User's Manual

CONFIDENTIAL

ADAS Reference Application
AI Library User's Manual

For R-Car V4H2

Rev.0.40.1 Sep. 2023

Renesas Electronics
www.renesas.com

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

How to Use This Manual

1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the MCU. It is intended for users designing application systems incorporating the MCU. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual.

The manual comprises an overview of the product; descriptions of the CPU, system control functions, peripheral functions, and electrical characteristics; and usage notes.

> Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

> The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

2. Notation of Numbers and Symbols

None

3. Register Notation

None

4. List of Abbreviations and Acronyms

Abbreviation	Full Form
DMS	Driver Monitoring System
VIN	Video Input
VOUT	Video Output
ISP	Image Signal Processot
IMP	Image Processing Unit
IMR	Image Renderer
CNN	Convolutional Neural Networks
DU	Display Unit
SDK	Software Development Kit
OSAL	Operating System Abstraction Layer
HIL	Hardware in the loop
CSI-2	Camera Serial Interface 2
V4L2	Video 4 (for) Linux 2
LDC	Lens Distortion Correct
ADAS	Advanced Driver-Assistance Systems
SOC	System On Chip
PMIC	Power Management IC
RGMII	Reduced Gigabit Media-Independent Interface
AI	Artificial Intelligence
KPI	Key Performance Indicators
FPS	Frames Per Second
API	Application Programming Interface
HW	Hardware
BSP	Board Support Package

- Table of Contents -

- List of Figures -

- List of Tables -

1. Overview

This document describes the ADAS reference application add-on package for R-Car platforms. Since the CDNN is available as an add-on package it can be built and linked to the application as a separate module. The package is compatible with the Renesas SDK releases and supports applications built for V4H2 board.

1.1. Contents of the package

The ADAS reference application add-on package will be available with the same folder structure as in R-Car SDK so that it can be easily merged with the SDK root. The detailed installation steps are mentioned in chapter 3.1

1.2. Requirements

Refer section 1.2 of the reference application (DMS / front camera / surround view) user manual for hardware and software development environment requirements.

1.3. Product Overview

The package structure of the ai_lib is shown in Table 1-1. The overview of the ai_lib is shown in Table 1-2 Required components are shown in Table 1-3.

Table 1-1 ai_lib software package structure

Location	Description
rcar-xos	SDK root folder
README_AI_LIB.md	ADAS add-on package documentation
v3.xx.0	Directory corresponding to the SDK version
cmake	Patch for cmake configuration 'rcar-xos-aarch64-gnu-linux-v4h2-export.cmake'
docs/sw/ai_lib/user_manual	AI library user manual directory
R-CarV4H2_ai_lib_User_Manual.pdf	ai_lib user manual
sw/aarch64-gnu-linux	Public header directory
Include/rcar-xos/ai_lib	Include directory for ai_lib
ai_lib.h	Public header
buffer_struct.h	Public header
lib	Directory for build-library files
libai_lib_v4h2.so	Build library file
sw_src/renesas/middleware/libraries/ai_lib	Library source code path
Include/rcar-xos/ai_lib	ai_lib public header directory
ai_lib.h	Public header
buffer_struct.h	Ai buffer structure header
src	Source code
include	Private header directory for ai lib
batch.h	Batch file header
buffer_configuration.h	Buffer configuration header
buffer_struct.h	Buffer structure header
common.h	Common header file
dsp_cdnn_dtcm.h	Dsp dtcm header
dsp_cdnn_dtcm0.h	Dsp dtcm header
dsp_cdnn_dtcm1.h	Dsp dtcm header
dsp_cdnn_dtcm2.h	Dsp dtcm header
dsp_cdnn_dtcm3.h	Dsp dtcm header
dsp_cdnn_extmem.h	Dsp extmem header
dsp_cdnn_extmem0.h	Dsp extmem header

dsp_cdnn_extmem1.h	Dsp extmem header
dsp_cdnn_extmem2.h	Dsp extmem header
dsp_cdnn_extmem3.h	Dsp extmem header
dsp_cdnn_ptcm.h	Dsp ptcm header
dsp_cdnn_ptcm0.h	Dsp ptcm header
dsp_cdnn_ptcm1.h	Dsp ptcm header
dsp_cdnn_ptcm2.h	Dsp ptcm header
dsp_cdnn_ptcm3.h	Dsp ptcm header
helpers.h	Helpers header
imp_demo.h	Imp demo header
imp_demo_fw.h	Imp demo framework header
jsmn.h	Jsmn header
parameters_parsing.h	Parameter parsing header
ai_lib.c	ai application main function
batch.c	Batch functions for application
cdnn_main.c	CDNN main
common.c	File to maintain ai_lib common API's
helpers.c	Other common functions used by the ai_lib
imp_demo_fw	Imp demo framework
imp_demo_sub.c	Memory and input file functions for ai Application
parameters_parsing.c	Parameter parsing
wrapper.c	wrapper file to handle memory management
CMakeLists.txt	CMakelist for ai_lib
module.cmake	Supported SoC, OS info for the library
samples	
dms_ref_app/application/src/cdnn	
include	
buffer_configuration.h	Buffer configuration
cdnn_main.h	CDNN pre-processing/post processing header
dms_v4h2_input.h	CDNN input header files
dms_v4h2_netinfo.h	
src	CDNN pre-processing/post processing
cdnn_main.c	
frontcam_ref_app/application/src/cdnn	
include	
buffer_configuration.h	Buffer configuration
cdnn_main.h	CDNN pre-processing/post processing header
fc_v4h2_objdet_input.h	CDNN input header files
fc_v4h2_objdet_netinfo.h	
fc_v4h2_semseg_input.h	
fc_v4h2_semseg_netinfo.h	
src	
cdnn_main.c	CDNN pre-processing/post processing
surroundview _ref_app/application/src/cdnn	
include	
buffer_configuration.h	Buffer configuration
cdnn_main.h	CDNN pre-processing/post processing header
sv_v4h2_semseg_input.h	CDNN input header files
sv_v4h2_semseg_netinfo.h	
src	
cdnn_main.c	CDNN pre-processing/post processing
build_linux_dev_board.patch	Patch for the application build script

Table 1-2 Overview of AI library

Library Name	ai_lib
Type	Shared Object (.so)
R-Car SDK	sdk1
Target SoC	R-Car V4H2
Target Environment	HIL

(*) The CDNN toolchain is used for generating AI model (QData) for the arch64 environment setup.

Table 1-3 Required components

Name	Description	Remarks
ADAS Reference application	dms_ref_app, frontcam_ref_app	Included in R-Car SDK. Ensure that the platform version is same as the AI Library
AI Library	ai_lib	This package
CDNN package	CEVA-SP_CDNN_ED_03.05.2023	Provided by CEVA. Ask Renesas sales for details
DSP add-on Package	DSP add-on 20230428	Provided by Renesas. Ask Renesas sales for details
CDNN model data		

2. Architecture

2.1. Relationship between AI Library and ADAS Reference Application (V4H2)

The AI function for V4H is separated into another library (ai_lib). The ADAS reference applications use it via ai_lib API. Figure 2-1 shows the relationship between the ADAS reference application, AI library, CDNN package and DSP add-on package.

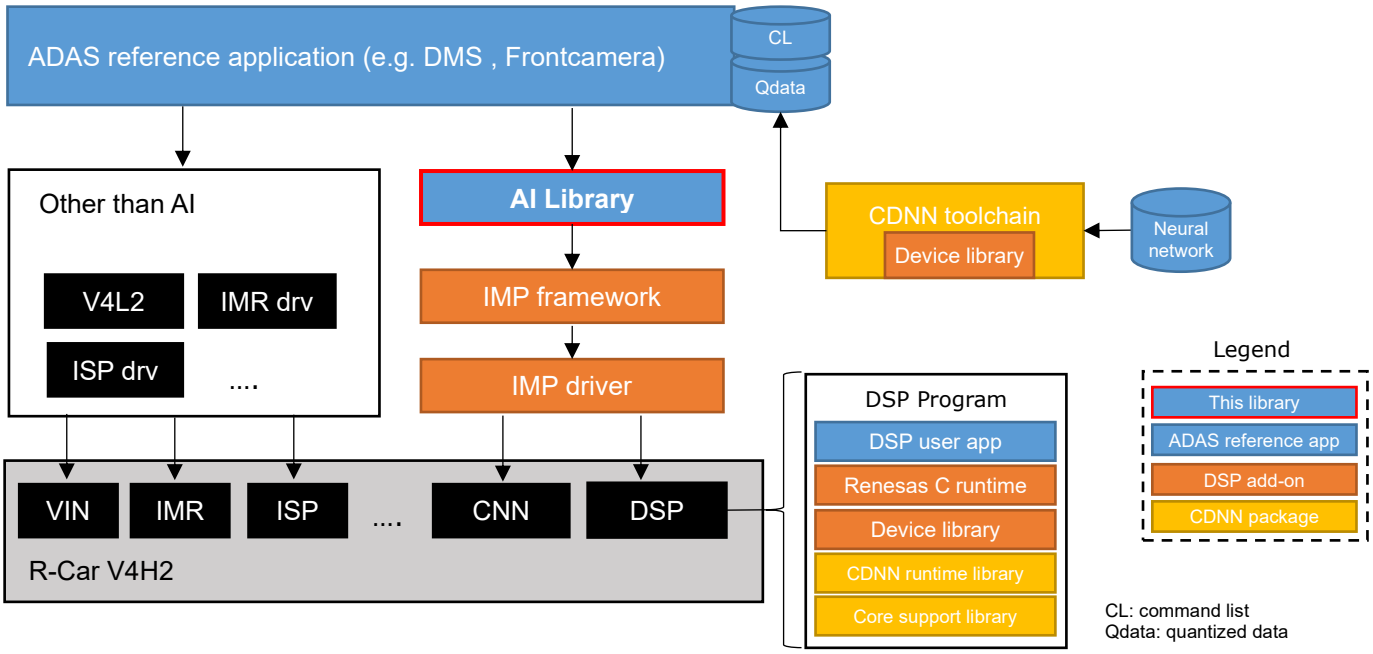

Figure 2-1: ADAS reference application and AI Library

3. Integration guide

This chapter explains how to integrate the dms_ref_app/frontcam_ref_app with the ADAS reference application add-on package.

3.1. Installation

a. Extract the ADAS reference application add-on package (rcar-xos_platform-sdk1_v3.xx.0_addon_ref_app_ai_lib.zip) and merge the extracted folder (rcar-xos) to your SDK root (rcar-xos) folder (Ensure that the SDK version is same with the add-on package).

b. Apply exportcmake.patch and exportreleasecmake.patch in rcar-xos/3.xx.0/cmake/ folder.

   a. On Linux, use the below command.

```
$ patch < “patchname”
```

   b. On Windows, use a merge tool such as WinMerge.

c. Apply patch "build_linux_dev_board.patch" from rcar-xos/3.xx.0/samples folder.
Linux command: patch < build_linux_dev_board.patch

d. (The following steps are optional) Already ai_lib shared library (libai_lib_v4h2.so) is available in the addon package. To rebuild the standalone ai_lib library apply patch ai_lib_cmake.patch in the folder rcar-xos/3.xx.0/sw_src/renesas/middleware/libraries/ai_lib and follow the build procedure as mentioned in ‘step e’ for linux environment and ‘step f’ for windows environment for building middleware libraries. Rebuilding of ai_lib is not always required. (In SDK 3.17.0 or later, the build on Windows does not work. Please build it on Linux)

e. The following steps have to be followed on Linux environment.

   1. Go to ai_lib folder in sw_src directory.

```
cd    /Renesas/rcar-xos/v3.xx.0/sw_src/renesas/middleware/libraries/ai_lib
```

   2. Create a directory with name "build" and open the build folder.

```
mkdir build
```

   3. Open terminal from the folder and give the following command.

```
cmake -G "Unix Makefiles" -DCMAKE_TOOLCHAIN_FILE="/home/quest/Renesas/rcar-xos/v3.xx.0/cmake/toolchain_poky_3_1_11_adas.cmake" -DCMAKE_PREFIX_PATH="/home/quest/Renesas/rcar-xos/v3.xx.0/cmake/" -DRCAR_AI_LIB_TYPE=SOURCE -DRCAR_SOC=V4H2 -DCMAKE_BUILD_TYPE=RELEASE ..
```

   4. Then give "make" command in terminal to build the ai_lib binary (libcustomized_ai_lib_v4h2.so) in the same folder.
   5. For building reference applications with newly created ai_lib binary rename the created binary libcustomized_ai_lib_v4h2.so file to libai_lib_v4h2.so
   6. Copy the binary to sw/aarch64-gnu-linux/lib/ folder and replace the libai_lib_v4h2.so.

f. The following steps have to be followed on Windows environment.

1. Go to ai_lib folder in sw_src directory.

```
cd D:¥Renesas¥rcar-xos¥v3.xx.0¥sw_src¥renesas¥middleware¥libraries¥ai_lib¥
```

2. Create a directory with name "build" and open the build folder.

```
mkdir build
```

3. To generate executable, we need to set the path for dependable libraries.

```
set PATH=%PATH%;D:/Renesas/rcar-xos/v3.xx.0/tools/cmake-3.21.0-windows-x86_64/bin;D:/Renesas/rcar-xos/v3.xx.0/tools/make;D:/Renesas/rcar-xos/v3.xx.0/tools/toolchains/mingw64/bin
```

4. Open terminal from the folder and give the following command.

```
cmake -G "Unix Makefiles" -DCMAKE_TOOLCHAIN_FILE="D:/Renesas/rcar-xos/v3.xx.0/cmake/toolchain_poky_3_1_11_adas.cmake" -DCMAKE_PREFIX_PATH="D:/Renesas/rcar-xos/v3.xx.0/cmake/" -DRCAR_AI_LIB_TYPE=SOURCE -DRCAR_SOC=V4H2 -DCMAKE_BUILD_TYPE=RELEASE ..
```

5. Then give "make" command in terminal to build the ai_lib binary (libcustomized_ai_lib_v4h2.so) in the same folder.
6. For building reference applications with newly created ai_lib binary rename the created binary libcustomized_ai_lib_v4h2.so file to libai_lib_v4h2.so
7. Copy the binary to sw/aarch64-gnu-linux/lib/ folder and replace the libai_lib_v4h2.so.

After the installation, the file tree on the host PC is as follows (**bold**: from ai_lib package).

```
rcar-xos
    ├── README_AI_LIB.md
    └── v3.x.x
        ├── cmake
        │   ├── exportcmake.patch
        │   └── exportreleasecmake.patch
        ├── docs
        │   └── sw
        │       └── ai_lib
        │           └── user_manual
        ├── samples
        │   ├── build_linux_dev_board.patch
        │   ├── dms_ref_app
```

```
│   │   └── application
│   │       └── src
│   │           └── cdnn
│   │               ├── include
│   │               │   ├── buffer_configuration.h
│   │               │   ├── cdnn_main.h
│   │               │   ├── dms_v4h2_input.h
│   │               │   └── dms_v4h2_netinfo.h
│   │               └── src
│   │                   └── cdnn_main.c
│   ├── frontcam_ref_app
│   │   └── application
│   │       └── src
│   │           └── cdnn
│   │               ├── include
│   │               │   ├── buffer_configuration.h
│   │               │   ├── cdnn_main.h
│   │               │   ├── fc_v4h2_objdet_input.h
│   │               │   ├── fc_v4h2_objdet_netinfo.h
│   │               │   ├── fc_v4h2_semseg_input.h
│   │               │   └── fc_v4h2_semseg_netinfo.h
│   │               └── src
│   │                   └── cdnn_main.c
│   └── surroundview_ref_app
│       └── application
│           └── src
│               └── cdnn
│                   ├── include
│                   │   ├── buffer_configuration.h
│                   │   ├── cdnn_main.h
│                   │   ├── sv_v4h2_semseg_input.h
│                   │   └── sv_v4h2_semseg_netinfo.h
│                   └── src
│                       └── cdnn_main.c
├── sw
│   └── aarch64-gnu-linux
│       ├── include
│       │   └── rcar-xos
│       │       └── ai_lib
│       │           ├── ai_lib.h
│       │           └── buffer_struct.h
│       └── lib
│           └── libai_lib_v4h2.so
└── sw_src
    └── renesas
        └── middleware
            └── libraries
                └── ai_lib
                    ├── CMakeLists.txt
                    ├── include
                    │   └── rcar-xos
                    │       └── ai_lib
                    │           ├── ai_lib.h
```

```
│               └── buffer_struct.h
├── module.cmake
└── src
    ├── ai_lib.c
    ├── batch.c
    ├── cdnn_main.c
    ├── common.c
    ├── helpers.c
    ├── imp_demo_fw.c
    ├── imp_demo_sub.c
    ├── include
    │   ├── batch.h
    │   ├── buffer_configuration.h
    │   ├── buffer_struct.h
    │   ├── common.h
    │   ├── dsp_cdnn_dtcm0.h
    │   ├── dsp_cdnn_dtcm1.h
    │   ├── dsp_cdnn_dtcm2.h
    │   ├── dsp_cdnn_dtcm3.h
    │   ├── dsp_cdnn_dtcm.h
    │   ├── dsp_cdnn_extmem0.h
    │   ├── dsp_cdnn_extmem1.h
    │   ├── dsp_cdnn_extmem2.h
    │   ├── dsp_cdnn_extmem3.h
    │   ├── dsp_cdnn_extmem.h
    │   ├── dsp_cdnn_ptcm0.h
    │   ├── dsp_cdnn_ptcm1.h
    │   ├── dsp_cdnn_ptcm2.h
    │   ├── dsp_cdnn_ptcm3.h
    │   ├── dsp_cdnn_ptcm.h
    │   ├── helpers.h
    │   ├── imp_demo_fw.h
    │   ├── imp_demo.h
    │   ├── jsmn.h
    │   └── parameters_parsing.h
    ├── parameters_parsing.c
    └── wrapper.c
```

3.2. Build Application

This chapter explains how to build the dms_ref_app/frontcam_ref_app/surroundview_ref_app with the ADAS reference application add-on package. This is an updated instruction in 4.3 of the dms_ref_app/frontcam_ref_app/surroundview_ref_app user manual.

3.2.1.V4H2 (Linux Environment)

a. Go to the ‘samples’ folder of your SDK and change the user permission before running the script using the below command.

```
$ chmod +x build_linux_dev_board.sh
```

To generate executable, follow one of the below steps.

a. Run the build script build_linux_dev_board.sh for V4H as shown below.

```
$ ./build_linux_dev_board.sh
```

Select the application to build, SoC and build type as shown below.

- Application: <app_name>_ref_app
- SoC: v4h2
- Build Type: debug or release

1) Run the build script build_linux_dev_board.sh along with the command line argument as shown below.

```
$ ./build_linux_dev_board.sh -a <app_name>_ref_app -d v4h2 -b release
```

After running the build script, if the SDK is configured correctly then build will begin as shown in below example.

```
-- Configuring done
-- Generating done
-- Build files have been written to: /home/quest1021377/Renesas/rcar-xos/v3.12.0/samples/dms_ref_app/build_linux_dev_board

==========================================================================
                        Build the Linux HIL sample app
==========================================================================

/home/quest1021377/Renesas/rcar-xos/v3.12.0/tools/cmake-3.21.0-linux-x86_64/bin/cmake -S/home/quest1021377/Renesas/rcar-xos/v
ar-xos/v3.12.0/samples/dms_ref_app/build_linux_dev_board --check-build-system CMakeFiles/Makefile.cmake 0
```

If an issue comes, fix the issue and build the App again (For known build issues refer chapter 4.12 of dms_ref_app user manual).

Successful build will create built target as below:

For release build: **<app_name>_ref_app_v4h2, <app_name>_with_cdnn_ref_app_v4h2**

For debug build: **<app_name>_ref_app_v4h2_d, <app_name>_with_cdnn_ref_app_v4h2_d**

The executable will be generated in specific application path:

samples/<appname>_ref_app/build_linux_dev_board

3.2.2.V4H2 (Windows environment)

a. Open Command prompt from ~/Renesas/rcar-xos/v3.xx.0/samples/<app_name>_ref_app and make new directory build using the below command.

```
Mkdir build
cd build
```

b. To generate executable, we need to set the path for dependable libraries.

```
Set PATH=%PATH%;D:/Renesas/rcar-xos/v3.xx.0/tools/cmake-3.21.0-windows-x86_64/bin;D:/Renesas/rcar-xos/v3.xx.0/tools/make;D:/Renesas/rcar-xos/v3.xx.0/tools/toolchains/mingw64/bin
```

c. Run cmake command as shown below.

```
cmake -G "Unix Makefiles" -DCMAKE_TOOLCHAIN_FILE="../../../cmake/toolchain_poky_3_1_11_adas.cmake" -
```

```
DSDKROOT="D:/Renesas/rcar-xos/v3.xx.0/tools/toolchains/poky"  -DRCAR_PRIVATE_BUILD=ON -
DCMAKE_BUILD_TYPE=RELEASE -DRCAR_SOC=V4H2    ..
```

d. Run build command for dms_ref_app.

```
cmake --build . --target dms_ref_app_v4h2

or

cmake --build . --target dms_with_cdnn_ref_app_v4h2
```

If an issue comes, fix the issue and build the App again (For known build issues refer chapter 4.14 of dms_ref_app user manual).

Successful build will create built target as below:

For release build: **dms_ref_app_v4h2, dms_with_cdnn_ref_app_v4h2**

For debug build: **dms_ref_app_v4h2_d, dms_with_cdnn_ref_app_v4h2_d**

The executable will be generated in specific application path:

build/bin/

e. Run build command for frontcam _ref_app.

```
cmake --build . --target frontcam_ref_app_v4h2

or

cmake –build . –target frontcam_with_cdnn_ref_app_v4h2
```

If an issue comes, fix the issue and build the App again.

Successful build will create built target as below:

For release build: **frontcam_ref_app_v4h2, frontcam_with_cdnn_ref_app_v4h2**

For debug build: **frontcam_ref_app_v4h2_d, frontcam_with_cdnn_ref_app_v4h2_d**

The executable will be generated in specific application path:

build/bin/

f. Run build command for surroundview_ref_app.

```
cmake --build . --target    surroundview_ref_app_v4h2

or

cmake --build . --target    surroundview_with_cdnn_ref_app_v4h2
```

If an issue comes, fix the issue and build the App again.

Successful build will create built target as below:

For release build: **surroundview_ref_app_v4h2, surroundview_with_cdnn_ref_app_v4h2**

For debug build: **surroundview_ref_app_v4h2_d, surroundview_with_cdnn_ref_app_v4h2_d**

The executable will be generated in specific application path:

build/bin/

3.3. Usage

This chapter explains how to use the dms_ref_app with the ADAS reference application add-on package. This is an updated instruction in 4.6 of the dms_ref_app/frontcam_ref_app user manual.

3.3.1.V4H2 – with CDNN (DMS)

a. Create a folder in target board as shown below.

```
$ mkdir dms_ref_app_cdnn
```

b. Copy generated application executable (dms_with_cdnn_ref_app_v4h2) from the output directory (~/samples/dms_app/build_linux_dev_board) to the target board path ~/dms_ref_app_cdnn .

c. Copy given CDNN files:
(~/samples/dms_ref_app/test_data/sample/vgg16/app/)deploy003401901.cdnnQdata, CNN0_hil.bin, SDMAC0.bin, SDMAC1.bin and (~/samples/dms_ref_app/test_data/)weight.bin from test_data folder in the app to the target board path ~/dms_ref_app_cdnn.

d. Copy the files rcar-xos/v3.xx.0/sw/aarch64-gnu-linux/lib/libai_lib_v4h2.so and rcar-xos/v3.xx.0/sw/aarch64-gnu-linux/lib/cdlibadas_ref_fwk_v4h2.so to the target board path ~/dms_ref_app_cdnn.

e. Copy DMS customize file for v4h from **config/dms_customize_v4h.config** to config folder in target board path ~/dms_ref_app/config. If the user wants to customize the configuration, then modify the config file in the board according to chapter 3.5 of dms_ref_app user manual.

f. Copy the test image file **test_data/frame_buffer_vin** and **test_data/Test_Images** folder to the binary path in target board.
The dms_ref_app_cdnn folder structure in target board will be as shown below:

```
dms_ref_app_cdnn
 |-- config/
 |     `-- dms_customize_v4h.config
 |-- dms_with_cdnn_ref_app_v4h2
 |-- frame_buffer_vin
 |-- libadas_ref_fwk_v4h2.so
 |-- libai_lib_v4h2.so
 |     `--dms_v4h2/
 |           `-- app/
 |                 |-- deploy003401901.cdnnQdata
 |                 |-- CNN0.bin
 |                 |-- CNN0.txt
 |                 |-- data_t.bin
 |                 |-- dense_3.bin
 |                 |-- SDMAC0.bin
```

```
|                |-- SDMAC0.txt
|                |-- SDMAC1.BIN
|                |-- SDMAC1.txt
|                `-- weight.bin
`-- Test_Images/
      |-- DW1.rgb.yuv
      |-- HM1.rgb.yuv
      |-- LD1.rgb.yuv
      |-- PCL1.rgb.yuv
      |-- PCR1.rgb.yuv
      |-- RB1.rgb.yuv
      |-- RS1.rgb.yuv
      |-- SD1.rgb.yuv
      `-- TP1.rgb.yuv
```

g. Make sure that CDNN is enabled in the configuration file; if not (i.e. CDNN_Enable 0), edit the **config/dms_customize_v4h.config** file and make CDNN_Enable as 1.

h. Run the binary dms_ref_app_v4h2 using the command below.

```
$ ./dms_with_cdnn_ref_app_v4h2
```

3.3.2.V4H2 – with CDNN (Front camera)

a. Create a folder in target board as shown below.

```
$ mkdir frontcam_ref_app_cdnn
```

b. Copy generated application executable (frontcam_with_cdnn_ref_app_v4h2) from the output directory (~/samples/frontcam_app/build_linux_dev_board) to the target board path ~/frontcam_ref_app_cdnn .

c. Copy given CDNN files: deploy003401901.cdnnQdata, CNN0_hil.bin, SDMAC0.bin, SDMAC1.bin, weight.bin from test_data folder in the app to the target board path ~/frontcam_ref_app_cdnn.

d. Copy the files rcar-xos/v3.xx.0/sw/aarch64-gnu-linux/lib/libai_lib_v4h2.so and rcar-xos/v3.xx.0/sw/aarch64-gnu-linux/lib/ libadas_ref_fwk_v4h2.so to the target board path ~/frontcam_ref_app_cdnn.

e. Copy FC customize file for v4h from **config/frontcam_customize_v4h.config** to config folder in target board path ~/frontcam_ref_app/config. If the user wants to customize the configuration, then modify the config file in the board according to chapter 3.5 of frontcam_ref_app user manual.

f. Copy the test image file **test_data/frame_buffer_vin** and **test_data/Test_Images** folder to the binary path in target board.

The frontcam_ref_app_cdnn folder structure in target board will be as shown below:

```
frontcam_ref_app_cdnn
 |-- config/
 |     `-- frontcam_customize_v4h.config
 |-- frontcam_with_cdnn_ref_app_v4h2
 |-- frame_buffer_vin
 |-- libadas_ref_fwk_v4h2.so
 |-- libai_lib_v4h2.so
 |     `--fc_v4h2/
 |           |-- objdet/
 |           |    |-- deploy003401901.cdnnQdata
 |           |    |-- CNN0.bin
 |           |    |-- CNN0.txt
```

```
|           |   |-- data_t.bin
|           |   |-- dense_3.bin
|           |   |-- SDMAC0.bin
|           |   |-- SDMAC0.txt
|           |   |-- SDMAC1.bin
|           |   |-- SDMAC1.txt
|           |   `-- weight.bin
|           |-- poseest/
|           |   |-- deploy003401901.cdnnQdata
|           |   |-- CNN0.bin
|           |   |-- CNN0.txt
|           |   |-- images_t.bin
|           |   |-- output1_t.bin
|           |   |-- output2_t.bin
|           |   |-- SDMAC0.bin
|           |   |-- SDMAC0.txt
|           |   |-- SDMAC1.bin
|           |   |-- SDMAC1.txt
|           |   `-- weight.bin
|           `-- semseg/
|               |-- deploy003401901.cdnnQdata
|               |-- CNN0.bin
|               |-- CNN0.txt
|               |-- data_t.bin
|               |-- dense_3.bin
|               |-- SDMAC0.bin
|               |-- SDMAC0.txt
|               |-- SDMAC1.bin
|               |-- SDMAC1.txt
|               `-- weight.bin
`-- Test_Images/
      |-- FC_1.yuv
      |-- FC_2.yuv
      |-- FC_3.yuv
      `-- FC_4.yuv
```

g. Make sure that CDNN is enabled in the configuration file; if not (i.e. CDNN_Enable 0), edit the **config/frontcam_customize_v4h.config** file and make CDNN_Enable as 1.

h. Run the binary frontcam_ref_app_v4h2 using the command below.

```
$ ./frontcam_with_cdnn_ref_app_v4h2
```

3.3.3.V4H2 – with CDNN (Surround View)

i. Create a folder in target board as shown below.

```
$ mkdir surroundview_ref_app_cdnn
```

j. Copy generated application executable (surroundview_with_cdnn_ref_app_v4h2) from the output directory (~/samples/surroundview_ref_app/build_linux_dev_board) to the target board path ~/surroundview_ref_app_cdnn .

k. Copy given CDNN files: deploy003401901.cdnnQdata, CNN0_hil.bin, SDMAC0.bin, SDMAC1.bin, weight.bin from test_data folder in the app to the target board path ~/ surroundview_ref_app_cdnn.

l. Copy the files rcar-xos/v3.xx.0/sw/aarch64-gnu-linux/lib/libai_lib_v4h2.so and rcar-xos/v3.xx.0/sw/aarch64-gnu-linux/lib/ libadas_ref_fwk_v4h2.so to the target board path ~/surroundview_ref_app_cdnn.

m. Copy surround view customize file for v4h from **config/surroundview_customize_v4h.config** to config folder in target board path ~/surroundview_ref_app/config. If the user wants to customize the configuration, then modify the config file in the board according to chapter 3.5 of surroundview_ref_app user manual.

n. Copy the test image file **test_data/frame_buffer_vin** and **test_data/Test_Images** folder to the binary path in target board.

The surroundview_ref_app_cdnn folder structure in target board will be as shown below:

```
surroundview_ref_app_cdnn
 |-- config/
 |     `-- surroundview_customize.config
 |-- surroundview_with_cdnn_ref_app_v4h2
 |-- frame_buffer_vin
 |-- libadas_ref_fwk_v4h2.so
 |-- libai_lib_v4h2.so
 |     `--sv_v4h2/
 |           |-- objdet/
 |           |    |-- deploy003401901.cdnnQdata
 |           |    |-- CNN0.bin
 |           |    |-- CNN0.txt
 |           |    |-- SDMAC0.bin
 |           |    |-- SDMAC0.txt
 |           |    |-- SDMAC1.bin
 |           |    |-- SDMAC1.txt
 |           |    `-- weight.bin
 |           `-- semseg/
 |                |-- deploy003401901.cdnnQdata
 |                |-- CNN0.bin
 |                |-- CNN0.txt
 |                |-- SDMAC0.bin
 |                |-- SDMAC0.txt
 |                |-- SDMAC1.bin
 |                |-- SDMAC1.txt
 |                `-- weight.bin
 `-- Test_Images/
       |-- SV_Back_1.yuv
       |-- SV_Front_1.yuv
       |-- SV_Left_1.yuv
       `-- SV_Right_1.yuv
```

o. Make sure that CDNN is enabled in the configuration file; if not (i.e. CDNN_Enable 0), edit the **config/surroundview_customize_v4h.config** file and make CDNN_Enable as 1.

p. Run the binary surroundview_ref_app_v4h2 using the command below.

```
$ ./surroundview_with_cdnn_ref_app_v4h2
```

4. AI Model Customization Guide

4.1. Model preparation

4.1.1.DMS

More details about the model preparation, refer to the chapter 4.12 of dms_ref_app user manual.
More details about the model customization, refer to the chapter 5.1.2 of dms_ref_app user manual.

4.1.2. FC

More details about the model preparation, refer to the chapter 4.12 of frontcam_ref_app user manual.
More details about the model customization, refer to the chapter 5.2 of frontcam _ref_app user manual.

4.1.3.Surround View

More details about the model preparation, refer to the chapter 4.12 of surroundview_ref_app user manual.
More details about the model customization, refer to the chapter 5.2 of surroundview_ref_app user manual.

4.2. Model Conversion Using CDNN Compiler – V4H2

CEVA CDNN compiler supports the conversion of CNN models to executables for the Renesas R-Car Boards.

- CDNN generator converts a network that was created using an external framework ONNX into a CDNN-compatible network (Qdata). Please refer to “CDNN operation guide linux” for detail.
- Generated QData and CL files are converted to a C header file
- The header C file is used for inference execution on V4H2.
- Executables can be generated for both hardware and simulated environment.
- Inference of generated outputs Executable **Qdata** can be verified using Runtime in CDNN.
- Executable will contain the network architecture, weight values and memory allocation details for IMP-CNN/DSP.
- Executable can be run on V4H2 board, SIL and PC for the CNN inference.

Conversion using CDNN compiler is explained in the figure below.

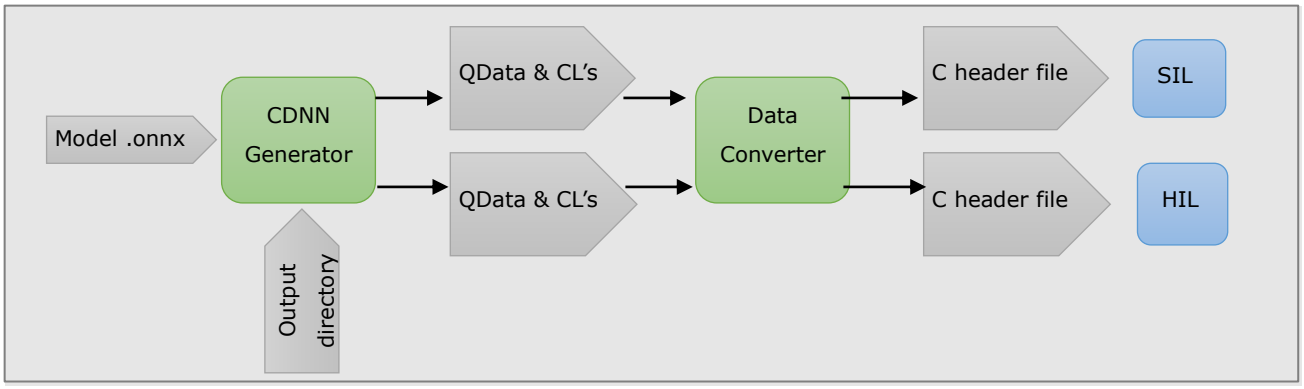

Figure 4-1: CDNN Compiler

For more details on CDNN, Qdata generation, refer to the documents below.

- In CEVA CDNN toolchain
  - CDNN product guide
- In CDNN Renesas Library package
  - CDNN operation guide linux
  - V4H_CNN-IP_SupportParam_List

Revision History	**ADAS Reference Application** **AI Library User's Manual**

Rev.	Date	Status	Description
0.10	May. 23, 2023	Released	Newly created
0.20	Jun. 20, 2023	Released	1.3 ▪ Table 1-1 is updated ▪ Table 1-3 is updated 3: ▪ The procedure of how to integrate is updated ▪ Contents for the front camera is added
0.30	Jul. 13, 2023	Released	1.3: Table 1-1 is updated 3 ▪ The procedure of how to integrate is updated ▪ Contents for the surround view is added
0.40	Aug. 7, 2023	Released	1.3: Table 1-1 is updated 3: The procedure of how to integrate is updated
0.50	Sep. 12, 2023	Released	3.1: The restriction described in d. is updated. The command in c. e. and f.

ADAS Reference Application AI Library

Publication Date: Rev.0.40.1 Sep. 13, 2023

Published by: Renesas Electronics Corporation

ADAS Reference Application
AI Library

Renesas Electronics Corporation