C.3 Framework

None.

Appendix C.4 Serial API Plus Applications

- ZF-491 – ZW_NetworkManagementSetMaxInclusionRequestIntervals Serial API call is missing. Implementation can be provided on request.

APPENDIX DFIXED BUGS IN 6.81.02

Bug#/Summary: ZSP-321 – Returns wrong TxStatus when sending a S2 message to a non-existing node.

Library: All slaves

ASIC: 500 series

Consequence: API calls ZW_SendData, ZW_SendDataMulti, and ZW_SendDataBridge returns TRANSMIT_COMPLETE_OK when a secure transmit fails and should return TRANSMIT_COMPLETE_NO_ACK.

Workaround: None.

APPENDIX E FIXED BUGS IN 6.81.01

Bug#/Summary:	ZSP-195 – Protocol does not conform with S2 inclusion timeouts.
Library:	All
ASIC:	500 Series
Detailed Description:	If waiting for more than 60 seconds before granting keys, the slave node will not answer the "kex set" with a "public key report" and the S2 inclusion will fail. CC:009F.01.00.11.087 specifies that TB2 must be 240 seconds.
Consequence:	None

Bug#/Summary:	ZSP-209 – RSSI in callback is incorrect.
Library:	All controllers
ASIC:	500 Series
Detailed Description:	In networks with heavy traffic or noise the RSSI values in the ZW_SendData() callback can be marked as not available for routed frames even though the RSSI values are gathered in the routed ack frame.
Consequence:	RSSI values are not always available in the ZW_SendData() callback for routed frames.

Bug#/Summary:	ZSP-236 - FLIRS multicast beams node ID 29 instead of the broadcast destination, 255.
Library:	All 3 channel libraries
ASIC:	500 Series
Detailed Description:	When sending Multicast frames to 3 channel FLiRS nodes, the multicast wakeup beam would be addressed to node ID 29 and the FLiRS nodes would not wake up until they received the singlecast follow-up. The problem only occurred when sending unsecure; when sending S2 multicast, the wakeup beam is correct.
Consequence:	Multicast beaming would not work, the FLiRS nodes would not wake up on the multicast but first on the singlecast follow-up frame.

Bug#/Summary:	ZSP-241 – ACK channel was wrong in callback.
Library:	All
ASIC:	500 Series
Detailed Description:	The SendData (and SendDataBridge) callback includes various data, including the ACK channel, rssi of each hop, and more. On 2CH networks, the ACK Channel is always reported as zero even if the frame is send on another channel.
Consequence:	ACK channel is wrong in callback for 2 channel systems.

Bug#/Summary:	ZSP-243 – S2 nodes do not reset if power cycled during 10 seconds key calculation.
Library:	Slaves
ASIC:	500 Series
Detailed Description:	If a Smart Start node is powercycled during the public key calculation, it will fail S2 inclusion and remain non-securely included. It will not reset, since it has forgotten about the ongoing SS inclusion.
Consequence:	S2 nodes might end up as non-secure included nodes if power cycled during inclusion

Bug#/Summary:	ZSP-267 - Routing slave ignores assigned return routes, when a SIS return route has been assigned to the same destination.
Library:	All slaves
ASIC:	500 Series
Detailed Description:	When a slave node is sending a frame to a SUC/SIS, it will always use the routes assigned with ZW_AssignSUCReturnRoute() even if routes to the same (SUC/SIS) destination is also assigned with ZW_AssignReturnRoute().
Consequence:	The behavior of the assign return routes calls is not obvious, and it will probably lead to wrong implementations.

Bug#/Summary:	ZSP-275 – ZW_SendConst ignores power settings in 3-channel regions.
Library:	All
ASIC:	500 Series
Detailed Description:	ZW_SendConst ignores power settings for channel 1 and 2 in 3-channel regions.
Consequence:	The power setting for channel 0 is always used for all channels.

Bug#/Summary:	ZSP-278 – Slave nodes could go to sleep during neighbor discovery.
Library:	All slaves
ASIC:	500 Series
Detailed Description:	Protocol goes to sleep immediately if ZW_SetSleepMode is called during Neighbor discovery. This could prevent rediscovery of neighbors from battery-powered nodes.
Consequence:	Battery powered nodes could, in some cases, not rediscover neighbors.

Bug#/Summary:	ZSP-279 – S2 slave node is unable to resynchronize SPAN when sending multicast.
Library:	All slaves
ASIC:	500 Series
Detailed Description:	Transport service was not used correctly in multicast. So when a S2 multicast frame exceeded max frame size, it was not split by transport service but discarded.
Consequence:	S2 multicast of frames with large payload would not work.

Bug#/Summary:	ZSP-280 – Changing power level setting in App_RFSetup.c has no effect.
Library:	All
ASIC:	500 Series
Detailed Description:	New power-level settings in App_RFSetup.c are not included when building application.
Consequence:	Always uses default power level settings.

Bug#/Summary:	ZSP-284 – 3 channel IMA returns ZW_RF_SPEED_40K when frame is not ACK'ed.
Library:	All
ASIC:	500 Series
Detailed Description:	3 channel always uses 100 kbps independent of channel used.
Consequence:	Returns 2 channel related parameter ZW_RF_SPEED_40K instead of ZW_RF_SPEED_100K.

APPENDIX F FIXED BUGS IN 6.81.00

Bug#/Summary:	ZSP-27 – Bridge reports are not a supported command class in new node registered.
Library:	Controller bridge
ASIC:	500 Series
Detailed Description:	When a bridge reports a new virtual node to a SIS controller, the new node registered frame always has command classes with the value 0 after the virtual nodes valid command classes.
Consequence:	None.

Bug#/Summary:	ZSP-71 – Network Inclusion through repeater seems to always fail the initial Z-Wave AssignID frame communication. However, the node still get included.
Library:	All
ASIC:	500 Series
Detailed Description:	Z-Wave AssignID frame through repeater/s fail consistently - but the following NOP transmissions will normally determine the node as included.
Consequence:	Inclusion latency.

REFERENCES

[1] Silicon Labs, SRN13926, Software Release Note, Z-Wave 500 Series SDK v6.81.00 Beta.
[2] Silicon Labs, INS13933, Instruction, Z-Wave 500 Series SDK Contents v6.8x.0x.
[3] Silicon Labs, INS12366, Instruction, Working in 500 Series Environment User Guide.
[4] Silicon Labs, INS13954, Instruction, Z-Wave 500 Series Application Programming Guide v6.8x.0x.
[5] Silicon Labs, SDS10242, Software Design Specification, Z-Wave Device Class Specification.
[6] Silicon Labs, SDS11847, Software Design Specification, Z-Wave Plus Device Types Specification.
[7] Silicon Labs, SDS11846, Software Design Specification, Z-Wave Plus Role Types Specification.
[8] Silicon Labs, SDS10865, Software Design Specification, Z-Wave Security Application Layer.
[9] Silicon Labs, SDS13781, Software Design Specification, Z-Wave Application Command Class Specification.
[10] Silicon Labs, SDS13782, Software Design Specification, Z-Wave Management Command Class Specification.
[11] Silicon Labs, SDS13783, Software Design Specification, Z-Wave Transport-Encapsulation Command Class Specification.
[12] Silicon Labs, SDS13784, Software Design Specification, Z-Wave Network-Protocol Command Class Specification.
[13] Silicon Labs, SDS13548, Software Design Specification, List of defined Z-Wave Command Classes.

INDEX

No index entries found.