EIQDVRTUG

eIQ DeepView RT User's Guide

Rev. 4 — 8 December 2022 **User guide**

Document information

Information	Content
Keywords	eIQ, DeepView, DeepView RT, Library
Abstract	This document describes the steps to download and start using the DeepView RT library and create an application for running pre-trained models.

1 Overview

DeepView RT is a proprietary neural network inference engine optimized for NXP microprocessors and microcontrollers, which implement its own compute engine and can leverage the third-party ones. For more information, see https://www.embeddedml.com/deepviewrt.

The MCUXpresso Software Development Kit (MCUXpresso SDK) provides a comprehensive software package with a pre-integrated DeepView RT library.

This document describes the steps to:

- Download and start using the library.
- Create an application for running pre-trained models.

Note: *It is assumed that you have a basic knowledge of machine learning frameworks for model training.*

2 Deployment

The eIQ DeepView RT sample is part of the eIQ machine learning software package, which is an optional middleware component of MCUXpresso SDK. The eIQ component is integrated into the MCUXpresso SDK Builder delivery system available on mcuxpresso.nxp.com.

To include eIQ machine learning into the MCUXpresso SDK package:

1. Open mcuxpresso.nxp.com.
2. Click Select Development Board to build and download a new package.
3. Log in with your email address and password.
4. Select the eIQ middleware component in the software component selector on the SDK Builder page.
5. Ensure that lwIP and FreeRTOS are selected in the MCUXpresso SDK Builder software component selector. For details, see Figure 1.

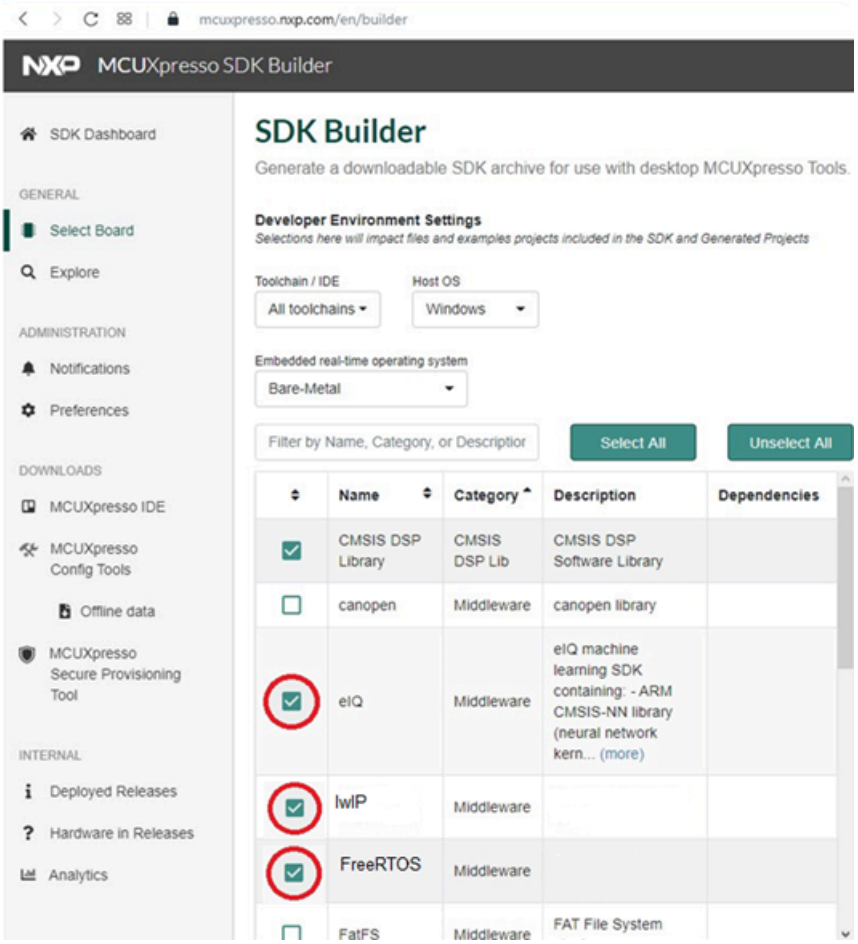

Figure 1. MCUXpresso SDK Builder software component selector

6. Once the MCUXpresso SDK package is downloaded, extract it in a folder on your local machine. Alternatively, import the package into the MCUXpresso IDE. For more information on the MCUXpresso SDK folder structure, see the Getting Started with MCUXpresso SDK User's Guide (document: MCUXSDKGSUG). Figure 2 shows the package directory structure. The eIQ deepviewRT sample directories are highlighted in red.

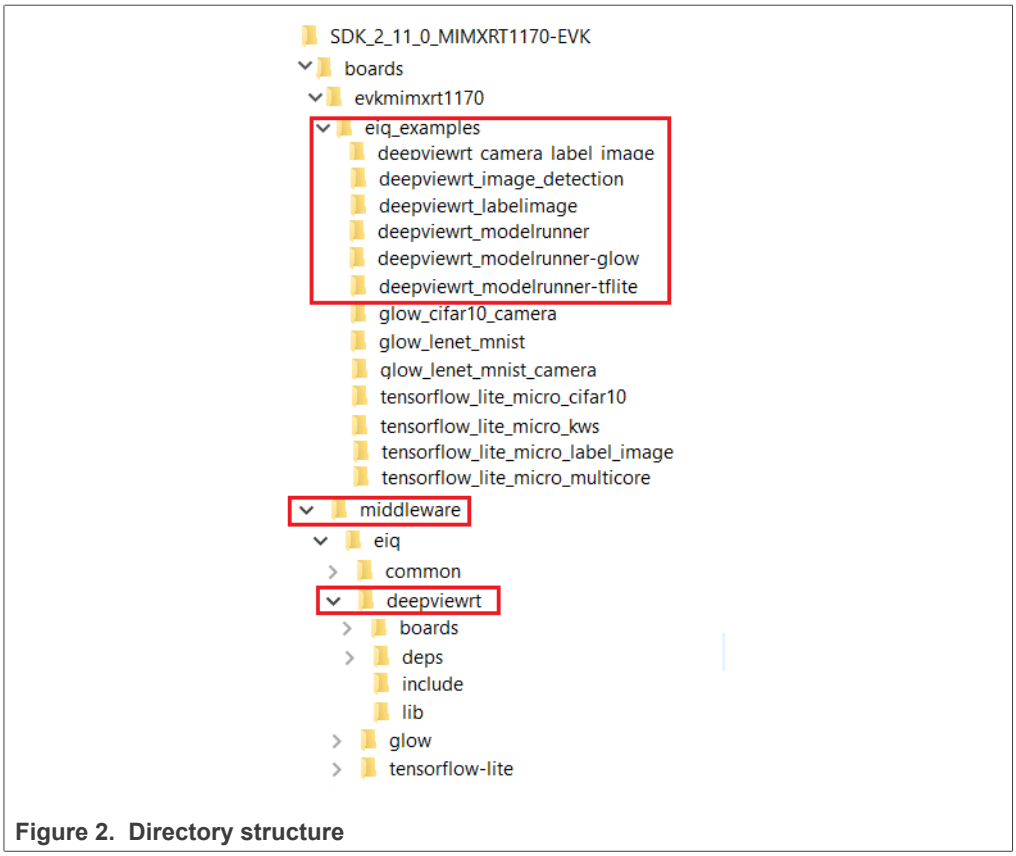

Figure 2. Directory structure

7. The *boards* directory contains example application projects for supported toolchains. For the list of supported toolchains, see the *MCUXpresso SDK Release Notes*. The *middleware* directory contains the eIQ library source code, pre-compiled library binaries, and example application source code and data.

3 Example applications

The eIQ DeepView RT includes a set of example applications. For details, see Table 1.

The applications demonstrate the usage of the library in several use cases and allow a rebuild of the library.

Table 1. List of example applications

Name	Description
`deepviewrt_modelrunner`	ModelRunner is a dedicated service for hosting and evaluating RTM graphs through a set of RPC protocols.
`deepviewrt_modelrunner-glow`	ModelRunner-glow is based on model runner and glow is integrated as plug-in.
`deepviewrt_modelrunner-tflite`	ModelRunner-tflite is based on modelrunner and support to run tflite models.
`deepviewrt_labelimage`	Labelimage demonstrates using DeepView RT C API to load and label an image file, returning the top-1 results along with their labels.

Table 1. List of example applications...*continued*

Name	Description
deepviewrt_image_detection	This application demonstrates using DeepView RT C API to load an image file and do objection detection, returning the objection bounding box along with their labels
deepviewrt_camera_label_image	This application demonstrates using DeepView RT C API to label image captured by camera, returning the top-1 results along with their labels on the LCD.

Note: *modelrunner and modelrunner-glow require network connection and depends on eIQ Toolkit. For details, see eIQ Toolkit document.*

For details on how to build and run the example applications with supported toolchains, see *Getting Started with MCUXpresso SDK User's Guide* (document: MCUXSDKGSUG). When using MCUXpresso IDE, the example applications is imported through the SDK Import Wizard as shown in Figure 3.

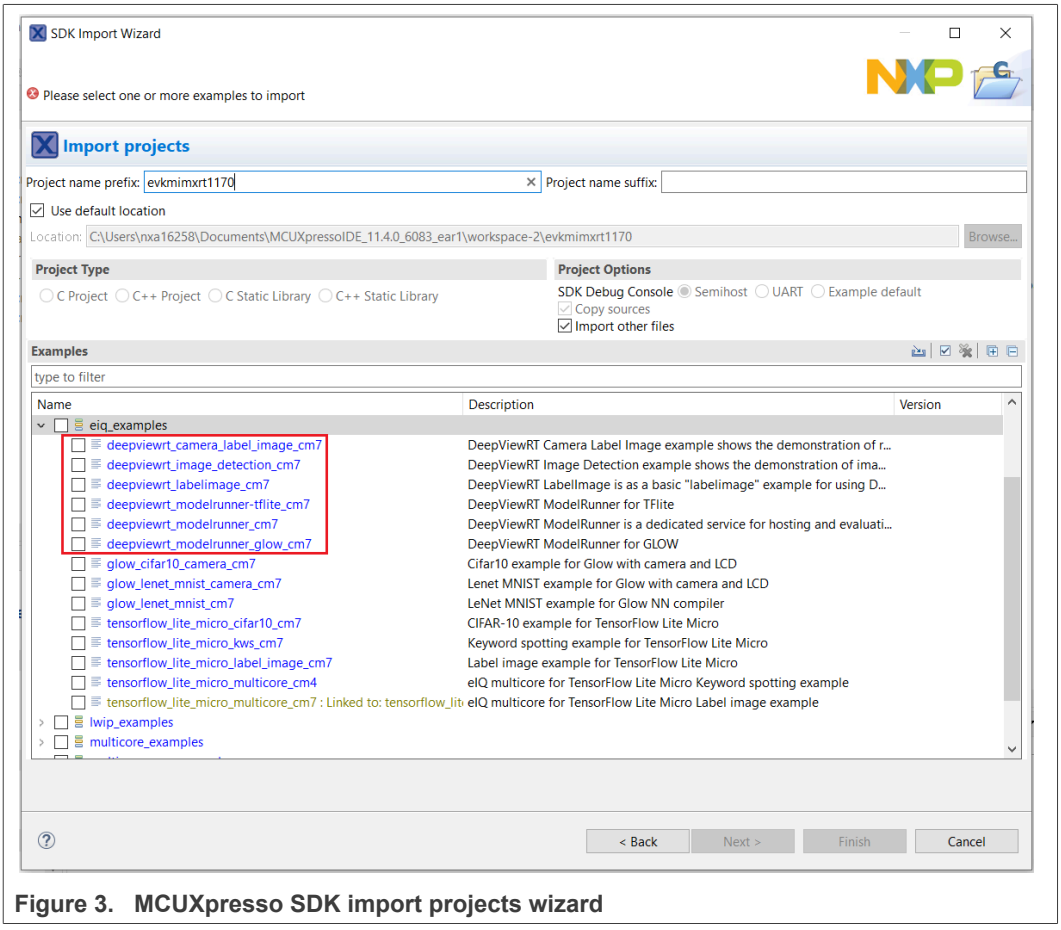

Figure 3. MCUXpresso SDK import projects wizard

After building the example application and downloading it to the target, the execution stops in the *main* function. When the execution resumes, an output message displays on the connected terminal. For example, Figure 4 shows the output of the labelimage example application printed to the MCUXpresso IDE Console window when semi hosting debug console is selected in the SDK Import Wizard.

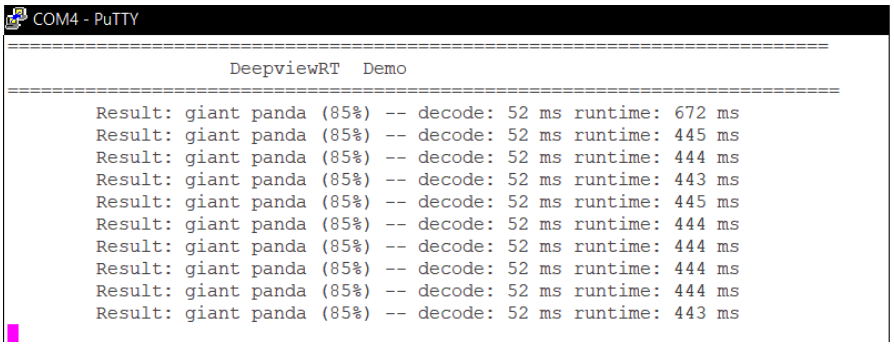

Figure 4. PuTTY console window

4 DeepView RT model

The DeepView RT Model (RTM) format supports in-place interpretation and is stored directly in flash and used as-is. DeepView RT Model enables resource constrained MCU, for example Arm Cortex-M, platforms as the actual model and the model's weights do not consume any RAM. Instead, the model's weight is kept in-place in flash memory. A small amount of memory is required for the network evaluation graph. Buffer cache is required for storing the volatile input/output data at inference time. However, no memory is required for the actual weights and can remain in flash. If the cache is large enough to host the weights, they are streamed from flash on-demand as a performance optimization. For maximum performance on parts with adequate memory, the entire model can be stored in RAM.

The DeepView RT Model (RTM) can be converted from Tensorflow, Tensorflow-Lite, Keras, and ONNX model. The DeepView RT Model can be float32 or quantized(int8/uint8) model. For details on model conversion, see eIQ Toolkit document.

Note: *The eIQ Toolkit will be available on NXP eIQ website* https://www.nxp.com/design/software/development-software/eiq-ml-development-environment:EIQ.

5 Run an inference

5.1 Library initialization

The DeepViewRT library header is named *deepview_rt.h* and is the only required header for the C API. The *deepview_ops.h* is also required for cases where operations (layers) are called directly as opposed to strictly under context control using an RTM model.

```
#include "deepview_rt.h"
#include "deepview_ops.h"
```

5.2 Loading model

To load a model, a context object is required to host the model and required runtime buffers.

```
/* DeepViewRT Model definition from model.S */
extern const unsigned char model_rtm_start;
```

```
extern const unsigned char model_rtm_end;
```

```
NNContext *context = nn_context_init(NULL, POOL_SIZE, NULL,
 CACHE_SIZE, NULL);
nn_context_model_load(context, st.st_size, model);
```

5.2.1 DeepviewRT model definition in model.S

```
model_rtm_start:
  .incbin "models/mobilenet_v1_0.25_160.rtm"
model_rtm_end:
```

5.3 Loading image

The **nn_tensor_load_image_ex** function loads the image data and attempts to decode it. The function supports PNG and JPEG images and the format is discovered by reading the buffers headers automatically. If the operation fails, an error is returned.

```
err = nn_tensor_load_image_ex(input, sample_image, (size_t)
 sample_image_size, 2);
```

5.4 Run inference

The **nn_context_run** function performs the actual model evaluation. This evaluates all the layers in the graph. If any error happens on any layer, this function returns an error and more details might be reported to stderr depending on the cause.

```
err = nn_context_run(context);
```

Classification models are typically arranged in a one-hot encoding. The output is a vector representing the known labels, the largest element in this vector represents the inferred label. This "argmax" can be used as an index into the known labels to report a text label result. If a label is not provided, the argmax value is reported. This can also happen if labels are provided but argmax is beyond the provided labels.

```
nn_argmax(output, &argmax, &softmax, sizeof(softmax));
const char *label = nn_model_label(model, argmax);
```

5.5 Replace image and model

Open and edit file model.S inside the source folder. Update incbin parameter to replace the image or model file.

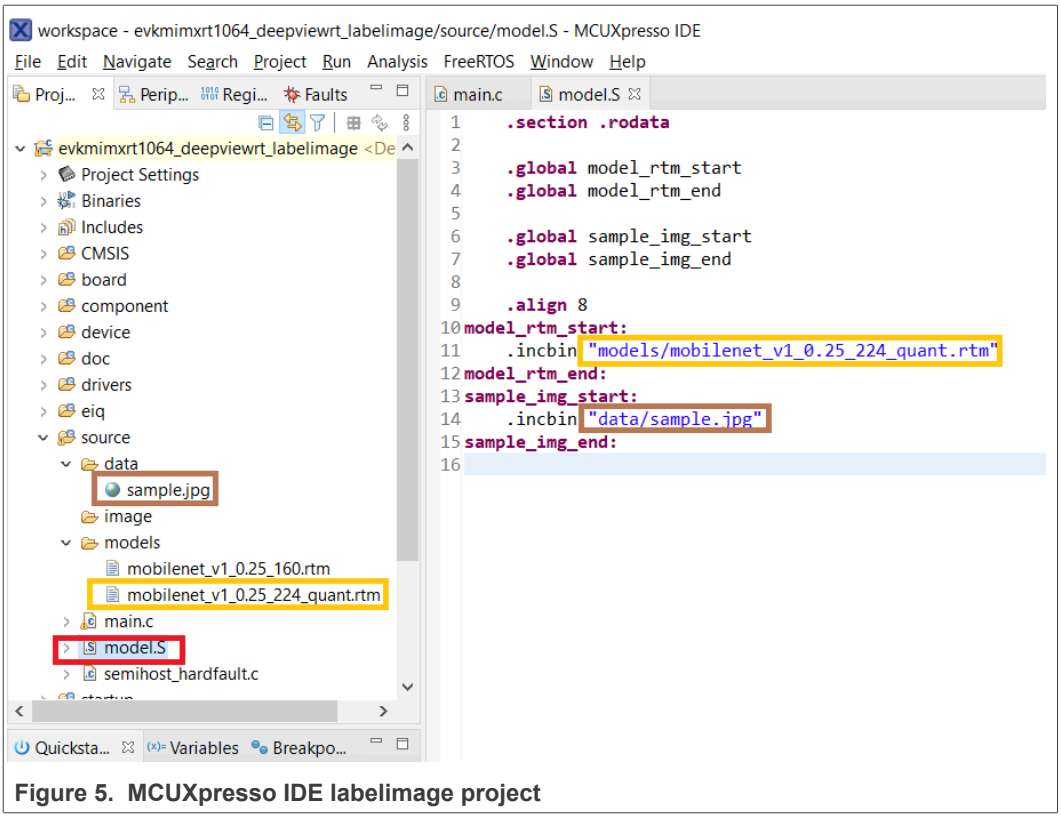

Figure 5. MCUXpresso IDE labelimage project

6 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2021 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- **Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.**
- **Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.**
- **Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.**

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,

WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

7 Revision history

Table 2 summarizes the changes done to this document since the initial release.

Revision history

Revision number	Date	Substantive changes
0	16 April 2021	Initial release
1	05 July 2021	Updated for MCUXpresso SDK v2.10.0
2	06 December 2021	Updated for MCUXpresso SDK v2.11.0
3	01 June 2022	Updated for MCUXpresso SDK v2.12.0
4	08 December 2022	Updated for MCUXpresso SDK v2.13.0

8 Legal information

8.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

8.2 Disclaimers

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.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

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.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

8.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Contents

For more information, please visit: http://www.nxp.com

Date of release: 8 December 2022
Document identifier: EIQDVRTUG
Document number: