

AN14240

i.MX 8MP AI Robot Autonomous Navigation with vSLAM

Rev. 1.0 — 11 March 2024

Application note

Document information

Information	Content
Keywords	AN14240, i.MX 8MP AI Robot, vSLAM ROS, i.MX 8MP EVK
Abstract	This document describes the autonomous navigation application based on the i.MX 8MP AI Robot board and the i.MX Robot software platform.



1 Introduction

This document describes the autonomous navigation application based on the i.MX 8MP AI Robot board and the i.MX Robot software platform. If you don't have the AI Robot board, you can still build and run some vSLAM ROS demos on the i.MX 8MP EVK platform.

2 Hardware platform

This section describes the hardware platform.

2.1 i.MX 8M AI Robot board

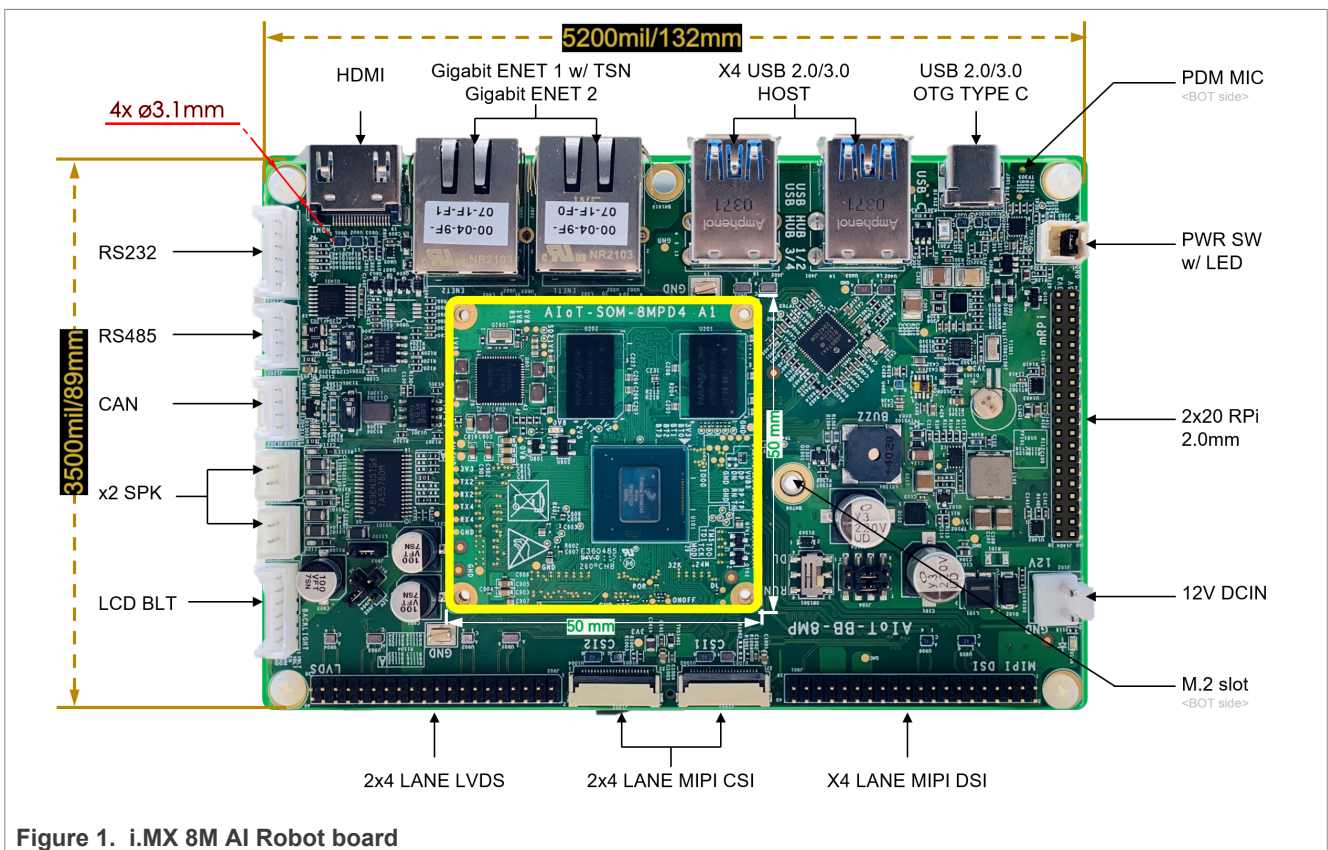


Figure 1. i.MX 8M AI Robot board

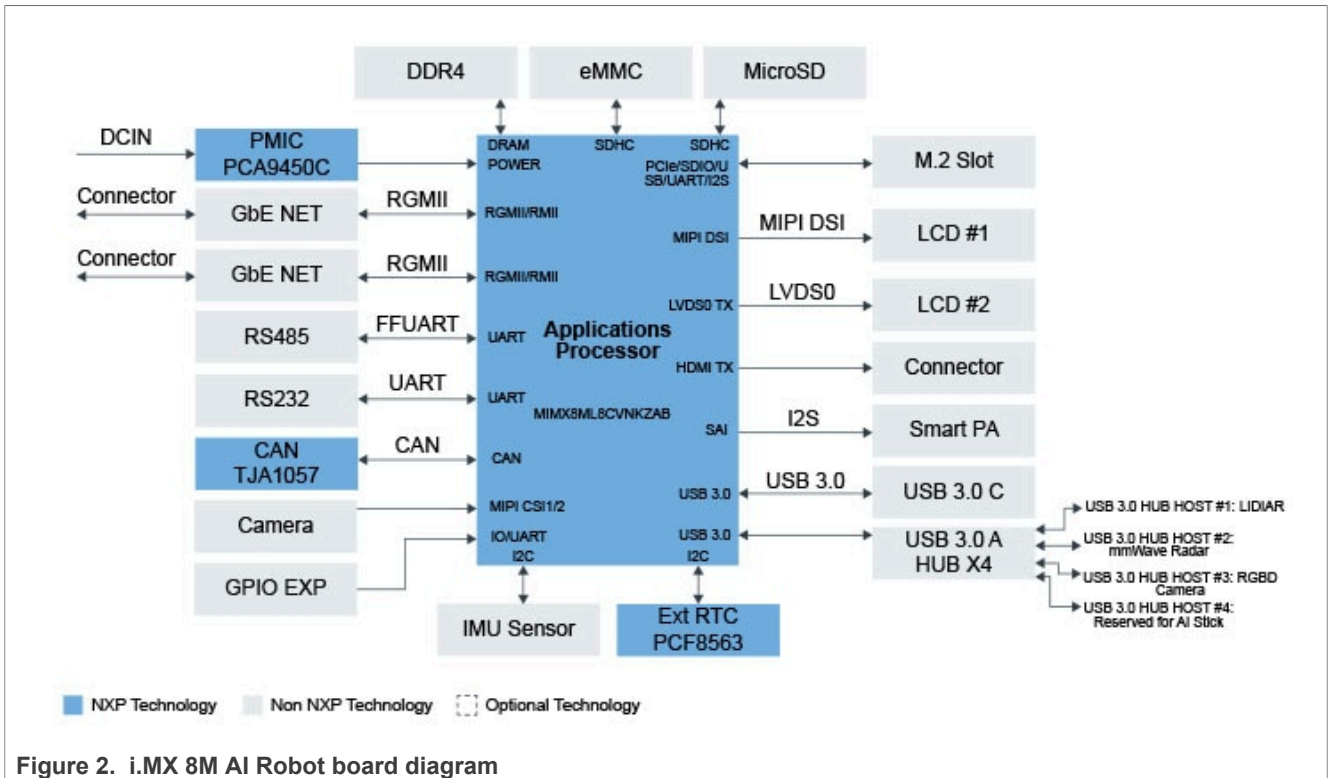


Figure 2. i.MX 8M AI Robot board diagram

2.2 Intel Realsense D455 and smart car



Figure 3. Intel Realsense D455

You can find the information about the Intel Realsense D455 at the following web page: <https://www.intelrealsense.com/depth-camera-d455>.



Figure 4. SeekFree Smart Car

You can buy the Smart Car, the i.MX RT1064 control board, batteries, and other accessories.

2.3 Hardware assembling

You can assemble the RGB-D camera, the AI Robot board, and the Smart Car as shown in [Figure 5](#). You may have to modify the `imxaibot1.urdf.xacro` file in the `imx-aibot1_description` folder or the `imxaibot.urdf` file in the `imx-aibot2_description` folder, based on the form factor of your smart car.

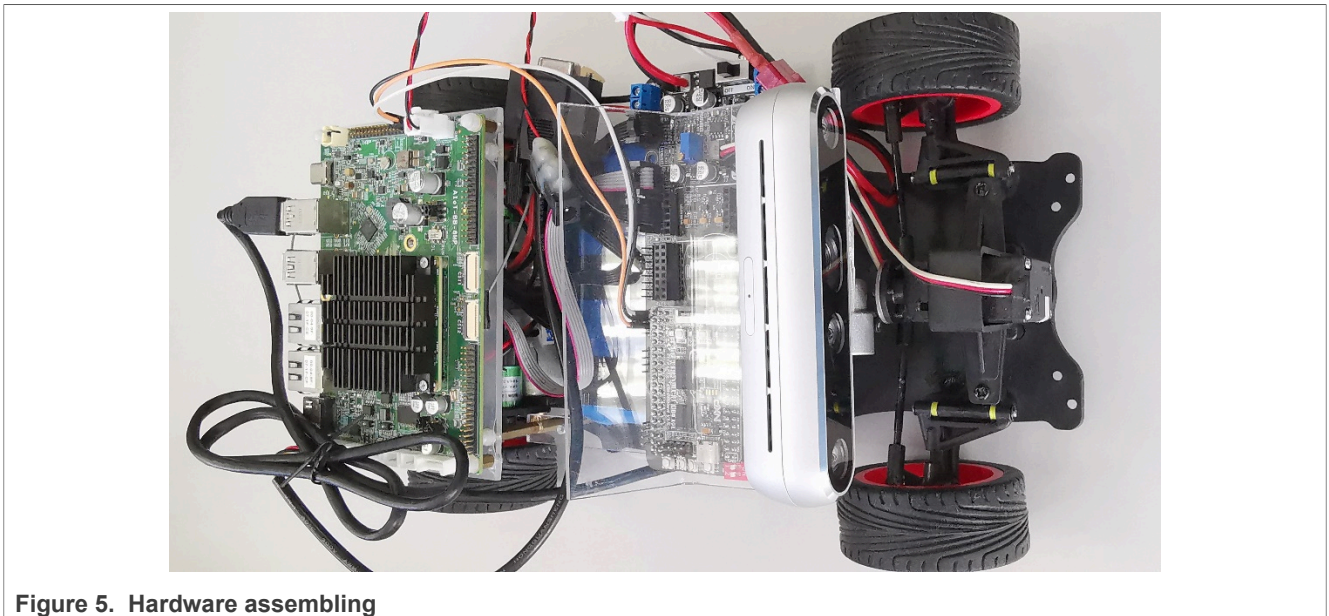


Figure 5. Hardware assembling

3 Software architecture

This section describes the software architecture.

3.1 i.MX robot platform architecture

Figure 6 shows the i.MX robot platform architecture.

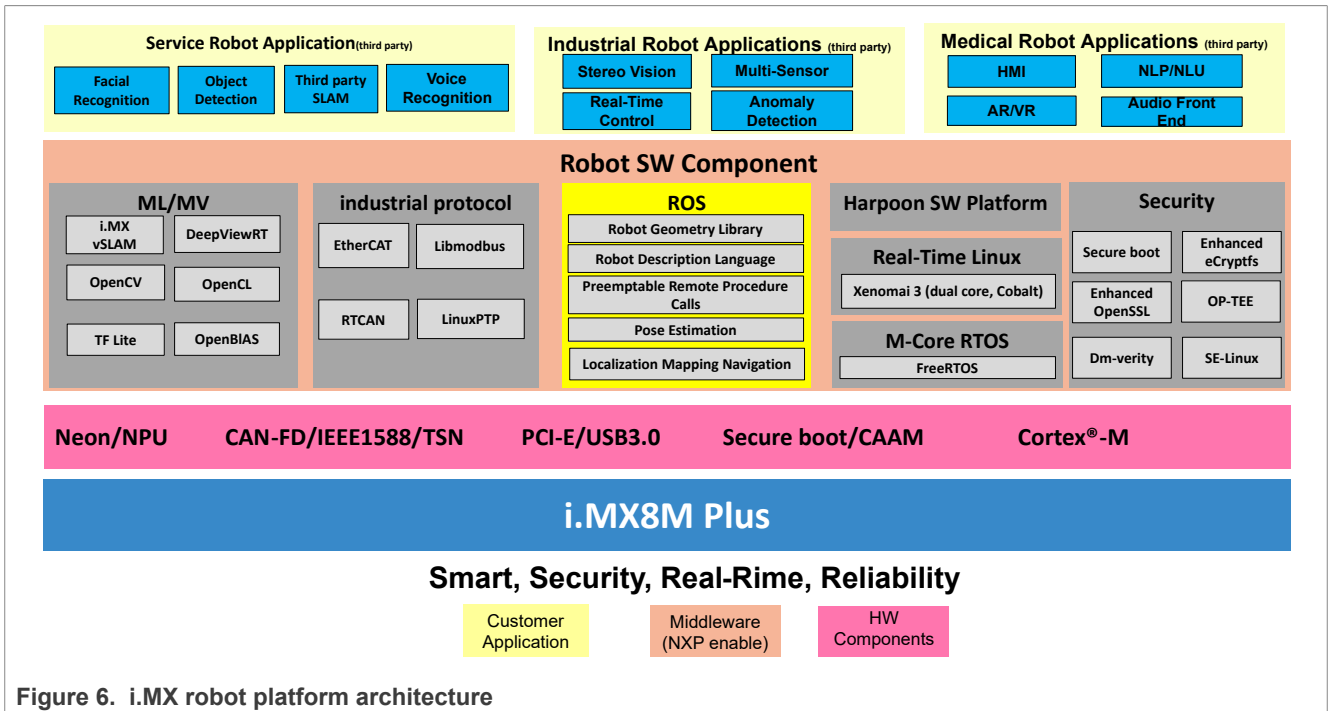


Figure 6. i.MX robot platform architecture

3.2 i.MX vSLAM SDK architecture

Figure 7 shows the i.MX vSLAM SDK architecture.

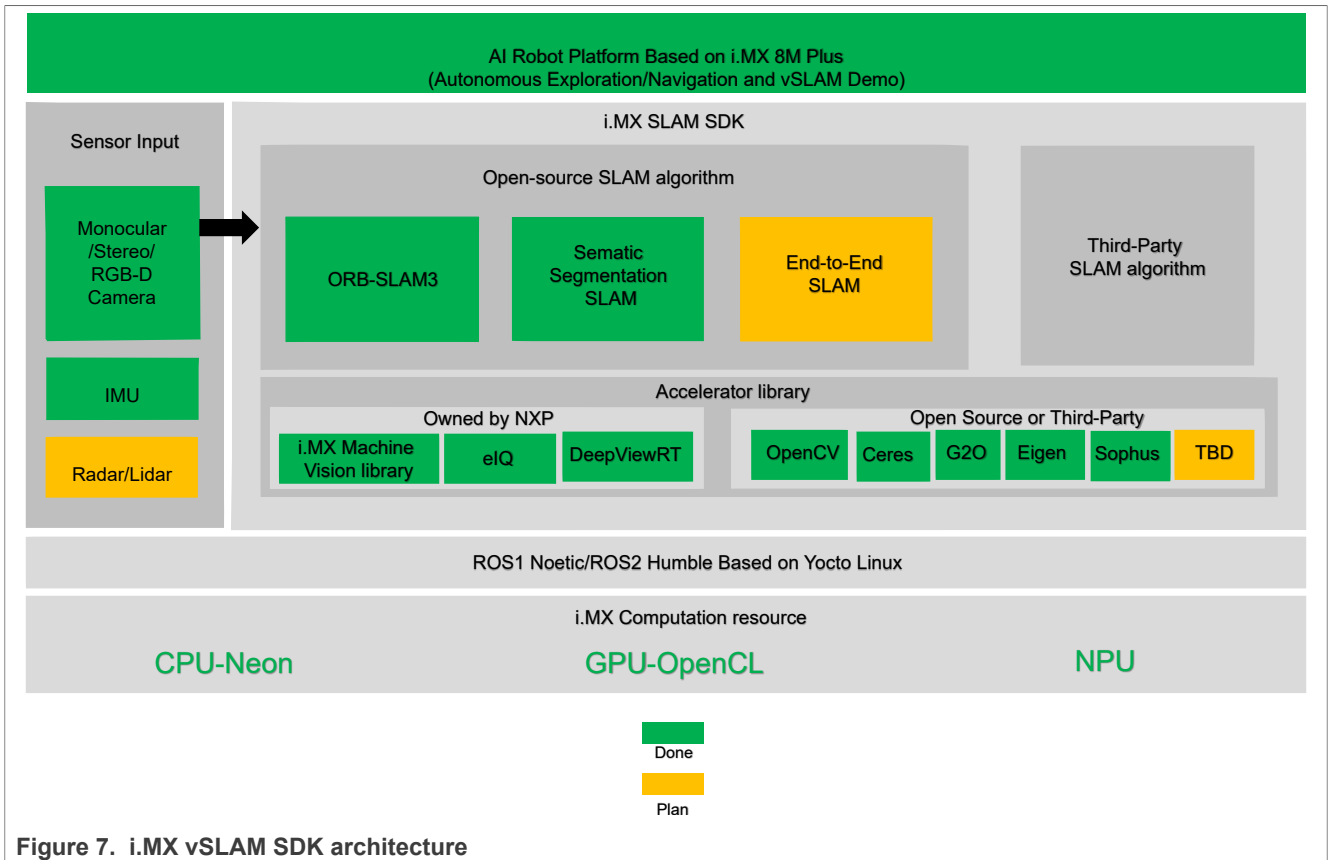


Figure 7. i.MX vSLAM SDK architecture

3.3 Software directory tree

The software directory tree is as follows:

```
meta-robot-platform/
|-- imx
| |-- meta-imx8mp-ai-robot /* BSP patches of i.MX8M Plus AI Robot */
| | |-- conf
| | |-- EULA.txt
| | |-- recipes-bsp
| | |-- recipes-kernel
| | |-- recipes-security
| | |-- SCR.txt
| | |-- setup
| |-- meta-robot /* i.MX Robot Yocto layer */
| | |-- conf
| | |-- COPYING.MIT
| | |-- README
| | |-- recipes-core /* Build Image type */
| | |-- recipes-demo /* demo folder for ROS1 and ROS2*/
| | |-- recipes-devtools
| | |-- recipes-industrial-protocol
| | |-- recipes-machine-vision /* OpenCV 3.4.15 for ROS1*/
| | |-- recipes-nnstreamer
| | |-- recipes-real-time /* RT Linux Xenomai */
| | |-- recipes-security /* Enhanced OpenSSL and GmSSL */
| | |-- recipes-slam-sdk /* i.MX SLAM SDK */
| | |-- recipes-support
```

```
| | `-- tools
| | `-- meta-ros /* ROS's Yocto layer */
| | -- meta-python2
| | -- meta-ros1
| | -- meta-ros1-noetic
| | -- meta-ros2
| | -- meta-ros2-foxy
| | -- meta-ros2-humble
| | -- meta-ros-common
| | -- meta-ros-python2
| | `-- scripts
| | -- LICENSE.txt
| | `-- SCR.txt
```

4 Building and running the demo

This section explains how to build and run the demo.

4.1 Setup build environment for Yocto project

This section describes the setup build environment for the Yocto project.

4.1.1 Downloading the repository (if needed)

Download the repository as follows:

```
$ mkdir ~/bin (this step may not be needed if the bin folder already exists)
$ curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
$ chmod a+x ~/bin/repo
$ export PATH=~/bin:$PATH
```

4.1.2 Setting up the Git (if needed)

Set up the Git as follows:

```
$ git config --global user.name "Your Name"
$ git config --global user.email "Your Email"
$ git config -list
```

4.1.3 Creating the Yocto build environment

Create the Yocto build environment as follows:

```
$ mkdir imx-yocto-bsp
$ cd imx-yocto-bsp
$ repo init -u https://github.com/nxp-imx/imx-manifest -b imx-linux-kirkstone -m
imx-5.15.71-2.2.0.xml
$ repo sync
```

4.2 Build the image

This section describes how to build the image.

4.2.1 Download meta-robot-platform from GitHub

Download the meta-robot-platform from GitHub as follows:

```
git clone -b imx-aibot-release_v3.3 https://github.com/nxp-imx-support/meta-robot-platform.git
```

4.2.2 Adding the meta-robot-platform layer to your build

Add the meta-robot-platform layer to your build as follows:

1. Copy the meta-robot-platform into the <i.MX Yocto folder>/source folder.
2. Create a symbol link as follows: setup-imx-robot.sh -> sources/meta-robot-platform/imx/meta-robot/tools/setup-imx-robot.sh.

4.3 Building the robot image

Build the robot image as follows:

```
$ DISTRO=imx-robot-xwayland MACHINE=imx8mp-ai-robot source setup-imx-robot.sh -r  
noetic-b imx8mp-ai-robot-noetic  
[or DISTRO=imx-robot-xwayland MACHINE=imx8mp-ai-robot source setup-imx-robot.sh  
-r foxy -b imx8mp-ai-robot-foxy ]  
[or DISTRO=imx-robot-xwayland MACHINE=imx8mp-ai-robot source setup-imx-robot.sh  
-r humble -b imx8mp-ai-robot-humble ]  
$ bitbake imx-robot-sdk  
[or bitbake imx-robot-core ]  
[or bitbake imx-robot-system ]  
[or bitbake imx-robot-agv ]
```

If you add `XENOMAI_KERNEL_MODE = "cobalt"` or `XENOMAI_KERNEL_MODE = "mercury"` into the `local.conf` file (by default `XENOMAI_KERNEL_MODE = "mercury"`), you can also build a real-time image with Xenomai using the following command:

```
bitbake imx-robot-core-rt  
[or bitbake imx-robot-system-rt ]
```

If you don't have the AI Robot board, you can also build the image for the EVK board.

```
$ DISTRO=imx-robot-xwayland MACHINE=imx8mpevk source setup-imx-robot.sh -r  
noetic-b imx8mpevk-robot-noetic  
[or DISTRO=imx-robot-xwayland MACHINE=imx8mpevk source setup-imx-robot.sh -r  
foxy -b imx8mpevk-robot-foxy ]  
[or DISTRO=imx-robot-xwayland MACHINE=imx8mpevk source setup-imx-robot.sh -r  
humble -b imx8mpevk-robot-humble ]  
$ bitbake imx-robot-sdk  
[or bitbake imx-robot-core ]  
[or bitbake imx-robot-system ]  
[or bitbake imx-robot-agv ]
```

If you add `XENOMAI_KERNEL_MODE = "cobalt"` or `XENOMAI_KERNEL_MODE = "mercury"` into the `local.conf` file (by default `XENOMAI_KERNEL_MODE = "mercury"`), you can also build a real-time image with Xenomai using the following command:

```
bitbake imx-robot-core-rt
```



```
[or bitbake imx-robot-system-rt ]
```

4.4 Robot image sanity testing

Flush the image in the <i.MX Yocto folder>/<build folder>/tmp/deploy/images/imx8mp-ai-robot/ folder before the sanity test.

```
//ROS1 Sanity Test
#source /opt/ros/noetic/setup.sh
#echo $LD_LIBRARY_PATH
#roscore &
#roscpp list
#roscpp info /rosout
//ROS2 Sanity Test
#source ros_setup.sh
#echo $LD_LIBRARY_PATH
#ros2 topic list
#ros2 interface list
#(sleep 5; ros2 topic pub /chatter std_msgs/String "data: Hello world") &
#ros2 topic echo /chatter
```

4.5 Running the i.MX AIBot navigation demo

Flush the `imx-robot-agv` or `imx-robot-sdk` wic image first. If you want to use RVIZ to observe the status of the robot or set a goal, set your environment in both the Ubuntu PC and the AI Robot board.

For ROS1, export `ROS_MASTER_URI` and `ROS_HOSTNAME` for both the Ubuntu PC and the AI Robot board.

```
//on UART console of AI Robot
#source /opt/ros/noetic/setup.sh
#roslaunch imx_aibot1_navigation imx_aibot1_navigation.launch
//on SSH console of AI Robot after login board by WiFi network
#source /opt/ros/noetic/setup.sh
#roslaunch imx_aibot1_teleop imx_aibot1_teleoperator.launch -screen
```

You can control the Smart Car using a keyboard in the SSH console of the AI Robot or set a goal in the RVIZ at this time.

If you want to try the autonomous exploration feature, see https://github.com/fazildgr/ros_autonomous_slam to set the Exploration region for the RRT in the RVIZ Window. The `imx_aibot1.rviz` file (in the `imx_aibot1_exploration/rviz/` folder) can be used to configure the RVIZ.

The `roslaunch imx_aibot1_exploration RRT.launch` file must be started before setting it in the RVIZ.

For ROS2, use the following command:

```
#source ros_setup.sh
#ros2 launch imx_aibot2_navigation navigation2.launch.py
#python3 /usr/lib/imx_aibot2_teleop/imx_aibot2_teleoperator.py
```

Note: Download the `orbvoc.dbow3` file from <https://github.com/rmsalinas/DBow3/blob/master/orbvoc.dbow3> and copy it to the `/opt/ros/noetic/share/imx_aibot1_vslam/param/` (ROS1) or `/usr/share/imx_aibot2_vslam/param/` (ROS2) folder on your board.

Install `scikit-learn` by running the following command on your board:

```
#pip3 install --trusted-host pypi.org --trusted-host files.pythonhosted.org
scikit-learn
#pip3 install scikit-learn
```

4.6 Running the vSLAM ROS demo

After subscribing to the topic (Image, Pointcloud2, Path, and TF) from `imx-vslam-ros1-demo` or `imx-vslam-ros2-demo`, the rendering results of feature points can be observed from the RVIZ or RVIZ 2 software. The white map point cloud and the green camera trajectory information generated on the right side of the window can also be observed as the camera moves.

```
//ROS1
#source /opt/ros/noetic/setup.sh
#roslaunch imx_vslam_ros1_demo imx_rgbd_d455.launch --screen
//ROS2
#source ros_setup.sh
#ros2 launch imx_vslam_ros2_demo imx_rgbd_d455.py
```

Note: Download the `orbvoc.dbow3` file from <https://github.com/rmsalinas/DBow3/blob/master/orbvoc.dbow3> and copy it to the `/opt/ros/noetic/share/imx-vslam-ros1-demo/param/` (ROS1) or `/usr/share/imx-vslam-ros2-demo/param/` (ROS2) folder on your board.

5 Note about the source code in the document

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

Copyright 2024 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. 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 must be provided with the distribution.
3. 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.

6 Revision history

[Table 1](#) summarizes the revisions to this document.

Table 1. Revision history

Document ID	Release date	Description
AN14240 v.1.0	11 March 2024	Initial public release

Legal information

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.

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 <https://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 document 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.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

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.

i.MX — is a trademark of NXP B.V.

Intel, the Intel logo, Intel Core, OpenVINO, and the OpenVINO logo — are trademarks of Intel Corporation or its subsidiaries.

Microsoft, Azure, and ThreadX — are trademarks of the Microsoft group of companies.

Contents

1 Introduction 2

2 Hardware platform 2

2.1 i.MX 8M AI Robot board 2

2.2 Intel Realsense D455 and smart car 3

2.3 Hardware assembling 4

3 Software architecture 4

3.1 i.MX robot platform architecture 5

3.2 i.MX vSLAM SDK architecture 5

3.3 Software directory tree 6

4 Building and running the demo 7

4.1 Setup build environment for Yocto project 7

4.1.1 Downloading the repository (if needed) 7

4.1.2 Setting up the Git (if needed) 7

4.1.3 Creating the Yocto build environment 7

4.2 Build the image 7

4.2.1 Download meta-robot-platform from GitHub 8

4.2.2 Adding the meta-robot-platform layer to
your build 8

4.3 Building the robot image 8

4.4 Robot image sanity testing 9

4.5 Running the i.MX AIBot navigation demo 9

4.6 Running the vSLAM ROS demo 10

5 Note about the source code in the
document 10

6 Revision history 10

Legal information 12

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.