Application Note: JN-AN-1250

Memsize Tool

1 Application Note Overview

The purpose of this Application Note is to show how to use the memory profiling tool, memsize, and apply it to another Application Note.

The memsize tool is a simple memory profiling tool that creates an ExcelWorkbook (.xlsx) with three sheets:

• **Sections** – all sections, their addresses, sizes, and regions.

	A	B	C	D
1	**Section name**	**Address**	**Size (bytes)**	**Region**
2	.bir	0x00080000	16	flash
3	.flashheader	0x00080010	40	flash
4	.rodata	0x00080038	7660	flash
5	.text	0x00081E24	199468	flash
6	.data	0x0400004C	2324	ram
7	.bss	0x04000960	18772	ram
8	.heap	0x040052B4	2000	ram
9	.stack	0x04006C78	5000	ram

Example Sections page from JN-AN-1218 ExtendedColorLight

• **Components** – all components, including a catch-all entry for unresolved components, and their respective region sizes.

	A	B	C
1	**Component name**	**flash (bytes)**	**ram (bytes)**
2	Application	903	2340
3	Application (Common)	5338	2418
4	BDB	7683	790
5	Libraries	45804	1065
6	N/A	3722	267
7	NTAG	4705	287
8	PDUM	2894	4602
9	ZCIF	17583	1260
10	ZCL	30102	692
11	ZPS	77570	7100
12	ZigBee Common	1225	12
13	**Totals**	**197529**	**20833**

Example Components page from JN-AN-1218 ExtendedColorLight

• **Symbols** – all symbols, their sections, sizes, components, the objects which they came from, and their respective library (if applicable).

	A	B	C	D	E
1	Symbol name	Section	Size (bytes)	Component	Object
2	APP_bNtagPdmLoad	.text	631		../../ExtendedColorLight/Build/app_ntag_icode.o
3	APP_cbNtagEvent	.text	112		../../ExtendedColorLight/Build/app_ntag_icode.o
4	APP_cbNtagTimer	.text	2025		../../ExtendedColorLight/Build/app_ntag_icode.o
5	APP_vNtagStart	.text	476		../../ExtendedColorLight/Build/app_ntag_icode.o
6	OSMIUM_HwVectTable	.text	78		../../ExtendedColorLight/Build/portasm_JN516x.o
7	ZPS_eEnterCriticalSection	.text	54		../../ExtendedColorLight/Build/port_JN516x.o
8	ZPS_eExitCriticalSection	.text	28		../../ExtendedColorLight/Build/port_JN516x.o
9	ZPS_u8GrabMutexLock	.text	55		../../ExtendedColorLight/Build/port_JN516x.o
10	ZPS_u8ReleaseMutexLock	.text	33		../../ExtendedColorLight/Build/port_JN516x.o
11	bFactoryReset	.bss	1		../../ExtendedColorLight/Build/app_ntag_icode.o
12	bMutexTaken	.bss	1		
13	bNtagTimer	.bss	1		../../ExtendedColorLight/Build/app_ntag_icode.o
14	bReset	.bss	1		../../ExtendedColorLight/Build/app_ntag_icode.o
15	eAppNtagMode	.bss	1		../../ExtendedColorLight/Build/app_ntag_icode.o
16	eAppNtagState	.bss	1		../../ExtendedColorLight/Build/app_ntag_icode.o
17	sNfcNwkInstallCodePdm	.bss	18		../../ExtendedColorLight/Build/app_ntag_icode.o
18	sNfcNwkNciPdm	.bss	46		../../ExtendedColorLight/Build/app_ntag_icode.o
19	sNfcNwkPayloadRead	.bss	90		../../ExtendedColorLight/Build/app_ntag_icode.o
20	sNfcNwkPayloadWrite	.bss	90		../../ExtendedColorLight/Build/app_ntag_icode.o
21	u32AppNtagMs	.bss	4		../../ExtendedColorLight/Build/app_ntag_icode.o
22	u32AppNtagTicks	.bss	4		../../ExtendedColorLight/Build/app_ntag_icode.o
23	u32NfcNwkAddress	.bss	4		../../ExtendedColorLight/Build/app_ntag_icode.o
24	u32WipTimeout	.bss	4		
25	u8TimerNtag	.bss	1		../../ExtendedColorLight/Build/app_ntag_icode.o
26	vAlignmentErrorHandler	.text	12		../../ExtendedColorLight/Build/port_JN516x.o
27	vBusErrorHandler	.text	12		../../ExtendedColorLight/Build/port_JN516x.o
28	vGenericHandler	.text	12		../../ExtendedColorLight/Build/port_JN516x.o
29	vIllegalInstructionHandler	.text	12		../../ExtendedColorLight/Build/port_JN516x.o
30	vShowException	.text	128		../../ExtendedColorLight/Build/port_JN516x.o
31	vStackOverflowHandler	.text	12		../../ExtendedColorLight/Build/port_JN516x.o

Example Symbols page from JN-AN-1218 ExtendedColorLight

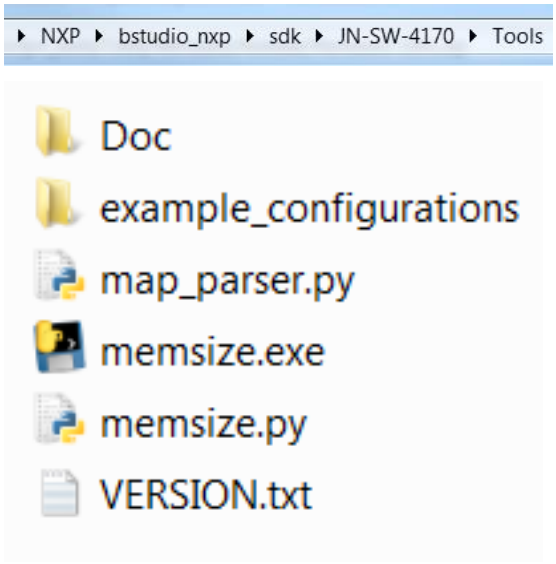
2 Capabilities

Product Type	Part Number	Build
JN516x ZLL/HA SDK	JN-SW-4168	1620
JN516x ZigBee 3.0 SDK	JN-SW-4170	1518
JN517x ZigBee 3.0 SDK	JN-SW-4270	1615
‘BeyondStudio for NXP’ Toolchain	JN-SW-4141	1308
LPCXpresso Toolchain	7.9.2	493

3 Integrating the tool

The memsize tool should be installed here:

(SDK)/Tools/Memsize2/



There are two changes that need to be made to the makefile in order to use the memsize tool. An example Makefile can be found in the folder 'example_configurations', for the JN-AN-1218 ExtendedColorLight. The alterations can be seen at lines 655 and 705. Please be aware that the examples used in this user guide are from the JN-AN-1218 ZigBee 3.0 Light Bulb v1003 app note.
First, a second linking should take place with LTO disabled (all one line);

$(CC) **-Wl,--cref** -Wl,--gc-sections -Wl,-u_AppColdStart -Wl,-u_AppWarmStart $(LDFLAGS) -T$(ZNCLKCMD) -L $(SDK_BASE_DIR)/Stack/ZCL/Build/ -o $@ -Wl,--start-group $(APPOBJS) $(addprefix -l,$(LDLIBS)) -lm -Wl,--end-group **-fno-lto** -Wl,-Map,$(DEV_BLD_DIR)/$(TARGET)_$(JENNIC_CHIP)$(BIN_SUFFIX)_**NOLTO**.map

This should take place immediately after the first linking. While the bold sections are specific to the second linking, non-bold sections from the line above may need to be adjusted to match the paths already used for the first linking, e.g. $(APP_BLD_DIR) rather than $(DEV_BLD_DIR).

Secondly, the memsize executable needs to be run (again, all one line);

$(SDK_BASE_DIR)/Tools/Memsize2/memsize.exe $(DEV_BLD_DIR)/$(TARGET)_$(JENNIC_CHIP)$(BIN_SUFFIX).elf $(DEV_BLD_DIR)/$(TARGET)_$(JENNIC_CHIP)$(BIN_SUFFIX)_NOLTO.map $(TARGET).json

Please note that memsize is a post build process and should therefore be invoked last.

A relevant config (.json) file should have the same name as the binary file's target prefix, and should be added to the appropriate location in the app note, most likely 'Common'. The name is defined by the last parameter of the above line; ($(TARGET).json).

For example, the configuration file for ExtendedColorLight can be found in "JN-AN-1218-ZigBee-3-0-Light-Bulb/Common_Light/Build/".

4 Configuration File

4.1 Description

The most complex part of the memsize tool is the sorting of symbols. They are split into user-defined logical categories called *Components*. Each component has two attributes: color (optional) and a list of patterns. The color is used for highlighting and the patterns are used to determine which symbol falls into the category. The way components are specified is via a config file, using the JSON format and it should conform to the following schema:

```
schema = {
        "type": "object",
        "additionalProperties": {
                "type": "object",
                "properties": {
                        "patterns": {
                                "type": "array",
                                "items": {"type": "string"}
                        },
                        "color": {"type": "string"}
                },
                "required": ["patterns"]
        }
}
```

To see a collection of example files, please refer to the folder 'example_configurations'.

4.2 Example: Extended Color Light

To begin, the application C files must be listed under ‘Application’. Therefore, the Extended Color Light application lists;

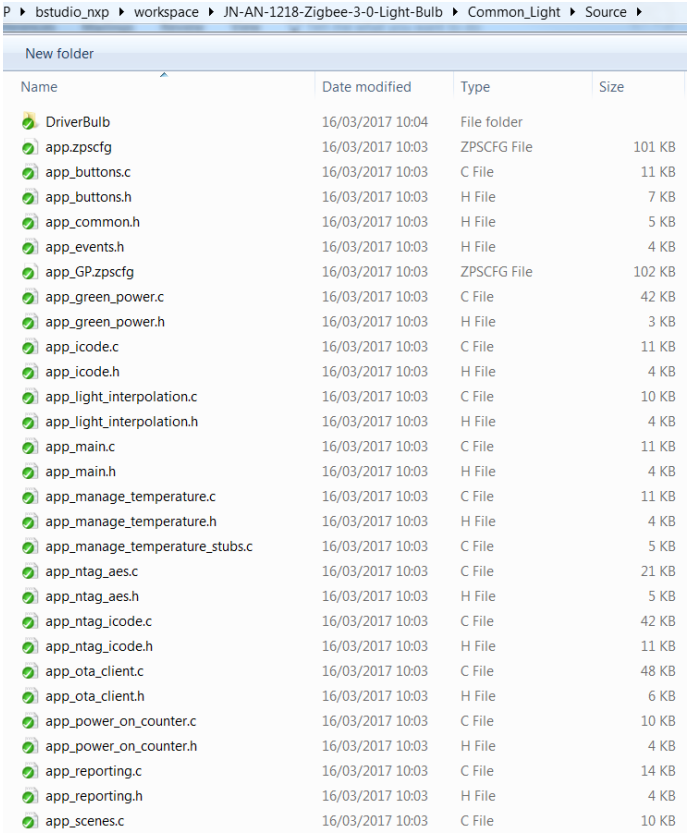
```
"Application": {
    "patterns": [
        "^.*App_ExtendedColorLight\\.o$"
    ],
    "color": "fffadbe0"
},
```

ExtendedColorLight.json

All C files in the common folder must also be listed under ‘Application (Common)’. In this example, the files are located in the ‘Common_Light’ folder;

```
"Application (Common)": {
    "patterns": [
        "^.*app_buttons\\.o$",
        "^.*app_green_power\\.o$",
        "^.*app_icode.c\\.o$",
        "^.*app_light_interpolation\\.o$",
        "^.*app_main\\.o$",
        "^.*app_manage_temperature\\.o$",
        "^.*app_manage_temperature_stubs\\.o$",
        "^.*app_ntag_aes.c\\.o$",
        "^.*app_ntag_icode.c\\.o$",
        "^.*app_ota_client\\.o$",
        "^.*app_power_on_counter\\.o$",
        "^.*app_reporting\\.o$",
        "^.*app_scenes\\.o$",
        "^.*app_start_light\\.o$",
        "^.*app_zcl_light_task\\.o$",
        "^.*app_zlo_light_node\\.o$",
        "^.*irq_JN516x\\.o$",
        "^.*DriverBulb_DR1173\\.o$",
        "^.*DriverBulb_DR1175\\.o$",
        "^.*DriverBulb_DR1190\\.o$",
        "^.*DriverBulb_DR1192\\.o$",
        "^.*DriverBulb_DR1221\\.o$",
        "^.*DriverBulb_DR1221_69\\.o$",
        "^.*DriverBulb_DR1221_Dimic\\.o$",
        "^.*DriverBulb_DR1223\\.o$",
        "^.*DriverBulb_JN516X\\.o$",
        "^.*DriverBulb_OM15008_Dimic\\.o$",
        "^.*DriverBulb_OM15045.o$",
        "^.*DriverBulb_OM15053.o$",
        "^.*DriverBulb_Shim\\.o$",
        "^.*DriverBulb_WS2812\\.o$",
        "^.*DriverBulb_WS2812_ba-elf\\.o$"
    ],
    "color": "fffadcd2"
```

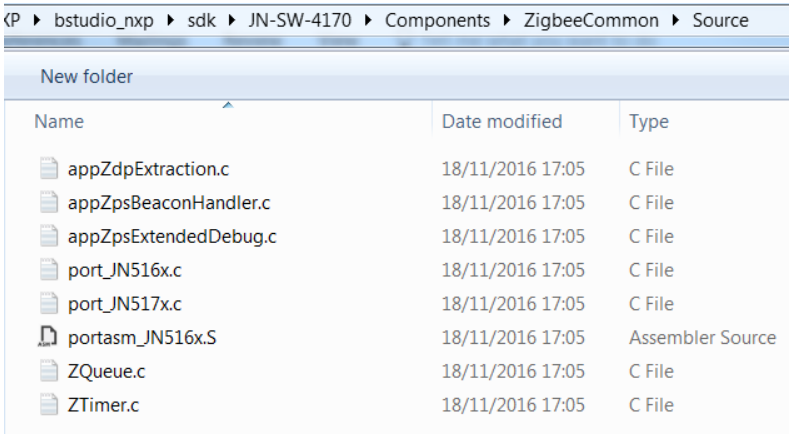
ExtendedColorLight.json



‘Common_Light’ Source folder

The NTAG files are also located in this directory, but are treated as their own object.

The rest of the objects (ZPS, PDUM, ZCL, ZCIF, BDB, ZigBee Common, ZigBee Utilities, Libraries) are found in the SDK, in a similar fashion to the Common files above;



'ZigBeeCommon' Source files, 4170 SDK v1518

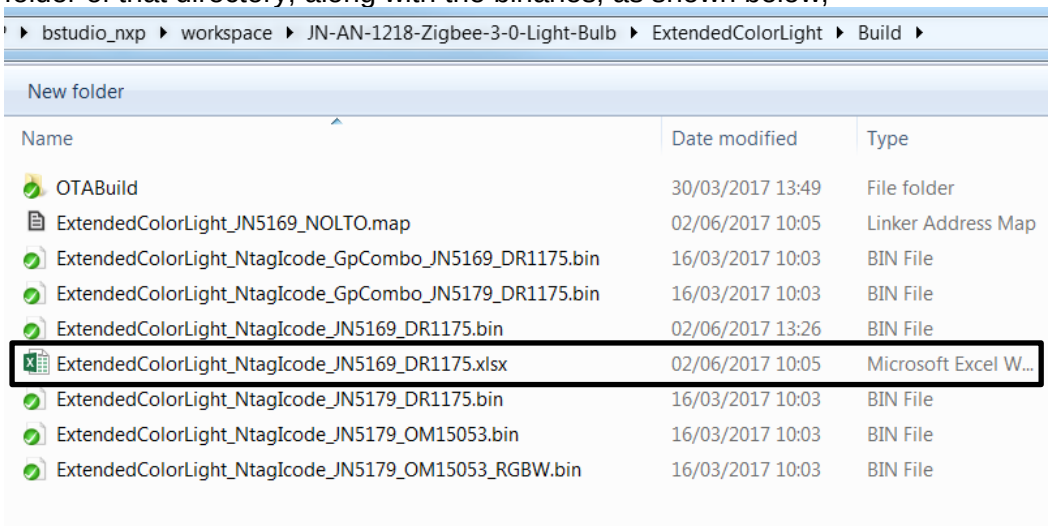
```
"ZigBee Common": {
    "patterns": [
        "^.*appZdpExtraction\\.o$",
        "^.*appZpsBeaconHandler\\.o$",
        "^.*appZpsExtendedDebug\\.o$",
        "^.*port\\.o$",
        "^.*portasm\\.o$",
        "^.*ZQueue\\.o$",
        "^.*ZTimer\\.o$"
    ],
    "color": "ffdee1f9"
},
```

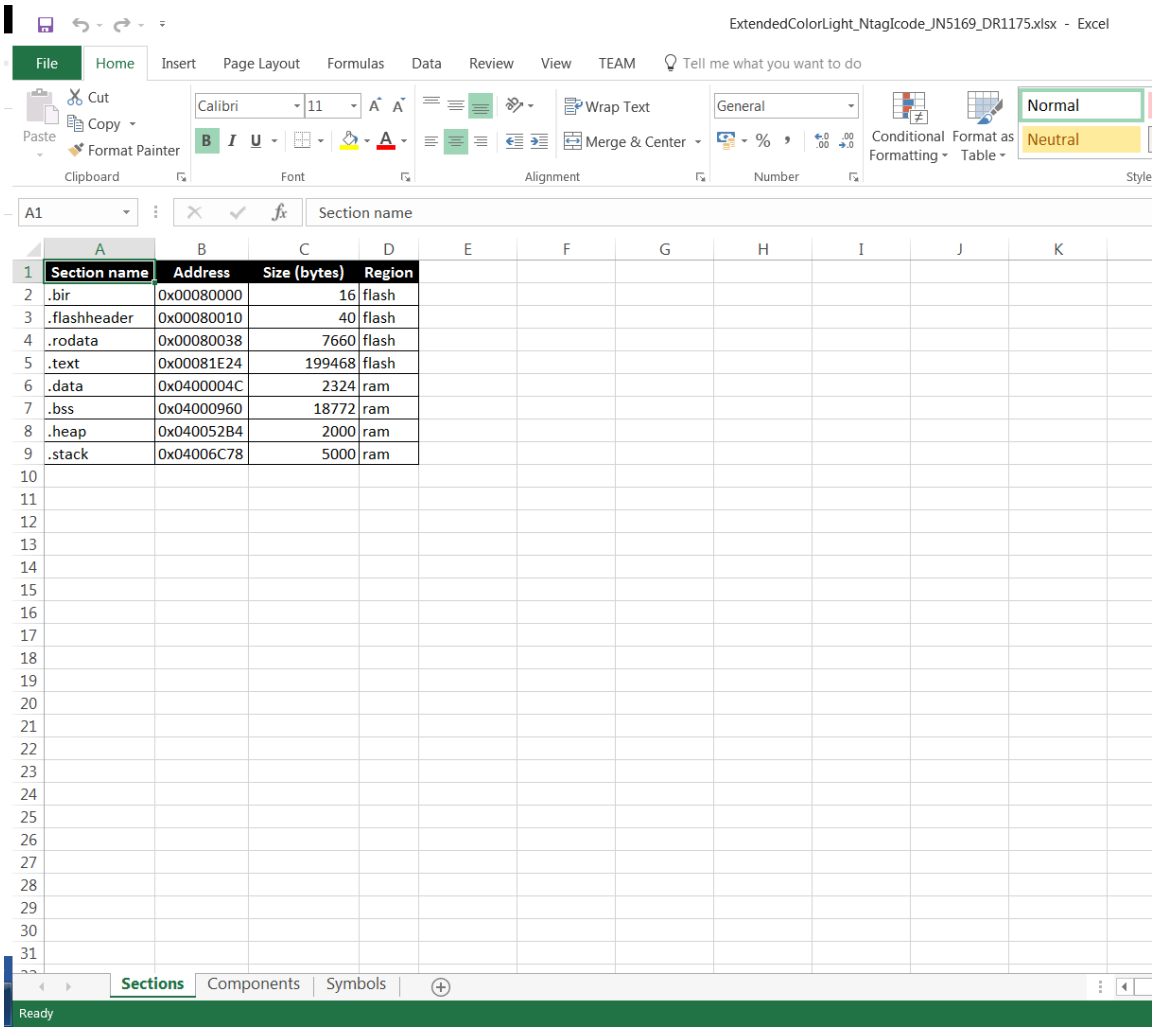
ExtendedColorLight.json

The required objects may differ between SDK and App Note application.

5 Running the demonstration

Before building, the .json configuration file must have been added to the 'Common' folder (or similar), and the makefile must be modified to include the second linking and Memsize tool invocation. Build the app note, and an ExcelWorkbook (.xlsx) will be created in the Build folder of that directory, along with the binaries, as shown below;





6 Creation of MemSize.exe

The process to create an executable file will be done with Pyinstaller. To know that the following process is applicable to all Python projects, only the section described as specific to MemSize application are mandatory for MemSize project.

6.1 Install Python 3.x

Ensure the latest Python 3.x version is installed in your computer:
https://www.python.org/downloads/

6.2 Install pip

1. Download and store get-pip.py from the following link: https://bootstrap.pypa.io/get-pip.py
2. Double-click on your downloaded file if you have configured your PC to open .py files with Python
   OR
   From a command prompt windows (e.g. cmd), go the directory containing the Python source code you just downloaded and tape the following command:
   `py -3 get-pip.py`
3. Go to Control Panel -> System -> Advanced system settings -> Environment Variables…
4. In the System variables, search for **Path** variable and edit it to add the path to pip.exe : C:\Users\<your_user_name>\AppData\Local\Programs\Python\Python36-32\Scripts
5. Click OK to exit all windows

6.3 Install PyWin32

Select the right installer in the following link for your system and install PyWin32:
https://github.com/mhammond/pywin32/releases

6.4 Install PyInstaller

From a command prompt windows (e.g. cmd), simply tape the following command:
`pip install pyinstaller`

6.5 Install jsonschema (specific to MemSize application)

From a command prompt windows (e.g. cmd), simply tape the following command:
`pip install jsonschema`

6.6 Install openpyxl (specific to MemSize application)

From a command prompt windows (e.g. cmd), simply tape the following command:
`pip install openpyxl`

6.7 Install jsonschema (specific to MemSize application)

From a command prompt windows (e.g. cmd), simply tape the following command:
```
pip install pyelftools
```

6.8 Process to get the executable file from the Python source files

From a command prompt windows (e.g. cmd), go the directory containing the Python source code for MemSize and tape the following command:
```
pyinstaller --onefile memsize.py
```

(`--onefile` is used to package everything into a single executable. If you do not specify this option, the libraries, etc. will be distributed as separate files alongside the main executable)

The executable file will be created inside **dist** folder.

7 Release Details

7.1 New Features

ID	Feature	Description
Version 1002		
N/A	N/A	• Update source code to handle .ARM sections in the MAP file • Update documentation to create the executable file of MemSize

7.2 Known Issues

ID	Severity	Description
Version 1002		
The tool lists that the data section is in RAM, but it is also in flash.		

7.3 Bug Fixes

ID	Description
Version 1002	
artf543194 [lpsw8737]	Memsize outputs map file warning

Revision History

Version	Notes
1000	First internal release
1001	Initial release
1002	• Update source code to handle .ARM sections in the MAP file • Update documentation to create the executable file of MemSize

Important Notice

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

All trademarks are the property of their respective owners.

NXP Semiconductors

For the contact details of your local NXP office or distributor, refer to:

www.nxp.com