Document identifier: SLN-VIZN3D-IOT-UG
Rev. 0, 01 November 2021

SLN-VIZN3D-IOT Kit User Guide

Contents

Chapter 1 Introduction..... 4
1.1 RT117F vision crossover processor overview.....4
1.2 Additional peripherals.....4

Chapter 2 Recommendations..... 7
2.1 Recommended computer configuration.....7
2.2 Recommended phone configuration.....7
2.3 Usage condition.....7

Chapter 3 Getting started with SLN-VIZN3D-IOT..... 8
3.1 Box contents.....8
3.2 Running the preconfigured smart-lock application..... 8

Chapter 4 Local face management.....10
4.1 Registering a face.....10
4.2 Deregistering a face..... 12
4.3 Liveness detection.....14

Chapter 5 Remote face management.....16
5.1 Installing the Smart Lock Android app manually..... 16
5.2 Managing kits..... 17
5.3 Managing users.....19
5.3.1 Adding users..... 19
5.3.2 Modifying users.....21
5.3.3 Deleting users..... 22

Chapter 6 Additional features..... 24
6.1 Low-power mode.....24
6.1.1 Pushbuttons..... 24
6.1.2 PIR sensor..... 25
6.1.3 Automatically enable low-power mode.....26
6.2 Serial command line interface..... 28
6.2.1 Connect to the serial command line interface.....28
6.2.2 List available commands.....29
6.3 Manage user database.....30
6.3.1 List all registered users..... 30
6.3.2 Manually add users.....31
6.3.3 Manually delete users..... 31
6.3.4 Rename users.....32
6.3.5 Manually save users into flash.....32
6.4 Display configuration..... 32
6.5 Camera configuration..... 33
6.6 Audio configuration.....34
6.7 Configure LED brightness..... 35
6.7.1 IR LED.....35
6.7.2 White LED.....36

6.8 Configure automatic low-power mode....36
6.9 Debug logging messages....36
6.10 Get BLE address....40
6.11 Get managers and manager components running....40
6.12 Print version information....41
6.13 Reset the SLN-VIZN3D-IOT....41
6.14 Terminate the SLN-VIZN3D-IOT CLI....42

Chapter 7 Troubleshooting....43
7.1 Registering a face....43
7.1.1 Adjust face proximity and position....43
7.1.2 Debug using log messages....43

Chapter 8 Document details....44
8.1 References....44
8.2 Acronyms and definitions....44

Chapter 9 Revision history....46

Chapter 1
Introduction

The SLN-VIZN3D-IOT development kit implements NXP's Edge Ready turnkey solution for 3D face recognition-based access control, using a 3D Structured Light Module (SLM) camera. This solution is enabled by a high-performance member of the i.MX RT1170 microcontroller (MCU) family, running a Real-Time Operating System (RTOS) at up to 1 GHz clock rates, with 2 MB on-chip SRAM. This 3D face recognition solution Software Development Kit (SDK), also called SLN-VIZN3D-IOT, provides OEMs with a fully integrated, self-contained, software, and hardware solution that includes pre-integrated machine learning face-recognition algorithms, as well as all required drivers for all necessary peripherals, including memories, cameras, display, and Bluetooth Low Energy (BLE) chipset. A runtime license to this software solution is bundled with the i.MX RT117F-numbered devices in the i.MX RT1170 family. This cost-effective, easy-to-use solution facilitates the deployment of highly accurate face recognition with robust liveness detection capability, that leverages 3D SLM cameras to operate effectively in even the most challenging lighting conditions. By leveraging an MCU platform, this solution is able to deliver the low cost and low power consumption required for battery powered consumer smart locks, combined with the quick inferencing and short boot times needed to deliver a great user experience.

TARGET APPLICATIONS

- **Smart door locks:** for consumer and hospitality applications, including single family homes, multiple dwelling units, and hotels.
- **Access control:** for office and industrial smart-building applications.

1.1 RT117F vision crossover processor overview

The i.MX RT117F is an Edge Ready member of the i.MX RT1170 family of crossover processors, targeting secured embedded 3D face-recognition applications. It features NXPs advanced implementation of the Arm® Cortex®-M7 core, which operates at speeds up to 1 GHz to provide high CPU performance and best realtime response. This i.MX RT117F-based solution enables system designers to add face recognition easily and inexpensively with 2D and 3D liveness detection capabilities for a wide variety of smart home and access control products in smart homes, smart industrial, and smart building applications. The i.MX RT117F processor is licensed to run NXP's i.MX RT run-time library for 3D face recognition, which includes:

- Camera drivers
- Image capture
- Image pre-processing
- Face alignment
- Face tracking
- Face detection
- Face recognition
- Liveness detection

1.2 Additional peripherals

In addition to the i.MX RT117F microcontroller, the main board embeds all the additional components, modules, and connectors required to create a secured and connected application. These peripherals are shown in Figure 1.

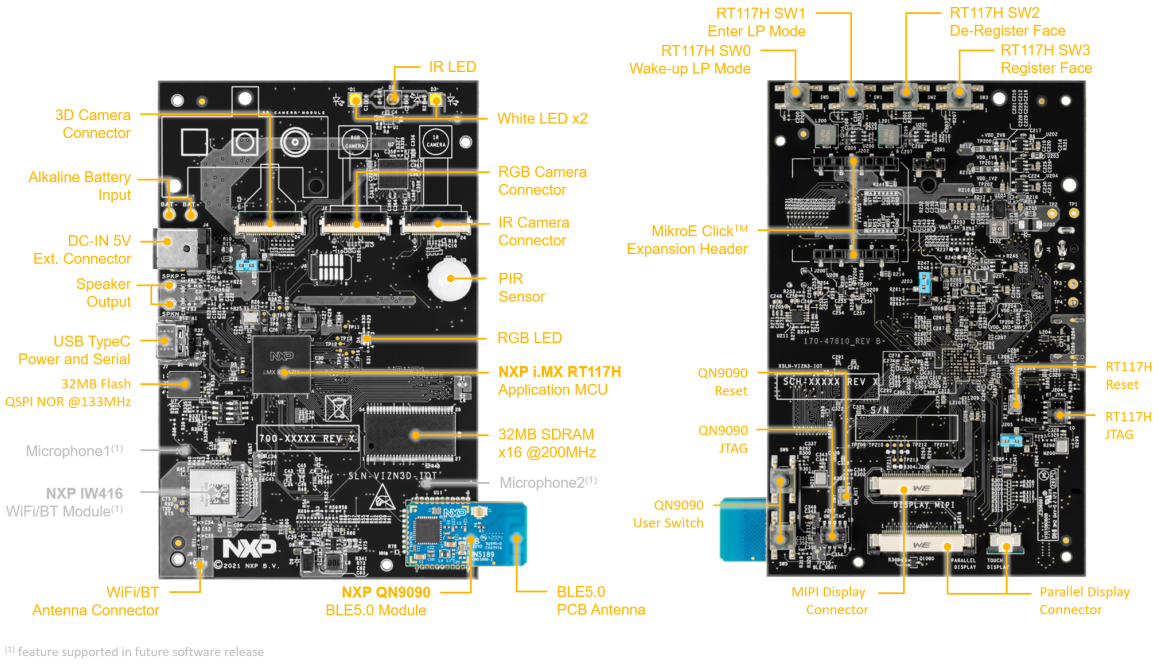

Figure 1. SLN-VIZN3D-IOT main board callout

The SLN-VIZN3D-IOT kit comes fully assembled with both a 3D camera and an RGB camera module, a 2.4 inch TFT display, and a speaker to allow for rapid prototyping of any smart lock project with 3D face-recognition capabilities. These additional peripherals are shown in Figure 2.

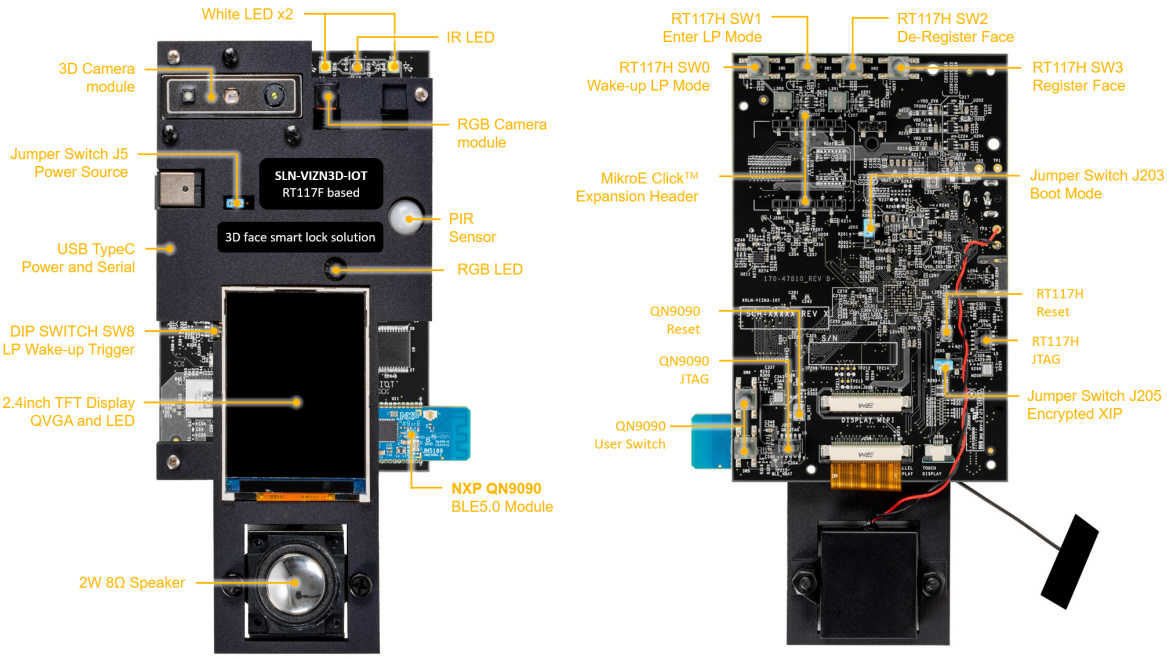

Figure 2. SLN-VIZN3D-IOT kit callout

CAUTION

Depending on the mode of operation, the VIZN3D kit can emit highly concentrated white or non-visible infrared light which can be hazardous to human eyes. Products which incorporate these devices must follow the safety precautions given in IEC 60825-1 and IEC 62471.

Chapter 2
Recommendations

2.1 Recommended computer configuration

The SLN-VIZN3D-IOT SDK requires an up-to-date computer which runs MCUXpresso IDE version 11.4 or higher. It also requires a terminal program to communicate with the device via USB.

Table 1. Supported computer configurations

Computer type	OS version	Terminal program
Apple	Mac OS	PuTTY
PC	Windows 7/10	PuTTY/Tera Term
PC	Linux	PuTTY

2.2 Recommended phone configuration

The SLN-VIZN3D-IOT kit requires an Android™ phone App to demonstrate the remote registration.

Our engineer team has verified and tested the Smart Lock App for Android using terminals running Android 8 or higher.

2.3 Usage condition

The following information is provided per Article 10.8 of the Radio Equipment Directive 2014/53/EU:

1. Frequency bands in which the equipment operates.
2. The maximum RF power transmitted.

Table 2. Bluetooth frequency and power

PN	RF technology	Frequency range	Maximum transmitted power
SLN-VIZN3D-IOT	Bluetooth	2402 - 2480 MHz	10 dBm

EUROPEAN DECLARATION OF CONFORMITY (Simplified DoC per Article 10.9 of the Radio Equipment Directive 2014/53/EU)

This apparatus, namely SLN-VIZN3D-IOT, conforms to the Radio Equipment Directive 2014/53/EU. The full EU Declaration of Conformity for this apparatus can be found at this location: http://www.nxp.com/mcu-vision3d

Chapter 3
Getting started with SLN-VIZN3D-IOT

3.1 Box contents

The SLN-VIZN3D-IOT kit is delivered fully assembled in a box with a printed quick start card and a USB-C cable.

Figure 3. SLN-VIZN3D-IOT box contents

Check your kit for damage or marks, and if you find any, contact your NXP representative.

3.2 Running the preconfigured smart-lock application

Before using your kit, remove the protection films on the 3D camera and the RGB camera (see Figure 4), which protect the lens during transport. Otherwise, the depth and image capture will be wrong.

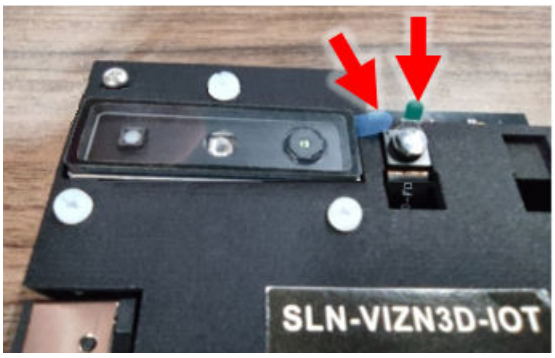

Figure 4. Removing camera protection films

To get started, take the USB-C cable provided in the kit and plug the USB-A end into your computer's USB port and the USB-C end into the kit's USB port.

Figure 5. Powering the SLN-VIZN3D-IOT kit

When connected, the RGB camera video appears, framed with some application settings.

Figure 6. Boot phase completed

Chapter 4
Local face management

To evaluate the smart-lock application pre-programmed on the SLN-VIZN3D-IOT kit, you must first register a face.

4.1 Registering a face

The onboard pushbuttons are pre-configured to allow direct access to some of the kit's most useful features, like face registering and de-registering and entering and exiting Low Power Mode. To manually register a new face, press and release the SW3 button located at the back of the kit.

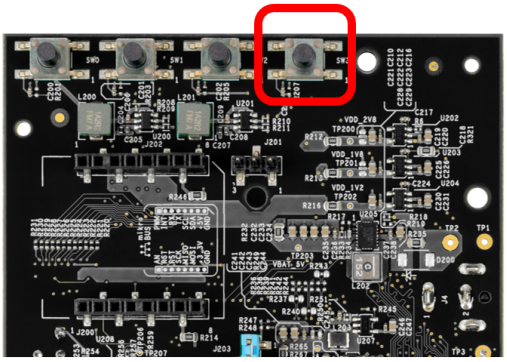

Figure 7. Manual registration button

The speaker plays an audio message, confirming that the registration started. During the face registration process, the "Registering" message appears in green at the top of the screen and a green bounding box appears to help align the user's face. Additionally, a timeout counter represented by a blue bar located at the bottom of the screen appears automatically and starts to count down.

To register your face, look straight-on to the camera while aligning your face inside the bounding box guidelines presented by the UI.

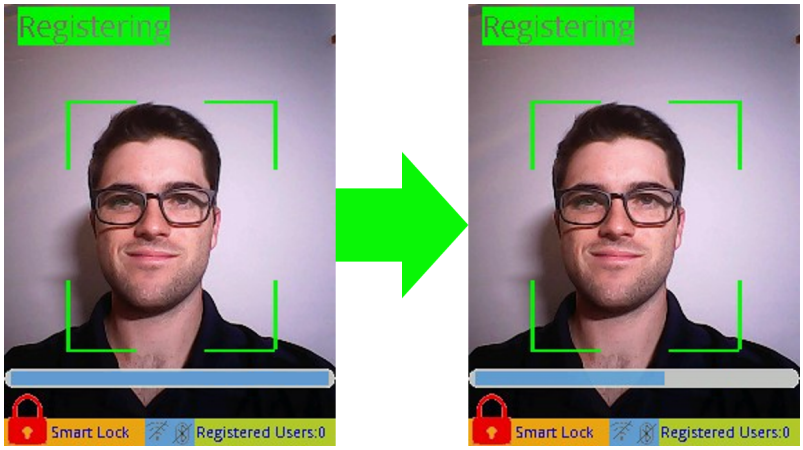

Figure 8. Registration in progress

To cancel the registration, press and release the SW3 button again or wait for the registration process to time out (~5 s).

When successfully registered, the speaker plays an audio message, confirming that the registration succeeded. Two messages (“Registration Successful” and “USERNAME added”) appear in green. The number of registered users increases by one and the user name shown on the screen is automatically assigned to the newly registered face.

Figure 9. Successful registration

NOTE

New faces can be added and usernames can be modified through the command-line interface or the phone app described in the following sections.

If your face fails to register due to the registration being canceled, timing out, or due to an attempt to register the same face twice, the speaker plays an error message and the screen displays a message in red, as shown in Figure 10.

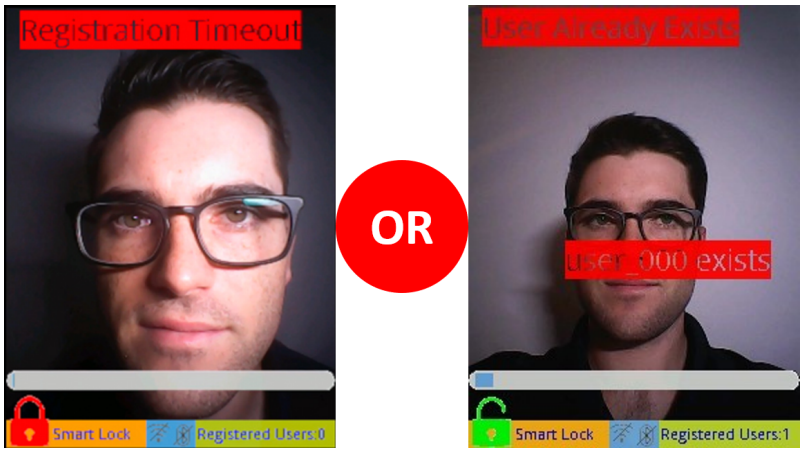

Figure 10. Registration failed

To retry, press and release the SW3 button again.

NOTE

If you have trouble successfully registering a face, try adjusting the proximity to the camera (either closer or further away) or move your head left-right and up-down so that the camera can get a better view of different angles of your face. For more information, see Troubleshooting.

When a face is registered, the 3D Smart Lock solution can automatically recognize your face. When a face is recognized, the screen displays the “Recognition Successful” message in green, the lock icon in the bottom left corner turns from red/locked to green/unlocked, and the speaker plays an audio message, confirming that a face is successfully recognized.

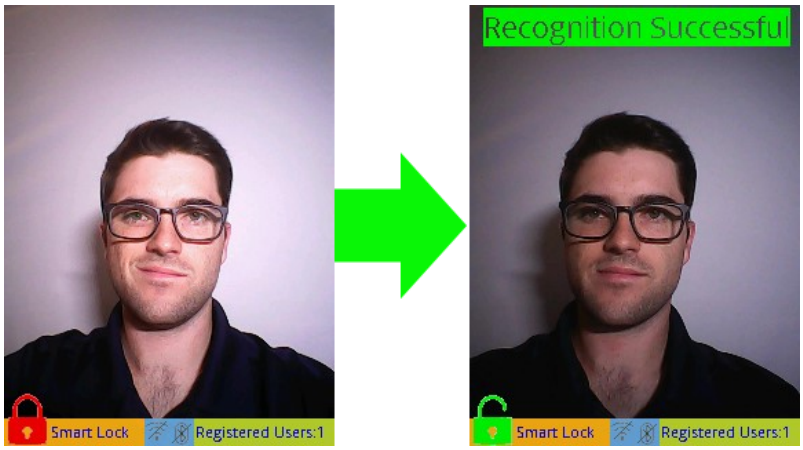

Figure 11. Successful recognition

Follow the same procedure to add additional faces into the system.

4.2 Deregistering a face

Similarly to the registration button discussed in the previous section, pressing and releasing the SW2 button (located at the back of the kit) triggers a manual de-registration, which removes the next recognized face from the kit's internal face database.

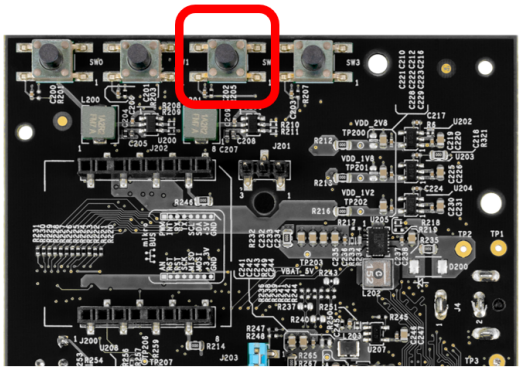

Figure 12. Manual deregistration button

The speaker plays an audio message, confirming that the de-registration started. During the face de-registration process, the "Deregistering" message appears in green at the top of the screen and a red bounding box appears to help you align your face. A timeout counter represented by a blue bar starts to count down.

To de-register your face, look straight-on to the camera and align your face in the bounding box presented by the UI.

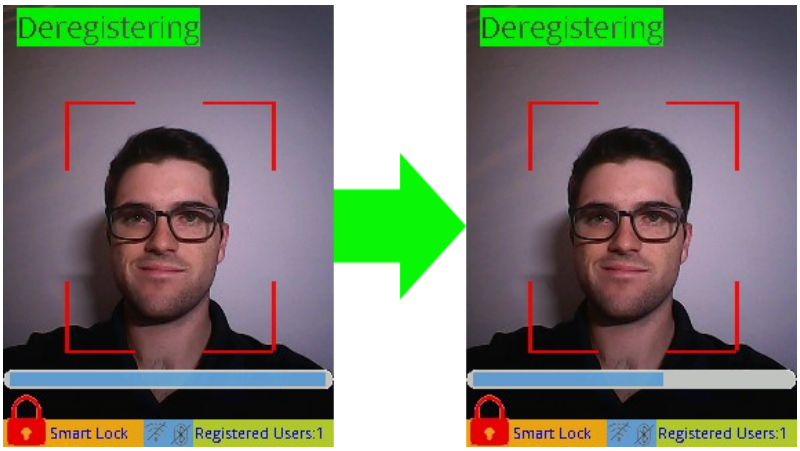

Figure 13. De-registration in progress

To cancel the de-registration process, press and release the SW2 button or wait for the registration process to time out (~10 s). When successfully de-registered, the speaker plays an audio message, confirming that the de-registration succeeded. An onscreen message confirms the user name of the user that was removed and the number of registered users is decremented by one.

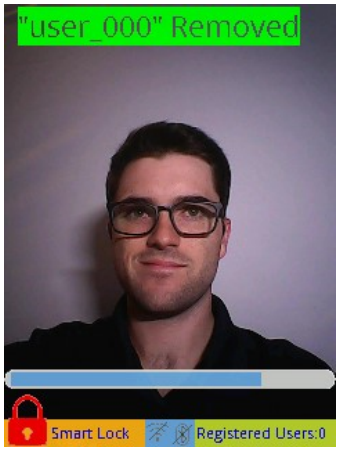

Figure 14. Successful de-registration

NOTE

Faces can also be deleted through the command-line interface or the phone app described in the following sections.

If your face fails to de-register due to the de-registration being canceled or timing out, the speaker plays an error message and the screen displays a message in red, as shown in Figure 15.

Figure 15. De-registration failed

To retry, press and release the SW2 button again.

NOTE

If you have trouble successfully de-registering a face, adjust the proximity to the camera (either closer or further away) and slowly move your head left-right and up-down so that the camera can get a better view of different angles of your face. For more information, see Troubleshooting.

4.3 Liveness detection

The 3D Smart Lock application preprogrammed in the SLN-VIZN3D-IOT kit comes with the liveness detection activated by default to make the most of the security features of the 3D camera. This solution can discern between an actual face and a printout, phone display picture, or 3D cast-mold of a face. This level of protection is crucial in applications that require security to prevent “face-spoofing” attacks.

As shown in the screenshot below, attempting to display a fake face to the camera results in a “Fake Face” UI message and audio prompt.

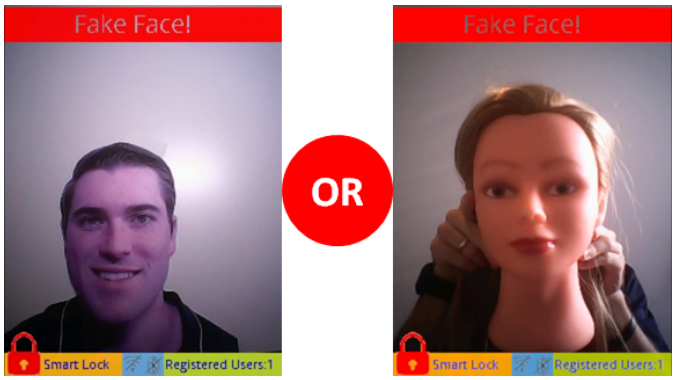

Figure 16. Printed picture (2D) and mask (3D) spoof attacks

By requiring a user’s actual face to unlock the system (as opposed to a simple picture of the user’s face), this feature helps to protect you from some of the most common face-recognition “spoof” attacks, where a malicious actor uses a picture of someone to gain access to their face-protected materials.

Using the combination of an IR camera and a 3D camera, outdoor environments with high and low lighting conditions do not impact the performance of the face-recognition algorithm.

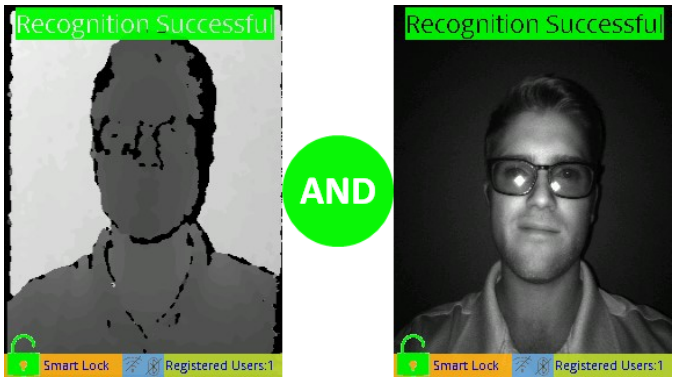

Figure 17. 3D camera depth and IR views

Chapter 5
Remote face management

The 3D Smart Lock application preprogrammed in the SLN-VIZN3D-IOT kit allows you to register and de-register faces locally using the onboard pushbuttons or serial commands. It also supports remote face management over Bluetooth Low-Energy (BLE) using an Android phone/tablet application. The Smart Lock Manager app for Android smartphones and tablets offers a friendly interface where you can both register new faces remotely and manage faces that are already registered in the SLN-VIZN3D-IOT's local face database. This section describes some major features provided by this dedicated Android app.

5.1 Installing the Smart Lock Android app manually

The pre-compiled APK and the complete source code for the Smart Lock Manager app are on the NXP website at the SLN-VIZN3D-IOT product page under the Software and Tools section.

To install the Smart Lock Manager app on your Android device, store the "Smart_Lock_Manager.apk" file on your smart device by downloading it directly from your device's web browser or by transferring it from your computer to your device. Before you can install it on your Android device, make sure that third-party apps are allowed on your device. Go to "Menu > Settings > Security" and tick "Unknown Sources" to allow your phone to install apps from sources other than the Google Play Store. As an alternative, you are prompted to allow your browser or file manager to install APKs the first time you attempt to do so. Open the Files App and search for the APK file location on your Android device (the "Downloads" folder by default), tap it, tap "Install" and wait for the installation to complete.

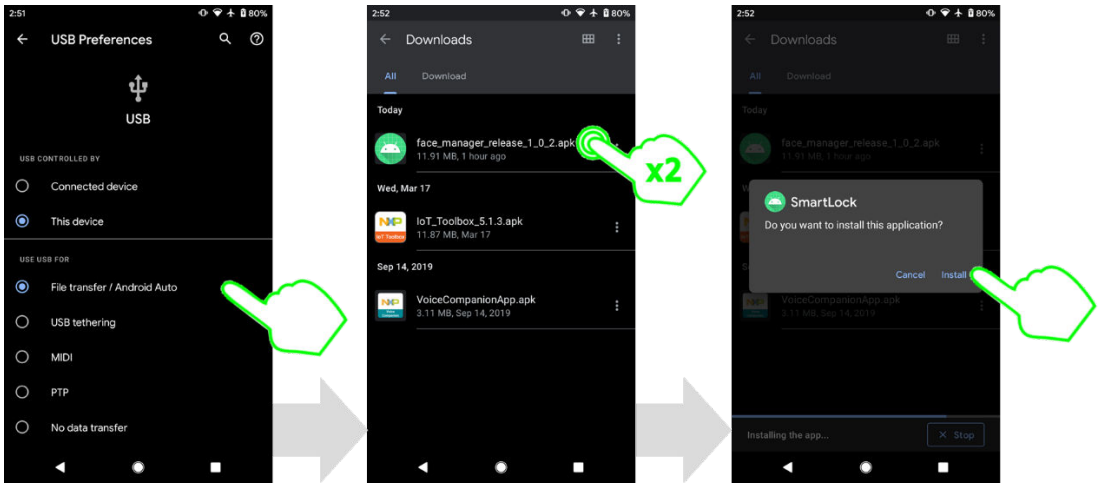

Figure 18. Installing the Smart Lock Manager app for Android

After opening the Smart Lock Manager app for the first time, you are asked to authorize the app to access your smart device's camera, your location (to enable Bluetooth), and your photographs. It is important that you approve those services to connect new kits or register new faces.

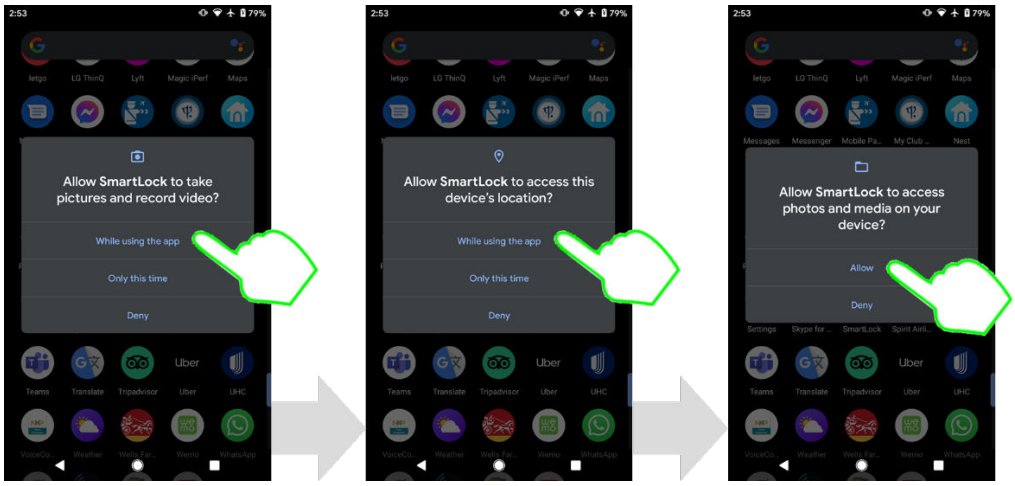

Figure 19. Allowing camera, location, and photograph access

If the Bluetooth radio is disabled on your Android device, the app asks you to turn it on. Bluetooth connectivity is required to pair your kit with the Smart Lock Manager app and remotely manage the registered faces stored in the SLN-VIZN3D-IOT's local database.

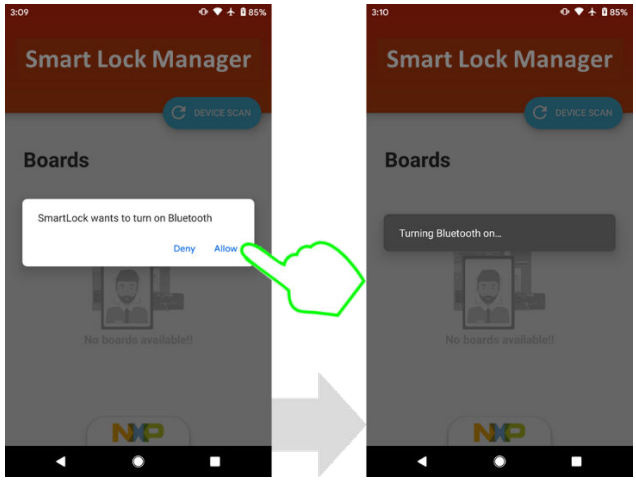

Figure 20. Enabling Bluetooth

NOTE

The Smart Lock Manager app is intended for use as an evaluation tool and as a reference/boiler plate upon which a customized smartphone/tablet companion application can be built.

5.2 Managing kits

To interact with your SLN-VIZN3D-IOT kit, pair your kit with the Smart Lock Manager app. This can be done from the app's main menu. Press the "Device Scan" button to detect the VIZN3D devices that are within the Bluetooth range of your Android device.

NOTE

The BLE feature of the SLN-VIZN3D-IOT is turned on by default, so the kit is automatically discovered by the app shortly after powering up.

While the search is in progress, any VIZN3D kits detected are displayed in the "Device" list. Select the device you want to add to your list of supported smart locks.

NOTE

Each kit broadcasts a unique Bluetooth SSID, starting with VN combined with its serial number to be easily identified.

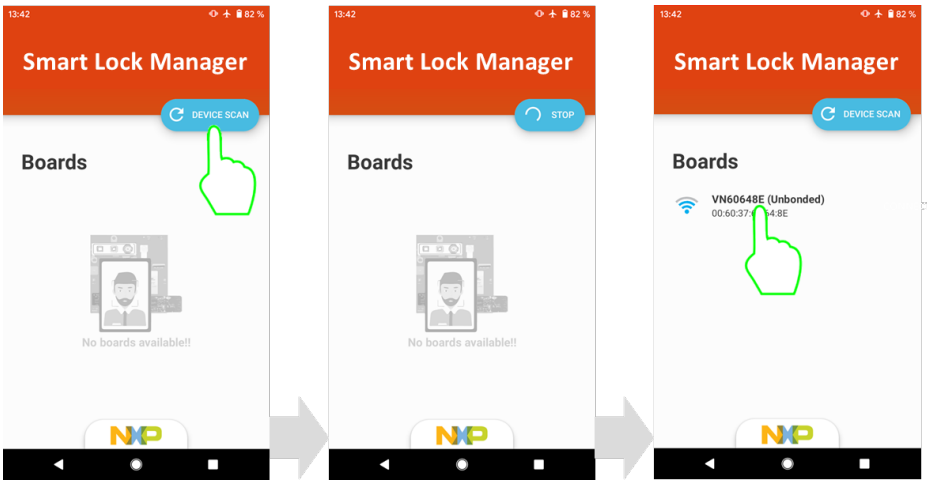

Figure 21. Pairing with a new smart lock

Before accessing your device, enter a password to secure the connection between your Android device and the SLN-VIZN3D-IOT kit you want to access. By default, this password is "000000".

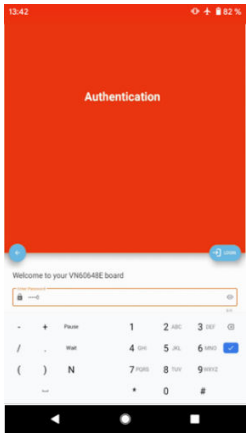

Figure 22. Smart lock password screen

When (securely) connected with the VIZN3D device, the Android app displays a list of users currently registered in the kit's local face database.

NOTE

Only one device can be connected and controlled at a time.

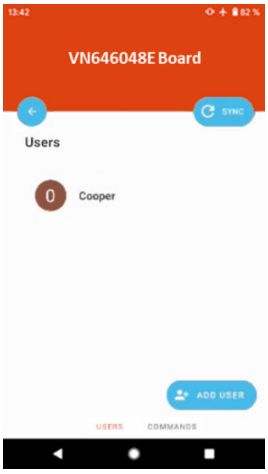

Figure 23. Smart lock list of users currently registered

By selecting the “Change Password” command, you can modify the password previously registered to secure the Bluetooth connection with the smart lock device. Enter the old password and the new password and click “Change”. The app automatically returns to the previous "Smart Lock Control" menu. A message is displayed in green at the bottom of the screen to confirm that the new password is successfully pushed to the device.

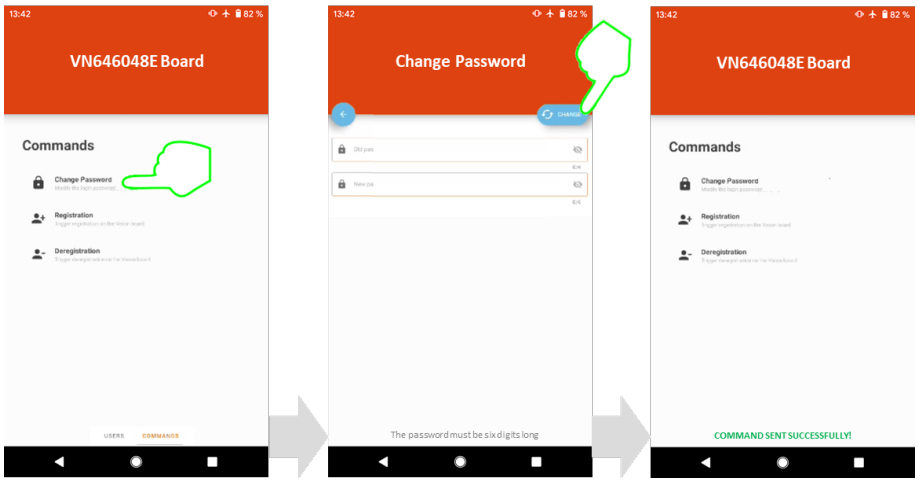

Figure 24. Changing smart lock password

5.3 Managing users

The Smart Lock Manager app can remotely manage the users registered to the SLN-VIZN3D-IOT kit using an Android device. This includes adding, listing, and deleting users from the SLN-VIZN3D-IOT’s local face database using 2-way synchronization. This section describes each of these features.

5.3.1 Adding users

One of the key features of the Smart Lock Manager app is registering new faces remotely from any Android device and uploading them to the VIZN3D device via Bluetooth. To start the remote registration process, tap the “Add User” button, as shown in Figure 25.

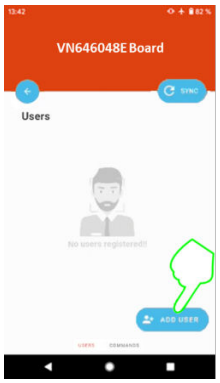

Figure 25. Remote registration menu

The Smart Lock Manager app automatically displays the live view of the front camera of your Android device. A green bounding box appears when your face is detected. To register your face, click the shutter icon at the bottom of the camera view (as shown in Figure 26) while the box around your face is green.

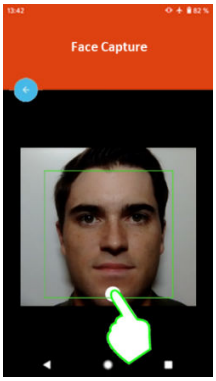

Figure 26. Remote face-capture screen

The app asks you to type a name of the new face. Tap the blue checkmark icon and select “Register” to finalize the registration.

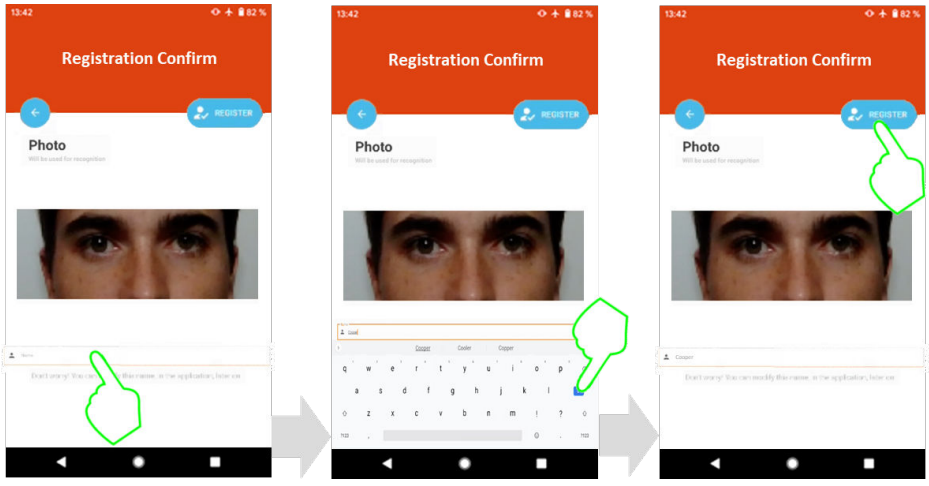

Figure 27. Adding name of the new face

The app automatically returns to the list of registered users for the selected smart lock device, which now includes the new face that was just added via the Android device.

To trigger a local registration remotely, go to the “Commands” menu and select the "Registration" option. This acts the same as if you press the local registration button (SW3) on the kit itself.

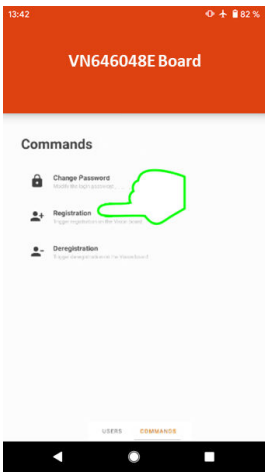

Figure 28. Trigger a local registration remotely

NOTE

If you locally register, update, or delete a face using the smart lock device, select the “Sync” option in the app to make the Android device aware of any changes to users that happened locally on the kit. A sync is automatically performed when you connect to a board.

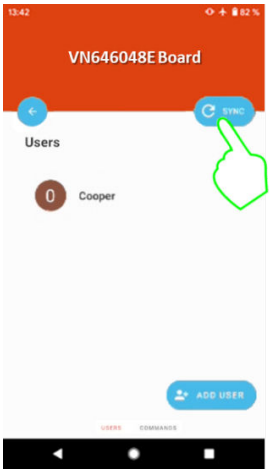

Figure 29. "Sync" option to update the face databases in the Android app and the kit

5.3.2 Modifying users

The Smart Lock Manager app also enables you to modify registered users. From the list of the Smart Lock registered users, select the user that you want to modify. The available options are "Delete User", "Update Name", and "Reregister".

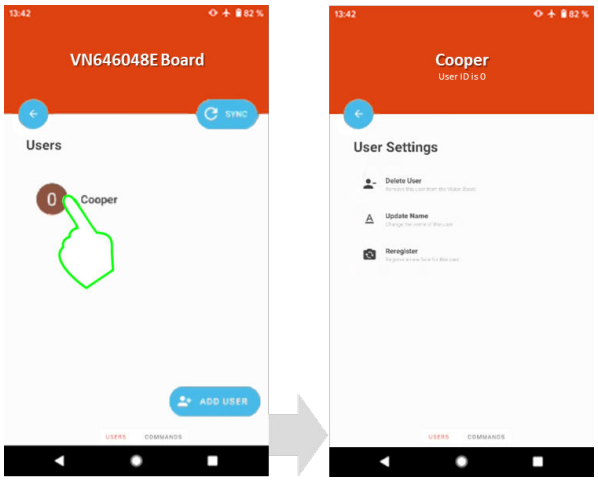

Figure 30. Managing registered users

To modify a user's name, choose "Update Name", type the new user name, and select "Update" to save the information.

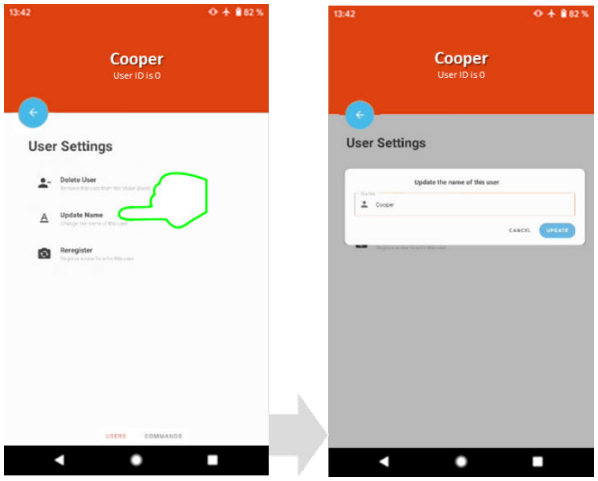

Figure 31. Renaming a registered user

NOTE

If you locally register, update, or delete a face using the Smart Lock device, select the "Sync" option in the app to make the Android device aware of any changes to users that happened locally on the kit. A sync is automatically performed when you connect to a board.

5.3.3 Deleting users

In addition to adding and modifying registered users, the Smart Lock Manager app also allows you to delete registered users as well. From the list of the Smart Lock registered users, select the user that you want to delete. To delete the user, select the "Delete User" option and confirm by selecting "Delete" to remove the user.

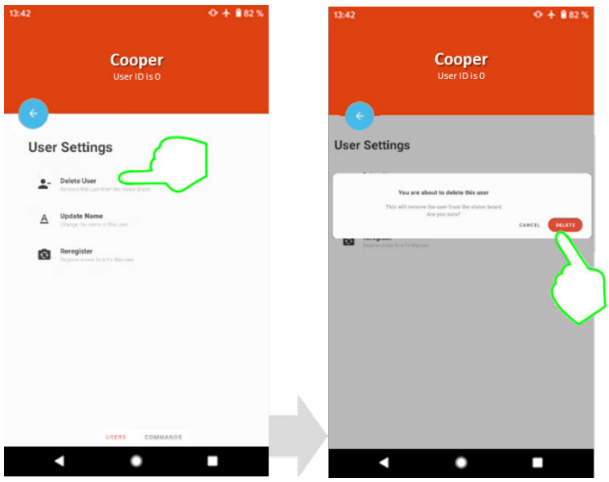

Figure 32. Remote deletion screen

To trigger a local deregistration remotely, navigate to the “Commands” menu and select the “Deregister” option. This works identically to pressing the deregistration button (SW2) on the board itself.

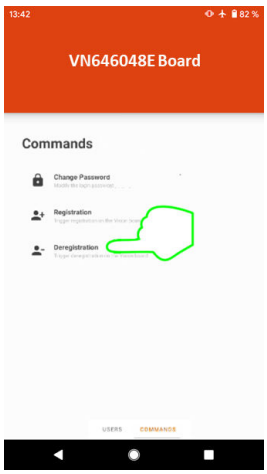

Figure 33. Triggering a local registration remotely

NOTE

If you locally register, update, or delete a face using the Smart Lock device, select the “Sync” option in the app to make the Android device aware of any changes to users that happened locally on the kit. A sync is automatically performed when you connect to a board.

Chapter 6
Additional features

6.1 Low-power mode

For many use cases, such as battery-based smart locks, managing power consumption is critical. Designed with these use cases in mind, the Smart Lock application preprogrammed on the SLN-VIZN3D-IOT kit supports low-power features out-of-the-box.

6.1.1 Pushbuttons

A user can manually set the SLN-VIZN3D-IOT kit to low-power mode by pressing and holding the SW1 pushbutton located on the back of the kit for >2 s and then releasing it.

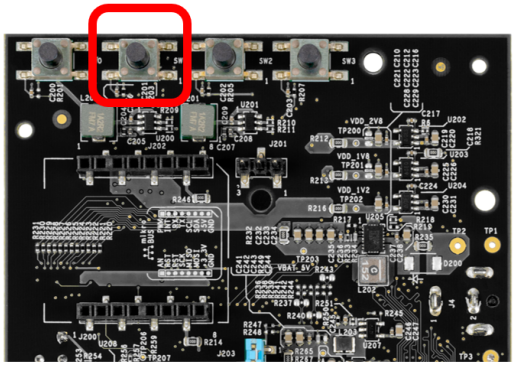

Figure 34. Start low-power mode (SW1 long press)

The screen displays a message and the speaker plays an audio message to confirm that the board is entering sleep mode before the display turns off.

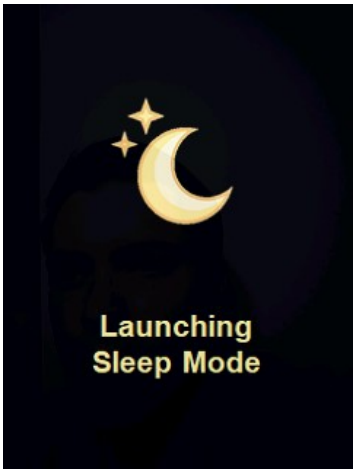

Figure 35. Entering low-power mode

During low-power mode, the RT117F MCU is set into the SNVS power state -- one of the lowest power modes supported by the processor -- and almost all onboard components (Flash, RAM…) and external modules (camera, display…) are powered off to conserve energy. As a result, the USB Serial Interface from the kit will not be active and will not be accessible until the board has

woken up again. Because only select peripherals are still active during sleep mode, the system will only wake-up in response to specific trigger(s).

By default, pressing and releasing the SW0 pushbutton located on the back of the kit will reawaken the kit.

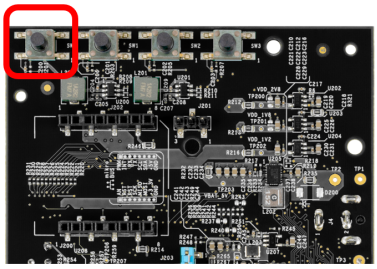

Figure 36. Wake-up from low-power mode (SW0 press)

The screen will immediately power back on once the button is pressed and released. Once the RGB camera feed begins showing up on the display, the 3D Smart Lock application is ready to recognize faces once again.

6.1.2 PIR sensor

The DIP Switch SW8 is located near the display and controls the wakeup source(s) for the processor. By default, it is configured to respond to the SW0 pushbutton and the QN9090 BLE (DIP switch configuration 1001).

CAUTION

Some combinations may generate unwanted behavior, refer to the Hardware Guide to identify the supported wake-up configurations.

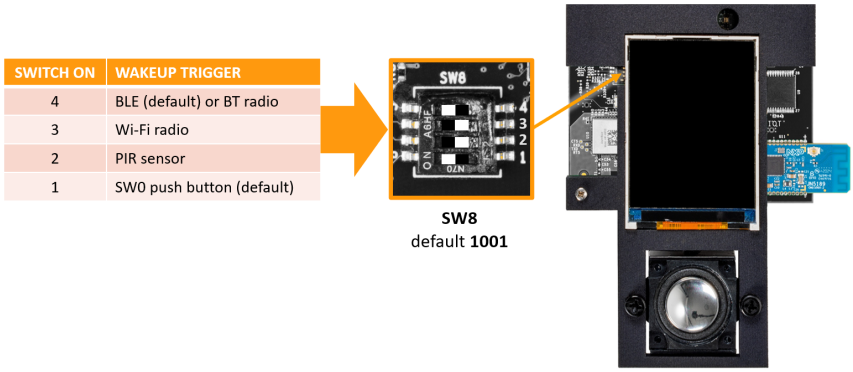

Figure 37. DIP switch SW8 to select the wake-up source

The SLN-VIZN3D-IOT kit also embeds a PIR sensor capable of waking the system when motion is sensed. To activate the PIR sensor (and disable the SW0 pushbutton and QN9090 BLE wake-up sources), change the configuration of the SW8 DIP Switch from 1001 to 0100.

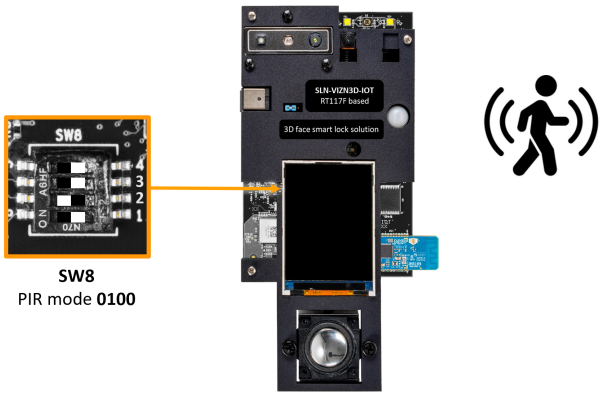

Figure 38. SW8 DIP switch configuration with PIR sensor activated

To test the motion activated wake-up, manually set the system into sleep mode by pressing and holding the SW1 pushbutton located on the back of the kit for >2s before releasing it. Once the board has entered sleep mode, any motion in front of the PIR sensor will automatically wake-up the system.

6.1.3 Automatically enable low-power mode

In addition to manually activating the low-power mode setting via the onboard pushbuttons, the SLN-VIZN3D-IOT can also be configured to automatically enter sleep mode. When automatic low-power mode is enabled, the kit uses a combination of timers and other event triggers to determine when it is okay to enter sleep. Configuring automatic sleep mode is handled via the “lpm enable/disable” shell command. See Configure automatic low-power mode for more details.

While automatic sleep mode is enabled, any of the following events causes the SLN-VIZN3D-IOT to enter sleep mode:

- A face is recognized
- 10s have passed with no face detected
- 25s have passed with no recognizable face detected

However, there are states in which the above triggers will not activate low-power mode and those states include:

- A BLE connection between the kit and the Smart Lock Manager app is currently active
- A registration/deregistration is taking place

In either of the above scenarios, entering low-power mode is deferred until after the event has completed to prevent the kit from falling asleep during processes that may take longer than the sleep timers allow for. Additionally, in the event of a registration/deregistration, the sleep mode timers reset after the registration/de-registration is complete.

As an example of how the automatic sleep mode process might work, consider the following scenarios:

The SLN-VIZN3D-IOT is power cycled and has previously been configured with automatic sleep mode enabled. After 2 seconds have passed with no face detected by the kit, a face is detected. After 1 more second, that face is recognized. The kit enters sleep mode.

The SLN-VIZN3D-IOT is power cycled and has previously been configured with automatic sleep mode enabled. After 9 seconds have passed with no face detected by the kit, an unrecognized face is detected. 10 more seconds pass with the face still unrecognized. A connection to the Smart Lock Manager remote registration app is established. A new user is remotely registered and the Smart Lock Manager app is closed after 30 seconds of use. As soon as the app is closed, the BLE connection between the app and kit has been severed and the kit enters sleep mode because the 25 second “face detected” timer completed while the remote registration app was in use.

The SLN-VIZN3D-IOT is power cycled and has previously been configured with automatic sleep mode enabled. After 9 seconds have passed with no face detected by the kit, a face is detected. After 15 seconds, a registration is started using the SW3

pushbutton and completes after 2 seconds. After registration completes, no face is detected, and the kit falls asleep after 10 seconds.

While low-power mode's automatic configuration is active and the kit is about to enter sleep, a "Recognition Failed" message like the one shown in the below figure is displayed on the screen as a warning that the SLN-VIZN3D-IOT is entering sleep mode within the next 1.5 seconds.

Figure 39. "Recognition timeout" message

Should 1.5 seconds pass without any events which disrupt low-power mode's activation taking place, the regular low-power mode splash screen is displayed and the kit goes to sleep.

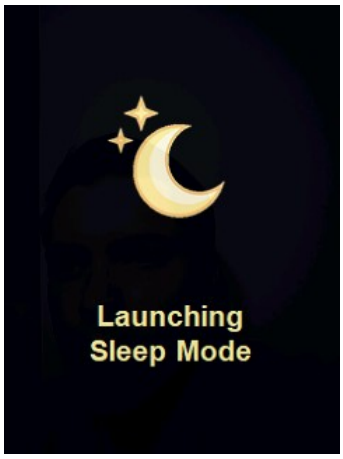

Figure 40. "Enter Sleep" Splash Screen

Once asleep, the board can only be awoken by the triggers mentioned in Figure 41: USB Power (power cycling the kit), the reset switch (SW4), SW0 switch, the PIR sensor, and the QN9090 Bluetooth module. Triggering a wake-up causes the board to return to normal operation.

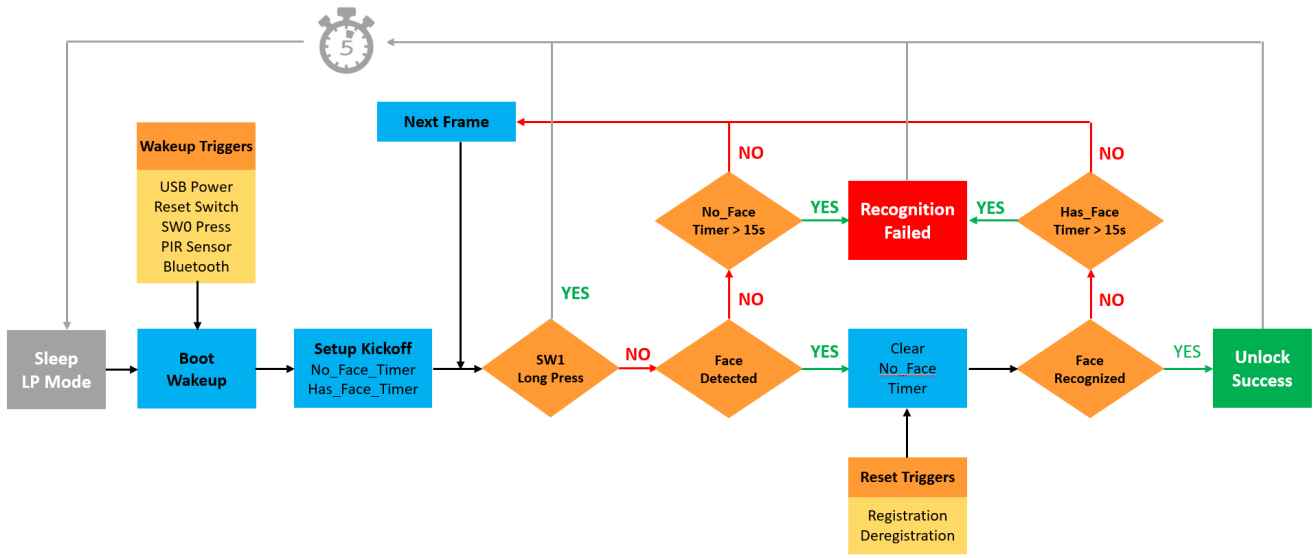

Figure 41. Low power diagram

6.2 Serial command line interface

6.2.1 Connect to the serial command line interface

The SLN-VIZN3D-IOT kit supports several serial commands which give users access to the full suite of features the preprogrammed Smart Lock application has to offer. Serial commands can be issued via the kit's built-in CLI, accessible using a serial terminal emulator like PuTTY or Tera Term.

To connect to the kit's serial-based CLI, identify the **COM** port associated with the kit and connect using the [115200, 8, 1, N, XON/XOFF] serial settings shown below.

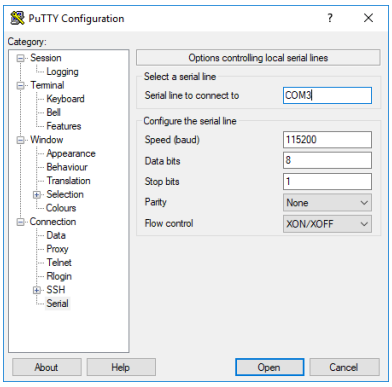

Figure 42. Serial connection settings

After connecting to the kit's serial interface, you will encounter a blank terminal screen which will echo back any characters that you type.

Use the "**help**" command to display a list of all the available serial commands and their usage. We will be discussing a few of these commands in the upcoming sections.

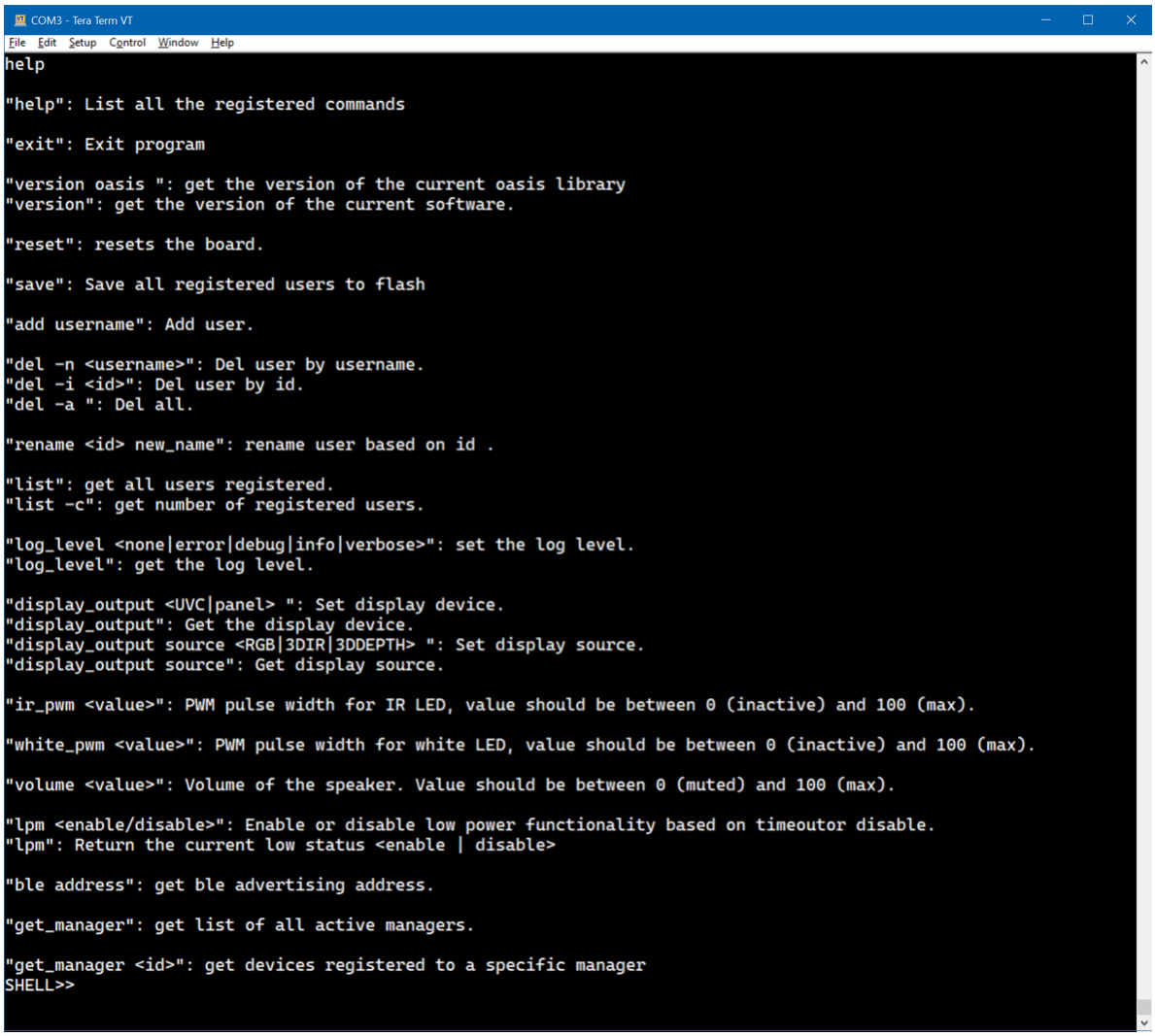

Figure 43. Serial commands "Help" menu

6.2.2 List available commands

To display a list of all available commands you can run, type the command "**help**." This command will output a list similar to that shown below which details all available commands and their arguments:

Table 3. Serial command table

Command	Arguments	Description
help		*List all the registered commands*
exit		*Exit program; closes serial terminal until reset*
version version oasis		*Get the version of the current application software* *Get the version of the current oasis library*

Table continues on the next page...

Table 3. Serial command table (continued)

reset		*Resets the board*
save		*Save all registered users to Flash Memory (this is done by default when users are added)*
add username	‹username›	*Add user*
del -n del -i del -a	‹username› ‹id› n/a	*Deletes user by username* *Deletes user by id* *Deletes all users*
rename	‹id› new_name	*Renames user based on id*
list list -c		*Get all registered users* *Get number of registered users*
log_level log_level	‹none | error | debug | info | verbose›	*Set the log level* *Get the current log level*
display_output display_output	‹UVC | panel›	*Set display device* *Get the current display device*
display_output source display_output source	‹RGB | 3DIR | 3DDEPTH›	*Set camera source* *Get the current camera source*
ir_pwm	‹value›	*Set IR LED power. Value between 0 (off) and 100*
white_pwm	‹value›	*Set White LED power. Value between 0 (off) and 100*
volume	‹value›	*Set Speaker volume. Value between 0 (off) and 100*
lpm lpm	‹enable | disable›	*Set Low Power Timer* *Get the current Low Power Timer status*
ble address		*Get BLE advertising address*
get_manager get_manager	‹id›	*Get list of all the running managers* *Get HAL devices registered to a specific manager*

6.3 Manage user database

6.3.1 List all registered users

Using the "**list**" command will list all the users currently registered in the system's local face database, whether they have been saved to flash, and their internal ID number, while the "**list -c**" command will display the total number of registered users.

```
SHELL>> list
Saved - Id 0 Name - user_000
Saved - Id 1 Name - Cooper
```

```
SHELL>> list -c
Users registered 2
```

Figure 44. List registered user names or count

6.3.2 Manually add users

To register a new face in manual mode, simply use the command "**add <username>**." This command acts nearly identically to manual registration using the SW3 pushbutton, but allows for the addition of a specific user name instead of a generic user name.

Additionally, using the "**add**" command with no arguments functions identically to the manual registration process using the SW3 pushbutton.

```
SHELL>> add Cooper
SHELL>> add
```

Figure 45. Manually add new user with specific or generic user name

6.3.3 Manually delete users

To delete a previously registered user, use the "**del -n**" command followed by the user name, or "**del -i**" followed by the user ID to delete that specified user.

```
SHELL>> del -n Cooper
Delete was successful
SHELL>> del -i 0
Delete was successful
```

Figure 46. Manually delete user

Additionally, using the "**del**" command with no arguments functions identically to the manual deregistration process using the pushbuttons SW2.

```
SHELL>> del
```

Figure 47. Manually delete next recognized user

Alternatively, running the command "**del -a**" will erase every user previously registered.

```
SHELL>> del -a
All users successfully deleted
```

Figure 48. Manually delete every user

6.3.4 Rename users

The **“rename”** command can be used to rename a user already registered into the database. To do so, simply use the command **“rename user_ID newusername.”** Below is an example of the command being used.

```
SHELL>> rename 0 Cooper
Update was successful
```

Figure 49. Rename registered user

6.3.5 Manually save users into flash

Registered faces are automatically saved into flash to ensure user access after a board reset or power cycle. However, if the autosave feature has been disabled (refer to the software developer's guide), the "save" command can be used to manually store registered users into flash. Users are saved in the order that they were added and a success message is returned when the command finishes running.

```
SHELL>> save
Saving users to flash…
User list saved
```

Figure 50. Manually save all registered users into Flash memory

6.4 Display configuration

The SLN-VIZN3D-IOT kit comes pre-assembled with the Rocktech RK024HH298 VGA portrait 2.4 inch parallel TFT screen, but the graphical interface for the Smart Lock application can also be pushed over USB to a computer screen similar to a regular USB web camera device.

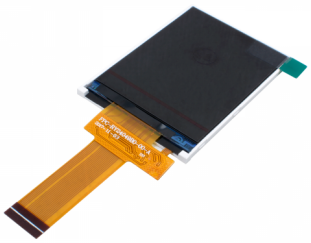

Figure 51. Rocktech RK024HH298

To activate the USB Video Class (UVC) and display the camera output with the graphical interface of the Smart Lock application on a computer screen, type the command **“display_output UVC”** in the serial terminal. After issuing the command, a reset of the board using either the “**reset**” command or physically power cycling the board is required for the change to take effect.

```
SHELL>> display_output UVC
Display output set. Reset the board for the change to take effect.
SHELL>> reset
```

Figure 52. Enable UVC video output command

To return to the factory configuration and (re)activate the screen included with the kit, simply type the command **“display_output panel.”**

```
SHELL>> display_output panel
Display output set. Reset the board for the change to take effect.
SHELL>> reset
```

Figure 53. Enable panel-based video output command

NOTE

The SLN-VIZN3D-IOT kits leverage the MIMXRT117F parallel display controller to control the supplied TFT screen with NXP driver/middleware available in the SLN-VIZN3D-IOT SDK package.

The VIZN3D board embeds parallel display connectors (J208, J209) compatible with the Rocktech RK043FN02H-CT WQVGA landscape 4.3inch TFT screen with capacitive touch. Low-level drivers for this display are available in the RT1060 EVK SDK package

The VIZN3D board also features a MIPI display connector (J206) to connect the Rocktech RK055HDMIPI4M HD720 landscape 5.5 inch TFT screen with capacitive touch. Low-level drivers for this display are available in the RT1170 EVK SDK package.

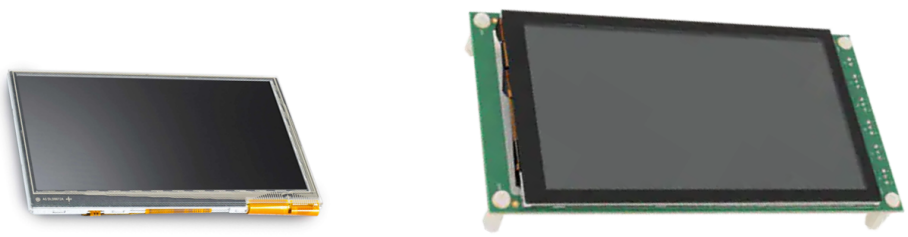

Figure 54. Rocktech RK043FN02H-CT (left) and RK055HDMIPI4M (right)

6.5 Camera configuration

The camera output displayed on the screen can be configured via the **“display_output source”** command choosing between RGB, 3DIR, and 3DDEPTH options.

```
SHELL>> display_output source RGB
Display output source set
SHELL>> display_output source 3DIR
Display output source set
SHELL>> display_output source 3DDEPTH
Display output source set
```

Figure 55. Three camera output commands

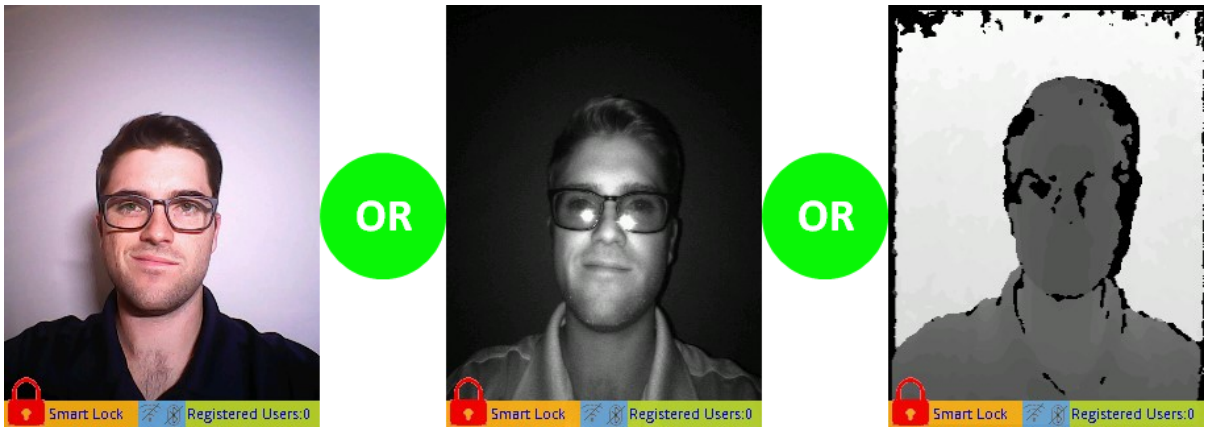

Figure 56. RGB, IR, and DEPTH camera output on screen

NOTE

This command will only change the camera output being shown, not the camera output(s) being used by the detection and the recognition algorithms.

6.6 Audio configuration

The Smart Lock application includes audio feedback prompts, which are used alongside the on-screen messages to enhance the user experience or emulate headless use-cases.

The SLN-VIZN3D-IOT kit comes pre-assembled with the PUI ASE02808MR-150-R enclosed 2W Speaker with audio circuitry controlled by the RT117F MQS interface.

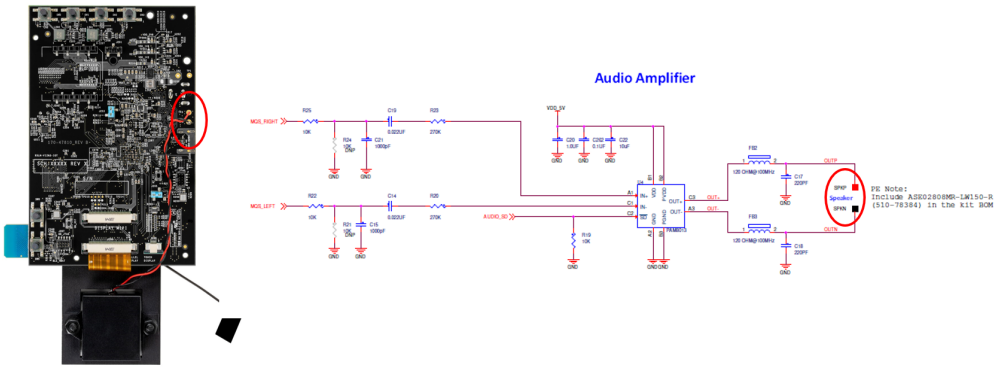

Figure 57. Speaker and audio circuitry

The full list of audio messages supported by the Smart Lock application is available in the table below.

Table 4. List of audio messages for the Smart Lock application

Voice ID	Audio Message	Activity
1	Starting Registration	*Registration*

Table continues on the next page...

Table 4. List of audio messages for the Smart Lock application (continued)

2	Registration Successful	*Registration*
3	Registration Failed	*Registration*
4	Look at Camera	*Registration*
5	Starting Deregistration	*De-Registration*
6	Deregistration Successful	*De-Registration*
7	Deregistration Failed	*De-Registration*
8	Recognition Successful	*Recognition and entering low-power mode*
9	Fake Face	*Recognition*
10	Recognition Failed	*Recognition and entering low-power mode*
11	Enter Sleep	*Entering low-power mode*
12	Bluetooth Connected	*Bluetooth connectivity*
13	Bluetooth Disconnected	*Bluetooth connectivity*

To change the speaker volume, type the **"volume" command** followed by a value between 0 and 100. To disable audio prompts entirely, simply enter the value 0 for the speaker volume.

```
SHELL>> volume 25
Volume set
SHELL>> volume
Current volume is 25
```

Figure 58. Speaker volume commands

6.7 Configure LED brightness

6.7.1 IR LED

The SLN-VIZN3D-IOT kit embeds an IR LED controlled by an LED driver which can be used to modulate its power according to the lighting conditions. The RT117F uses a PWM signal with a configurable duty cycle set by a serial command to control the LED driver.

To change the power emitted by the infrared LED, type the command **"ir_pwm"** followed by a value between 0 and 100.

CAUTION

The IR LED can be harmful for the human eye so we recommend not to exceed 35% power and never look at the IR LED directly.

```
SHELL>> ir_pwm 20
Brightness set
```

```
SHELL>> ir_pwm
IR LED Brightness is currently set to: 20
```

Figure 59. IR LED power commands

6.7.2 White LED

The VIZN3D board embeds two white LEDs controlled by an LED driver which can be used to modulate their power according to lighting conditions. The RT117F uses a PWM signal with a configurable duty cycle set by a serial command to control the LED driver.

To change the power emitted by the White LEDs, type the command **"white_pwm"** followed by a value between 0 and 100.

CAUTION

The White LEDs can be harmful for the human eye so we recommend not to exceed 35% power and never look at the White LEDs directly.

```
SHELL>> white_pwm 20
Brightness set
SHELL>> white_pwm
White LED Brightness is currently set to: 20
```

Figure 60. IR LED power command

6.8 Configure automatic low-power mode

The SLN-VIZN3D-IOT has been optimized for battery powered applications in part by supporting the ability to enter a low-power state either manually, using a pushbutton, or automatically using a combination of timers and other triggers. For more information regarding low-power mode, see Low-power mode.

To control the Lower Power Timer enter the Serial command **"lpm"** with the argument enable or disable to activate or deactivate it.

```
SHELL>> lpm enable
SHELL>> lpm disable
```

Figure 61. Control low-power timer command

To get the current Lower Power Timer mode, type the command **"lpm"** without any arguments.

```
SHELL>> lpm
lpm enable
```

Figure 62. Status low-power timer command

6.9 Debug logging messages

To help developers understand how our solution operates, the Smart Lock application preprogrammed on the kit offers five levels of log messages: **none**, **error**, **debug**, info, and **verbose**.

To get the current log level, type the serial command **"log_level"** without any arguments.

```
SHELL>> log_level
log_level none
```

Figure 63. Get log level command

Serial debug messages are disabled by default but can be enabled using the **"log_level <none|error|debug|info|verbose>"** serial command. Whichever mode is enabled, it also enables the receipt of all levels of message "above" the level specified. For example, the command "**log_level debug**" also enables the receipt of **error** level messages as well.

```
SHELL>> log_level debug
Log_level set
```

Figure 64. Set log level command

To receive debug logs, a USB to UART converter must be connected to the physical UART of the RT117F MCU accessible from the expansion header J202 located on the back of the kit. The MCU UART uses 3V3 signals so make sure that your probe is compatible/configured for this voltage. The figure below shows how to connect the UART probe.

CAUTION

Some USB-to-serial adapters already include an internal 3V3 source and do not require the 3V3 connection to the VIZN3D board. Check your adapter documentation to avoid malfunction.

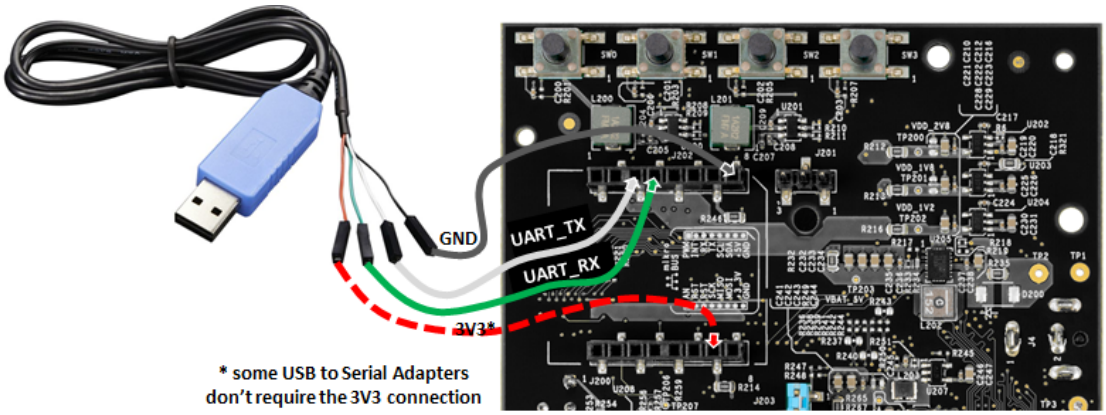

Figure 65. Get log level command

With the USB to UART probe connected, use a serial terminal emulator like PuTTY or Tera Term and connect to the COM port associated with the USB to UART convertor using the [115200, 8, 1, N, XON/XOFF] serial settings shown below:

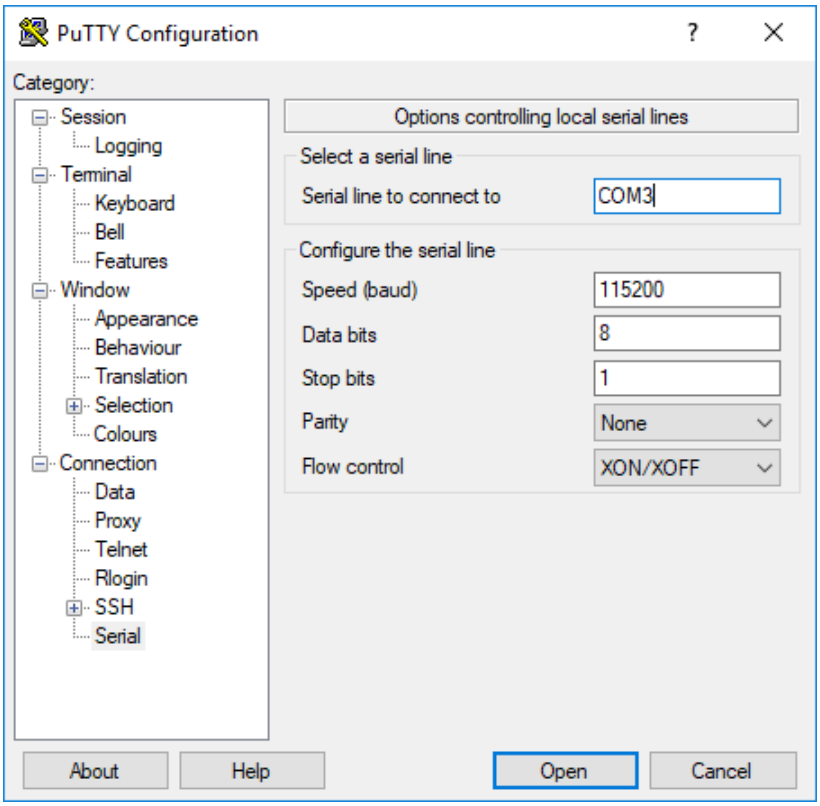

Figure 66. USB to UART serial settings

Error – Communicates error-related information. Includes messages related to the initialization routine for HAL devices and Framework Manager. Unique to error messages is the fact that, unlike other types of messages, they also includes the file and line number where the error message was printed to assist in more easily identifying where the error originated.

Example:

```
.main.c:88: In function "APP_InitFramework":
[ 0. 84][Error] - FWK_CameraManager_Init error 1
```

Figure 67. Error logs

Debug – Information useful for debugging the program. This includes messages regarding various managers, HAL devices, and the latest inference results from the Oasis Lite face recognition engine.

Example:

```
[ 0. 6][Debug] - [MAIN]:Started
[ 0. 77][Debug] - [LpmManager]:Starting...
[ 0. 93][Debug] - [LpmManager]:INIT lpm dev[0]
[ 0.339][Debug] - START lpm dev [0]
[ 0.384][Debug] - [LpmManager]:started 0
[ 0.401][Debug] - ++HAL_FlashDev_Littlefs_Init
[ 9.612][Debug] - Littlefs mount success
[ 9.641][Debug] - --HAL_FlashDev_Littlefs_Init
[ 9.756][Debug] - Littlefs directory exists: -17
[ 10.140][Debug] - HAL_GfxDev_Pxp_Register
[ 105. 10][Debug] - START camera dev[0]
```

```
[ 105. 61][Debug] - HAL_CameraDev_FlexioGc0308_Start(
[ 211. 18][Debug] - --HAL_InputDev_ShellUsb_start
[ 211. 52][Debug] - START input dev [2]
[ 211.116][Debug] - [BleWirelessUartTask] start.
[ 289.126][Debug] - --display_dev_init
[ 289.177][Debug] - [DisplayManager]:START dev[0]
[ 289.196][Debug] - ++display_dev_start
[ 289.216][Debug] - --display_dev_start
[ 308. 9][Debug] - START camera dev[1]
[ 308. 47][Debug] - HAL_CameraDev_CsiOrbbecU1s_Start
[ 543.324][Debug] - [OASIS]DET: - No face detected.
[ 561.105][Debug] - HEADER_PACKET:
[ 561.186][Debug] - HEADER PACKET PARSE SUCCESSFUL[27].
[ 561.267][Debug] - [BleWirelessUartTask]: BLE QN9090 start scan.
[ 561.307][Debug] - DATA_PACKET:
[ 561.331][Debug] - DATA PACKET PARSE SUCCESSFUL[27].
[ 627.380][Debug] - HEADER_PACKET:
[ 632.938][Debug] - HEADER PACKET PARSE SUCCESSFUL[27].
[ 633. 18][Debug] - [BleWirelessUartTask]: BLE QN9090 already start scan.
[ 633. 52][Debug] - DATA_PACKET:
[ 633. 72][Debug] - DATA PACKET PARSE SUCCESSFUL[27].
[ 744.816][Debug] - [OASIS]DET: - No face detected.
[ 940.486][Debug] - [OASIS]DET: - No face detected.
[ 1138.537][Debug] - [OASIS]DET: - No face detected.
```

Figure 68. Debug logs

Info – Messages that may be useful for developers to know, but generally not as important as debug messages. Includes messages for enqueueing and dequeueing of camera frames and more fine-grained Oasis-related messages.

Example:

```
[ 504940.528][Info] - [OASIS]INVALID_FACE.
[ 504991.520][Info] - --HAL_CameraDev_CsiOrbbecU1s_Dequeue
[ 505005.386][Info] - ++HAL_CameraDev_CsiOrbbecU1s_Enqueue
[ 505005.404][Info] - Submitting empty buffer
[ 505005.421][Info] - --HAL_CameraDev_CsiOrbbecU1s_Enqueue
```

Figure 69. Info logs

Verbose – The lowest level of log message. Mostly concerned with internal status of message passing between managers. Generally unnecessary to enable this level of logging.

Example:

```
[ 505205. 5][Verbose] - Task:[2] put message:[0x8038ADA4]:[5]
[ 505205. 46][Verbose] - Task:[2] put message:[0x8038ADA4]:[5] done
[ 505214.641][Info] - --HAL_CameraDev_CsiOrbbecU1s_Dequeue
[ 505217.338][Verbose] - Task:[0] put from isr message:[0x8038AC6C]:[0]
[ 505217.376][Verbose] - Task:[0] put from isr message:[0x8038AC6C]:[0] done
[ 505238.127][Verbose] - Task:[2] put message:[0x8038AD70]:[5]
[ 505238.175][Verbose] - Task:[2] put message:[0x8038AD70]:[5] done
[ 505238.199][Info] - ++HAL_CameraDev_CsiOrbbecU1s_Enqueue
[ 505238.221][Info] - Submitting empty buffer
[ 505238.237][Info] - --HAL_CameraDev_CsiOrbbecU1s_Enqueue
```

```
[ 505245.613][Verbose] - Task:[8038AC44]:[0]:[8038B640] Waiting to receive message
[ 505245.649][Verbose] - Task:[8038AECC]:[1]:[8038B6C0] Waiting to receive message
```

Figure 70. Verbose logs

In addition to the log messages shown above, setting “log_level” to anything other than “none” or “error” results in some useful on-screen information as well. This on-screen output reflects information about the face currently being recognized.

In the screenshot below, the face of user 0 has been recognized (hence "id:0") with 89.60 % similarity to its matching face in the kit’s internal database (hence "sim 89.60 %"), has passed the internal blurriness check (hence “blur: 0”), and most importantly both the 3D and 2D liveness checks (hence “3d_fake: 0” and “2d_fake: 0”). Finally, the face recognized below is also positioned correctly because it is facing the camera (hence “pose:front”).

Figure 71. On-screen info for recognized face

6.10 Get BLE address

Type the command **“ble address”** to easily display the Bluetooth ID from your kit. This identifier will be necessary to quickly identify your kit in a crowded environment to pair it with the Android Smart Lock Manager App.

```
SHELL>> ble address
BLE SSID [VN60648E]
```

Figure 72. Get BLE address command

6.11 Get managers and manager components running

To display a list of the Managers running on your solution, type the serial command “**get_manager**”.

```
SHELL>> get_manager
ID- 0 priority- 5 Name- camera_manager
ID- 1 priority- 4 Name- display_manager
ID- 2 priority- 3 Name- vision_algo_manager
ID- 4 priority- 3 Name- output_manager
ID- 5 priority- 6 Name- input_manager
```

Figure 73. Inference engine and oasis version commands

To display a list of the devices registered to a specific manager, type the command “**get_manager**” followed by the ID of the manager obtained previously.

```
SHELL>> get_manager 2
SHELL>> ManagerID-2. DeviceID 0 DeviceName - OASIS_3D
```

Figure 74. Inference engine and oasis version commands

More information on Framework managers and HAL devices can be found in the Software Developer’s Guide.

6.12 Print version information

To quickly identify the features supported by the application, it is necessary to know which version of the Smart Lock application is currently running on the kit. Use the **“version”** command to display the version of the application currently running on your kit. This command also prints the current application bank being used by the running application. This is useful when performing Mass Storage Device drag and drop flashing to determine which type of binary to provide to the kit. For more information on Mass Storage Device drag and drop flashing, see the SLN-VIZN3D-IOT Software Developer’s Guide.

```
SHELL>> version
App running in bankA
Version 1.1.0
```

Figure 75. "version" command

Alternatively, the **“version oasis”** command identifies the version of the Oasis Lite library running on your kit.

```
SHELL>> version oasis
Oasis version 1.23.0
```

Figure 76. “version oasis” command

6.13 Reset the SLN-VIZN3D-IOT

Some situations may require restarting the kit without physically power cycling the kit or pressing the reset button. This is possible using the **“reset”** command, which performs a soft reset of the kit, going through the entire boot process again.

```
SHELL>> reset
System will reset
```

Figure 77. Soft reset command

NOTE

After any reset of the SLN-VIZN3D-IOT kit, you may need to restart any computer application used to control or communicate with the board (for example, Serial Terminal for USB to Serial Commands, Camera application for UVC mode.).

6.14 Terminate the SLN-VIZN3D-IOT CLI

The command **“exit”** ends the Serial Terminal Program and terminates serial communication with the board until the next power cycle. There is generally no scenario that requires use of this command.

```
SHELL>> exit
```

Figure 78. Exit command

Chapter 7
Troubleshooting

7.1 Registering a face

This section describes the steps that can be taken to help debug potential points of failure when attempting to perform face registration and recognition. If after following these steps you still cannot identify the cause of the issue, head over to the section of the NXP forum dedicated to the SLN-VIZN3D-IOT kit to reach out for additional assistance.

7.1.1 Adjust face proximity and position

Because face recognition makes use of face “identifiers” to determine the face being looked at, it is important that these identifiers can be seen by both cameras. Oftentimes, an improper face angle and/or the proximity to the camera can cause registration to fail. Shown below are a few examples of improper usage.

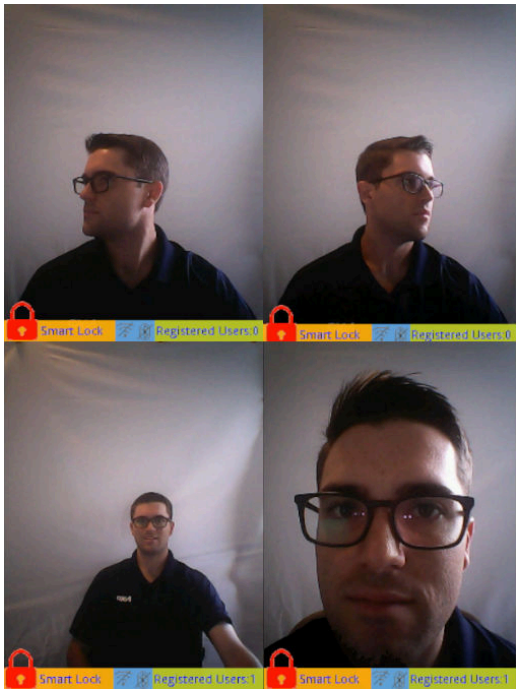

Figure 79. Improper face angle & proximity

When attempting to register a face, the face should ideally take up a majority of the bounding box, be centered inside the bounding box, and staring straight-on at the camera. To help ensure that the proper face angle is used, it can be helpful to slowly move the face from left to right and up to down.

The 3D module of the Orbbec camera is calibrated to properly scan a face between 0.3 m and 1.0 m. If the user's face is positioned closer or further away, the depth information will be invalid and the application will generate a "fake face" message.

7.1.2 Debug using log messages

Log messages can provide useful information when debugging registration issues, especially when it comes to debugging issues with liveness detection. Follow the steps in the Debug logging messages section of the guide for more information on how to activate logging messages. The on-screen indicators can be particularly helpful in identifying what is going wrong during registration.

Chapter 8
Document details

8.1 References

The following references are available to supplement this document:

Table 5. Reference documents

Document/Link	Remark
http://www.nxp.com/MCUXpresso	**MCUXpresso IDE Download**
https://www.nxp.com/docs/en/user-guide/MCUXpresso_IDE_User_Guide.pdf	**MCUXpresso IDE User Guide**
https://www.nxp.com/docs/en/user-guide/SLN-VIZN3D-IOT-DG.pdf	**SLN-VIZN3D-IOT Software Developer Guide**
https://www.nxp.com/mcu-vision3d	**SLN-VIZN3D-IOT Home Page**

8.2 Acronyms and definitions

Table 6. Acronyms and definitions

Acronym	Definition
FTDI	Future Technology Devices International
GUI	Graphic User Interface
IOT	Internet of Things
IVT	Instruction Vector Table
JTAG	Joint Test Action Group
MCU	Microcontroller Unit
MEMS	Micro-Electro-Mechanical System
MSD	Mass Storage Device
OEM	Original Equipment Manufacturer
OTW	Over the Wire
OTP	One Time Programmable
ROM	Read-Only Memory

Table continues on the next page...

Table 6. Acronyms and definitions (continued)

Acronym	Definition
RTOS	Real-Time Operating System
SDK	Software Development Kit
UART	Universal asynchronous receiver-transmitter

Chapter 9
Revision history

The following table lists the substantive changes done to this document since the initial release.

Table 7. Revision history

Revision number	Date	Substantive changes
0	01 November 2021	Initial release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Limited warranty and liability — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

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.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. 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, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX,EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE, VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. 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. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.

© NXP B.V. 2021. **All rights reserved.**

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

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 01 November 2021

Document identifier: SLN-VIZN3D-IOT-UG

arm