APPLICATION NOTE

Apollo3 Blue Family Typical Security Guidelines

Ultra-Low Power Apollo SoC Family

A-SOCAP3-ANGA06EN v1.0

Legal Information and Disclaimers

AMBIQ MICRO INTENDS FOR THE CONTENT CONTAINED IN THE DOCUMENT TO BE ACCURATE AND RELIABLE. THIS CONTENT MAY, HOWEVER, CONTAIN TECHNICAL INACCURACIES, TYPOGRAPHICAL ERRORS OR OTHER MISTAKES. AMBIQ MICRO MAY MAKE CORRECTIONS OR OTHER CHANGES TO THIS CONTENT AT ANY TIME. AMBIQ MICRO AND ITS SUPPLIERS RESERVE THE RIGHT TO MAKE CORRECTIONS, MODIFICATIONS, ENHANCEMENTS, IMPROVEMENTS AND OTHER CHANGES TO ITS PRODUCTS, PROGRAMS AND SERVICES AT ANY TIME OR TO DISCONTINUE ANY PRODUCTS, PROGRAMS, OR SERVICES WITHOUT NOTICE.

THE CONTENT IN THIS DOCUMENT IS PROVIDED "AS IS". AMBIQ MICRO AND ITS RESPECTIVE SUPPLIERS MAKE NO REPRESENTATIONS ABOUT THE SUITABILITY OF THIS CONTENT FOR ANY PURPOSE AND DISCLAIM ALL WARRANTIES AND CONDITIONS WITH REGARD TO THIS CONTENT, INCLUDING BUT NOT LIMITED TO, ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHT.

AMBIQ MICRO DOES NOT WARRANT OR REPRESENT THAT ANY LICENSE, EITHER EXPRESS OR IMPLIED, IS GRANTED UNDER ANY PATENT RIGHT, COPYRIGHT, MASK WORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT OF AMBIQ MICRO COVERING OR RELATING TO THIS CONTENT OR ANY COMBINATION, MACHINE, OR PROCESS TO WHICH THIS CONTENT RELATE OR WITH WHICH THIS CONTENT MAY BE USED.

USE OF THE INFORMATION IN THIS DOCUMENT MAY REQUIRE A LICENSE FROM A THIRD PARTY UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF THAT THIRD PARTY, OR A LICENSE FROM AMBIQ MICRO UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF AMBIQ MICRO.

INFORMATION IN THIS DOCUMENT IS PROVIDED SOLELY TO ENABLE SYSTEM AND SOFTWARE IMPLEMENTERS TO USE AMBIQ MICRO PRODUCTS. THERE ARE NO EXPRESS OR IMPLIED COPYRIGHT LICENSES GRANTED HEREUNDER TO DESIGN OR FABRICATE ANY INTEGRATED CIRCUITS OR INTEGRATED CIRCUITS BASED ON THE INFORMATION IN THIS DOCUMENT. AMBIQ MICRO RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN. AMBIQ MICRO MAKES NO WARRANTY, REPRESENTATION OR GUARANTEE REGARDING THE SUITABILITY OF ITS PRODUCTS FOR ANY PARTICULAR PURPOSE, NOR DOES AMBIQ MICRO 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 WHICH MAY BE PROVIDED IN AMBIQ MICRO 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. AMBIQ MICRO DOES NOT CONVEY ANY LICENSE UNDER NEITHER ITS PATENT RIGHTS NOR THE RIGHTS OF OTHERS. AMBIQ MICRO PRODUCTS ARE NOT DESIGNED, INTENDED, OR AUTHORIZED FOR USE AS COMPONENTS IN SYSTEMS INTENDED FOR SURGICAL IMPLANT INTO THE BODY, OR OTHER APPLICATIONS INTENDED TO SUPPORT OR SUSTAIN LIFE, OR FOR ANY OTHER APPLICATION IN WHICH THE FAILURE OF THE AMBIQ MICRO PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR. SHOULD BUYER PURCHASE OR USE AMBIQ MICRO PRODUCTS FOR ANY SUCH UNINTENDED OR UNAUTHORIZED APPLICATION, BUYER SHALL INDEMNIFY AND HOLD AMBIQ MICRO AND ITS OFFICERS, EMPLOYEES, SUBSIDIARIES, AFFILIATES, AND DISTRIBUTORS HARMLESS AGAINST ALL CLAIMS, COSTS, DAMAGES, AND EXPENSES, AND REASONABLE ATTORNEY FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PERSONAL INJURY OR DEATH ASSOCIATED WITH SUCH UNINTENDED OR UNAUTHORIZED USE, EVEN IF SUCH CLAIM ALLEGES THAT AMBIQ MICRO WAS NEGLIGENT REGARDING THE DESIGN OR MANUFACTURE OF THE PART.

Revision History

Revision	Date	Description
1.0	July 12, 2023	Initial release

Reference Documents

Document ID	Description
A-MCUA3B-GGNA02EN	Apollo3 Blue Security Getting Started Guide
A-SOCA3B-UGGA02EN	Apollo3 Blue Secure Update Flow Users Guide
A-SOCAP3-UGGA04EN	AMOTA Example User's Guide
A-SOCA3B-WPNA01EN	Apollo3 Blue Security White Paper

Table of Contents

SECTION

1 Introduction

Apollo3 Blue and Apollo3 Blue Plus parts from the Ambiq factory are pre-programmed with a Secure Boot Loader, and an uninitialized INFO0. This initial Life Cycle state is the Customer Manufacturing state. While in this state, the device can be locked/partially locked/open based on customer security policy to support additional development prior to entering the Production state.

This document is a supplement for the *Apollo3 Blue Getting Started Guide* and the *Apollo3 Blue Secure Update Flow Users Guide* and will provide a walkthrough for provisioning INFO0 and disabling/re-enabling debugger access.

1.1 Life Cycle State

The following life cycle states are defined for the Apollo3 Blue:

- Customer Manufacturing
  - Device is open to initial programming using SWD or another wired interface.
    - UART, SPI/ I^2C
  - Customer debug is unlocked
- Customer Production
  - Customer debug is locked/unlocked based on customer security policy.

This is the boot mode for all non-secure SKU devices. In addition, even for secure SKU, if the life-cycle state is Customer Manufacturing (INFO0 has not yet been programmed by customer), or if the **INFO0:SECURITY:SECBOOT** is set to disabled, the Ambiq Secure Boot Loader defaults to non-secure boot.

If the image is invalid, a valid firmware image needs to be installed using either a debugger or supported wired interface for update.

For more details about Secure Life Cycles, see *Apollo3 Blue Security Getting Started Guide*.

1.2 Non-Secure Boot Enabled

Apollo3 Family devices ship by default in non-secure mode. In this configuration, the INFO0 block is not required to be programmed. Customer firmware can be installed either by using a debugger port (SWD) from the Ambiq-provided Flash Helper functions, or by a supported wired interface. The firmware is expected to be installed at offset 0xC000 in internal Flash. The Ambiq Secure Boot Loader provides the loader support to install the firmware and begin execution at the installation offset. No additional programming is required.

During the development phase, this configuration will allow for firmware programming and reprogramming as needed and is also more suitable for software debug.

If the device is to be programmed as a "non-secure boot" part by initializing the INFO0 block, care must be taken to ensure the device is properly initialized. Although the device may use 'non-secure boot', other security features can and will still be leveraged, such as secure OTA.

Information on how to do this is in *Section 5 - Firmware Programming of the Apollo3 Blue Security Getting Started Guide*.

1.3 Customer Production

In this state, customers can choose the appropriate security configuration depending on their production flow and security requirements. See *Apollo3 Blue Security* documentation for more details.

SECTION

2

Configuring and Programming INFO0 with SWD

This section will walk through the process of generating an INFO0 configuration for wired update, programming the INFO0 configuration using JLINK Commander, and disabling and re-enabling SWD debugger access.

Ambiq recommends becoming familiar with programming INFO0 first, since re-enabling the debugger can only be done by reprogramming INFO0. The user will first need to be able to generate INFO0 using the python scripts in the SDK located here:

```
%ambiqsuite%/tools/apollo3_scripts/
```

This section is particularly important. If the user wishes to disable the SWD, and re-enable it later, they should have two separate **info0.bin** files. One disabling the debugger and one re-enabling it.

> **NOTE:** The INFO0 bin file disabling SWD must also enable the wired updated feature, or the user may never be able to re-enable the debugger, as the default way to update INFO0 and FW image is JLINK through SWD.

Two scripts are needed for this purpose:

- **create_info0.py** – Used to create the correct INFO0 configuration based on the user's needs.
- **create_cust_wireupdate_blob.py** – Used to update over the wire.

This guide leverages the UART interface for wired update.

2.1 Generate INFO0 for UART Operation

First, configure INFO0 with a GPIO override provision. Setting a GPIO override allows for a forced image update or recovering a failing device. The **INFO0:SECURITY_WIRED_CFG:TIMEOUT** setting specifies the time the SBL will poll the configured wired update interface before proceeding through the boot flow.

> **NOTE:** Apollo3 Blue and Blue Plus devices ship with an uninitialized INFO0 and flash. When INFO0 is uninitialized, the default wired update interface is available. The interface defaults to UART0 using UART-RX on pin 49 and UART-TX on pin 48, baudrate 115200 with no flow control.

Create INFO0 image with the following parameters:

1. GPIO Override is set to pin 16 (0x10) active low.
2. Baud rate for INFO0 UART is set at 115200 (0x1C200).
3. Main image is expected at 0xC000.
4. Apollo3 Family is configured for UART-RX pin 23 (0x17) & UART-TX pin 22 (0x16).
5. Timeout is set at 5 seconds (--wTO 5000).

The resulting python command is as follows:

- `cp keys_info0.py keys_info.py`
- `python3 create_info0.py --valid 1 info0 --pl 1 --u0 0x1C200c0 --u1 0xFFFF1617 --u2 0x2 --u3 0x0 --u4 0x0 --u5 0x0 --main 0xC000 --gpio 0x10 --version 0 --wTO 5000 --chipType apollo3p`

2.2 Program INFO0 Through JLink Commander

The AmbiqSuite SDK provides a Windows batch file **program_info0.bat** which uses the JLink Commander scripting language to program INFO0 using SWD. This is in: **SDK/tools/apollo3_scripts**. The script needs to edit the file for the location of info0.bin if it is not in the same directory as the batch file.

Run this from windows command line:

```
./program_info0.bat AMA3B2KK-KBR
```

It is important to note that when using this method there is no built-in error checking. Users need to independently verify that programming was successful (e.g., by reading the InfoSpace back and then comparing with expected values).

Ambiq provides a method to do that over JLink with the following command:

```
./verify_info0.bat AMA3B2KK-KBR
```

Once the user is familiar with this process and its limitations, continue to the next section.

SECTION

3

SWD Access

The **SECURITY:DEBUG_PROT** bit controls SWD access, and when cleared (e.g., programmed to '0'), disables debugger access completely. This provides the first line of defense against unwanted entry. This section explains the options for disabling SWD debugger access, and how to re-enable it.

> **NOTE:** Prior to starting the process of disabling the debugger, the user should save INFO0 to avoid losing INFO0 information later in the re-enabling step.

3.1 Disabling the Debugger (SWD)

The user can do either of the following to disable the debugger as they both perform the same functionality.

1. Set **INFO0-> INFO0_SECURITY_DEBUG_PROT = 0** using INFO0 programming.
   Using **create_info0.py** python file with [--dbgprot {0,1}]
   (--dbgprot 0 disables protection)
   - `cp keys_info0.py keys_info.py`
   - `python3 create_info0.py --valid 1 info0 --pl 1 --u0 0x1C200c0 --u1 0xFFFF1617 --u2 0x2 --u3 0x0 --u4 0x0 --u5 0x0 --main 0xC000 --gpio 0x10 --dbgprot 0 --version 0 --wTO 5000 --chipType apollo3p`
2. Call **am_hal_flash_debugger_disable** inside the code.

3.2 Re-enabling the Debugger (SWD)

To re-enable the debugger, the user must save INFO0, set the I**NFO0->SECURITY_DEBUG_PROT** bit to 1, and then reprogram INFO0. This is done through a wired update or OTA, as described in the next Chapter.

SECTION

4

Wired Update and Over-The-Air (OTA)

This section provides walkthroughs for updating INFO0 using the Wired Update procedure and the Over-The-Air (OTA) procedure. First, we will go over the development environment.

4.1 Development Environment

4.1.1 Hardware

This example runs on the Apollo3 EVB and the Apollo3 Blue Plus EVB, make sure you have one available to run the examples.

For details on the EVBs, check:

- Apollo3 EVB Quick Start Guide
- Apollo3 Blue Plus EVB Quick Start Guide

4.1.2 Software

- Install the latest AmbiqSuite for the Apollo3 and Apollo3 Blue Plus
- Install Python 3.8.x to run the helper scripts for file generation.
- Install either Keil, IAR, or gcc compiler for code generation if needed.

4.1.3 Applications (for OTA)

- Android
- Install Ambiq_AMOTA_1.0.0.apk from the AmbiqSuite/tools/apollo3_amota directory

4.2 Programming INFO0 Over the Wire Update

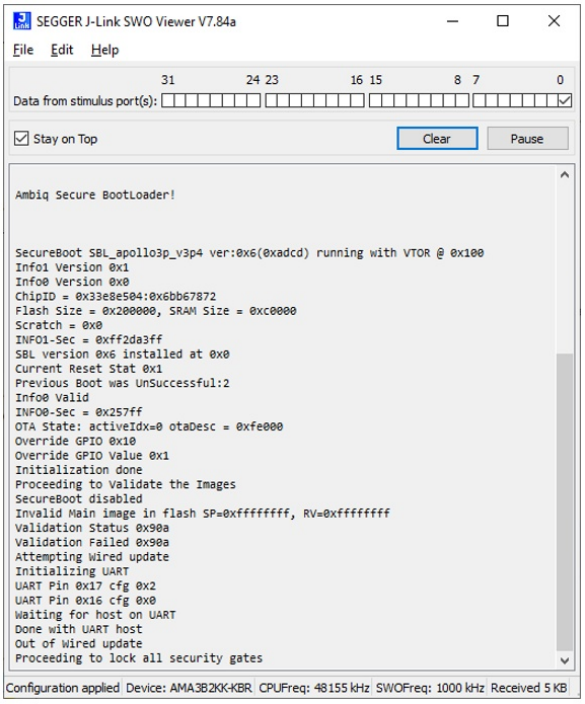
Use the following procedure to program INFO0 through wired update.

For the Wired Update procedure, the user will need to navigate to:

%ambiqsuite%/tools/apollo3_scripts/

For this example, an Apollo3 Blue Plus EVB is used and powered with the UBS-C cable.

1. Use the **create_info0.py** python file with [--dbgprot {0,1}] with the following commands:
   - **cp keys_info0.py keys_info.py**
   - **python3 create_info0.py --valid 1 info0 --pl 1 --u0 0x1C200c0 --u1 0xFFFF1617 --u2 0x2 --u3 0x0 --u4 0x0 --u5 0x0 --main 0xC000 --gpio 0x10 --version 0 --wTO 5000 --chipType apollo3p**
2. Create INFO0-NOOTA Wired Update Image blob from the INFO0 image in the previous step.
   - **python3 create_cust_wireupdate_blob.py --bin info0.bin -i 32 --load-address 0**
3. Upload the file to the device using wired update and the following method:
   a. Reset the Apollo3 Blue Plus EVB by pressing **SYSTEM RESET**.
   b. Hold **BTN0** while pressing **SYSTEM RESET** to see the following SWO output: Note the "Info0 Valid". The Apollo3 Blue Plus EVBs are shipped with valid INFO0 and the binary_counter example pre-programmed.

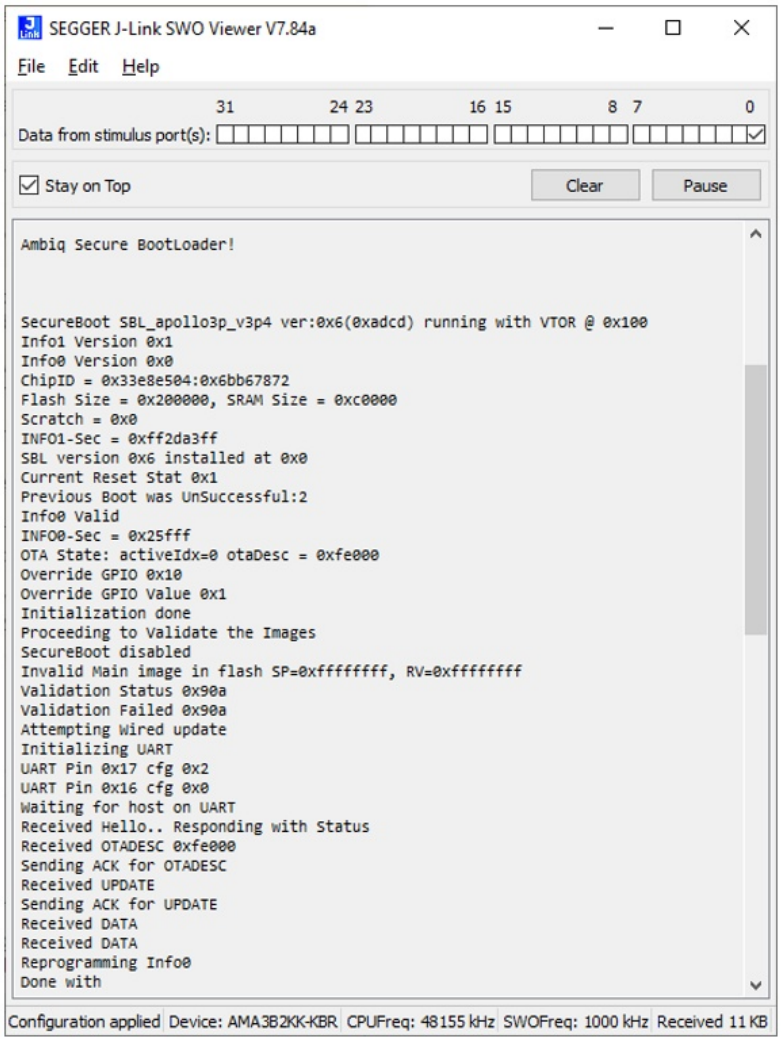


c. Within 5 seconds, use the UART Wired Update script to transfer the wired update blob to the Secure Bootloader:

```
python uart_wired_update.py -b 115200 COM<X> -r 0 -f
wuimage.bin -i 32
```

Where **COM<X>** is the PC COM port connected to the Apollo3 Blue Plus EVB.

After which the display on the JLink SWO viewer should be:



4.3 Programming INFO0 Using Over the Air Update (OTA)

Use the following procedure to program INFO0 through OTA update.

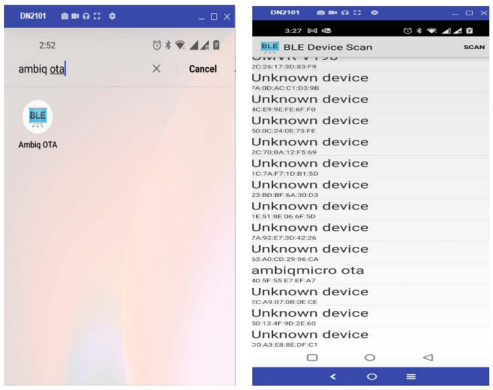
1. For the OTA Update procedure, navigate to:
   `%ambiqsuite%/tools/apollo3_amota/scripts/`
2. Type the following commands in sequence, and press **Enter** to take the **info0.bin** from the wired update section:
   a. `cp ../../apollo3_scripts/info0.bin starter_bin_apollo3p_blue.bin`
   b. `cp ../../apollo3_scripts/keys_info0.py ../../apollo3_scripts/keys_info.py`
   c. `python3 ../../apollo3_scripts/create_cust_image_blob.py --bin starter_bin_apollo3p_blue.bin --load-address 0x0 --magic-num 0xcf -o ../../apollo3_scripts/temp_info0_nosecure_ota --version 0x0`
   d. `python3 ota_binary_converter.py --appbin ../../apollo3_scripts/temp_info0_nosecure_ota.bin -o update_binary_apollo3p_blue`
   e. `rm ../../apollo3_scripts/temp_info0_nosecure_ota.bin`

This creates the file **update_binary_apollo3p_blue.bin** for the OTA update. Change the output names as desired or leave them as they are.

4.4 Android

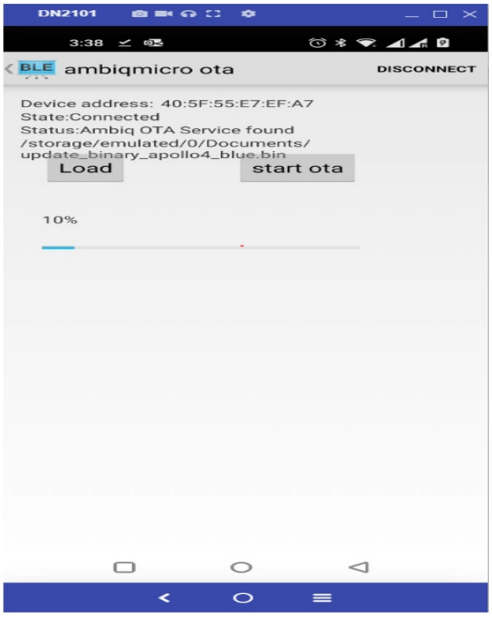
Use the following procedure to use Ambiq OTA in a Android device.

1. Go to Google Play, find Ambiq OTA, and install.
2. Find the Ambiq OTA app on the device, and open it.
3. Copy the **update_binary_apollo3p_blue.bin** to a directory on the device.
4. Check and make sure the **apollo3p_evb** is running the **SDK/boards/apollo3p_evb/examples/ble_freertos_amota** example.
5. Open the Ambiq OTA app, and scan for BLE devices.



6. Look for either Packet or **ambiqmicro_ota**.
7. Select the device and click **Load**.
8. Select the file **update_binary_apollo3p_blue.bin**, and then select start OTA. The system will reboot automatically after the upgrade is successful.

The device will start downloading and showing progress:



9. Once the download is complete, click **Reset** on the **apollo3p_evb** and the device will update the info0 space and start the BLE app again.

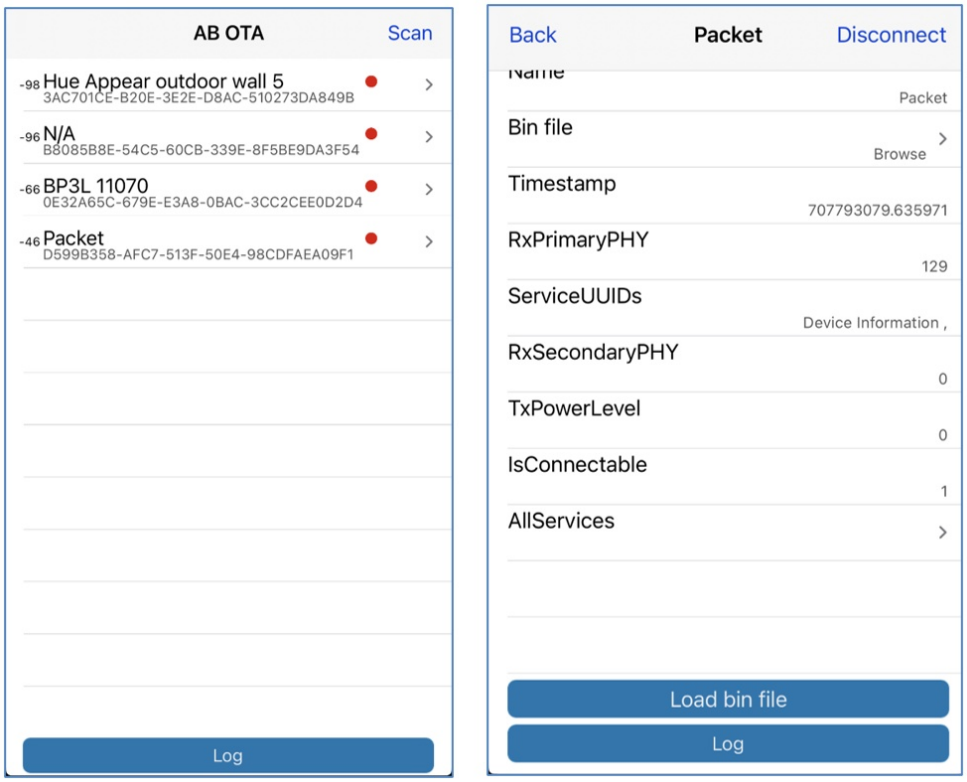
4.5 Apple iOS

Use the following procedure to use Ambiq OTA app on a Apple iOS device.

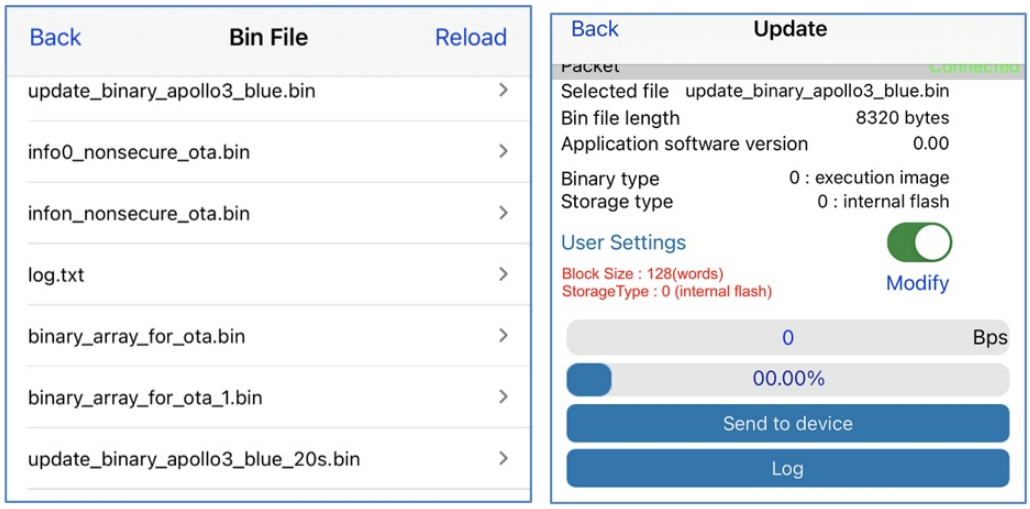
> **NOTE:** If the iOS is in Dark Mode, switch to Light Mode.

1. Go to the **App Store** and download the Ambiq OTA app.

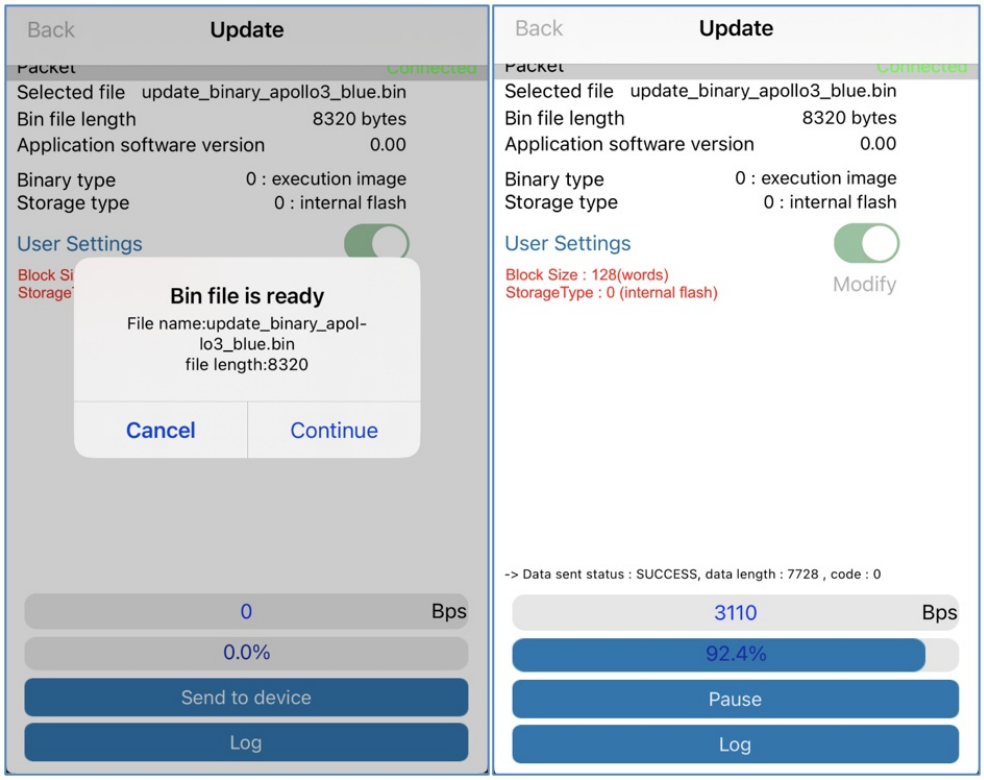
2. Using iTunes with the device connected, select the Ambiq OTA app.
3. Use **File Sharing** to place the **update_binary_apollo3p_blue** file in the Ambiq OTA app file section.
4. Check and make sure the **apollo3p_evb** is running the **SDK/boards/apollo3p_evb/examples/ble_freertos_amota** example.
5. Open the Ambiq OTA app, and scan for BLE devices.
6. Look for either Packet or **ambiqmicro_ota**.
7. Select the device and click **Load bin file**.



8. Select the **update_binary_apollo3p_blue.bin** file.
9. On the next window, click **Send to device**.



The device will start downloading and showing progress:

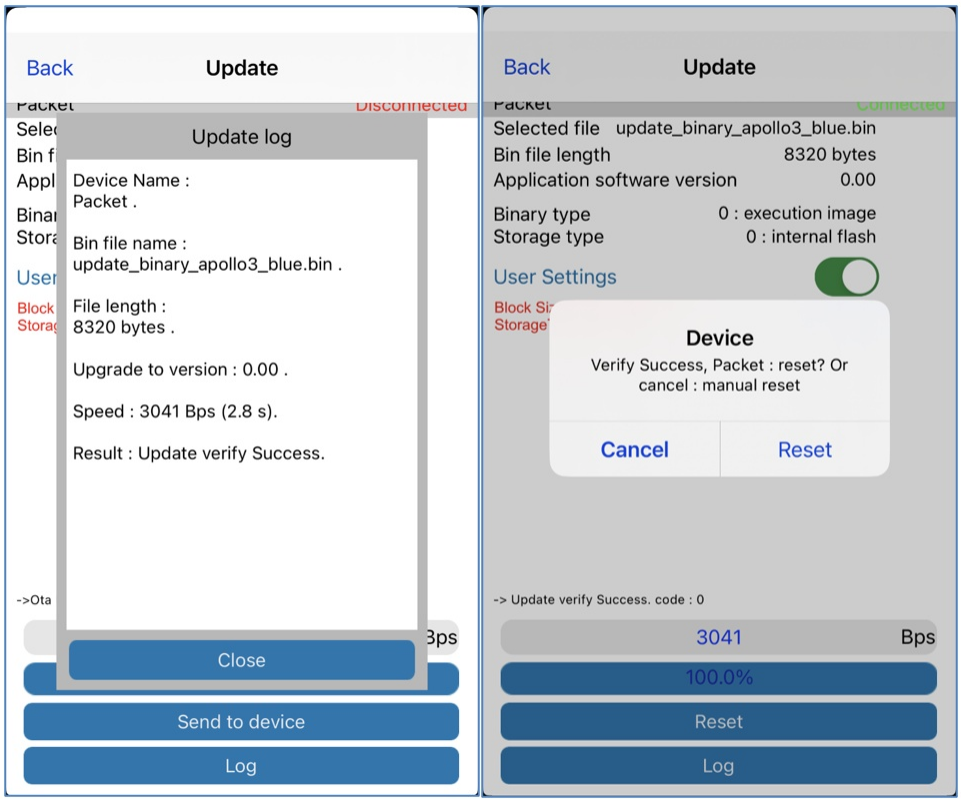


Once the download is complete, you will get confirmation.

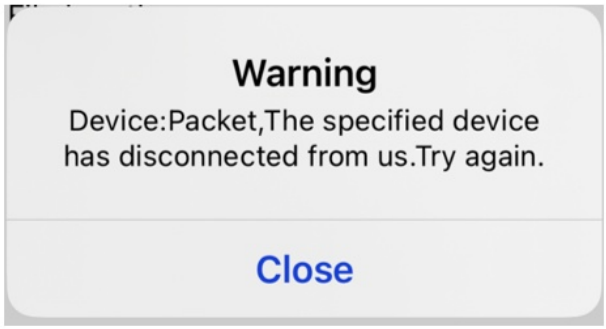
10. Close the confirmation and another dialog opens.

11. Click **Reset**.

The device will update the info0 space and start the BLE app again.



Once the **Reset** is clicked, you will lose connection with the device. This is known and you can continue and finish the process by checking that your INFO0 space is updated.



4.6 SECURITY Register

Security protection bits
OFFSET: 0x00000010
ADDRESS: 0x50020010

This 32-bit word contains the customer programmable security.

Table 4-1: SECURITY Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD					SECPOL			KEYWRAP				RSVP	SECBOOTONRST			RSVP	SECBOOT			PLONEXT	SDBG	BOOTLOADER_AT_RESET	EN_CUST_INFO_ERASE	EN_CUST_INFO_PROG					SECURE_LOCK	RSVD	RSVD	DEBUG_PROT

Table 4-2: SECURITY Register Bits

Bit	Name	Reset	RW	Description
0	DEBUG_PROT	0x1		Debugger protection.

6500 River Place Boulevard, Building 7, Suite 200, Austin, TX 78730
www.ambiq.com
sales@ambiq.com
+1 (512) 879-2850

A-SOCAP3-ANGA06EN v1.0
July 2023