



SmartCow[®]
An AI Engineering Company

Apollo Audio/Video AI Engineering Kit

User's Manual

Copyright and legal statement

©2022 SmartCow AI Technologies Ltd. All rights reserved.

No part of this document may be reproduced, translated, modified, published, distributed, transmitted, or displayed in any form or by any means, without the prior written permission from SmartCow AI Technologies Ltd. SmartCow® is a registered trademark of SmartCow AI Technologies Ltd. Otherwise, you will be responsible for any infringement of copyright law.

All other product names, brands, or logos used herein are the trademarks or registered trademarks of their respective owners. All such material is used with the permission of the owners. The content of this document is furnished confidential, privileged and for informational and instructional use only. It is subject to change without notice, and should not be construed as a commitment by SmartCow. SmartCow acts in good faith and attempts to ensure that content is accurate, complete or reliable, but it does not represent it to be error-free. SmartCow, its subsidiaries, the directors, employees, and agents assume no responsibility or liability for any inaccuracies, omissions, or errors that may appear in the content of this document.

Contents

Copyright and legal statement.....	ii
Chapter 1: About the Apollo engineering kit.....	4
1.1 Features and key benefits.....	4
1.2 Specifications.....	4
1.3 Programmable buttons.....	6
1.4 Apollo pin header layouts.....	7
1.5 About the OLED display.....	9
Chapter 2: Setting up Apollo hardware.....	10
2.1 Setting up audio peripherals.....	10
2.2 Configuring the camera.....	12
2.3 Enabling the I2S interface and installing and updating libraries.....	12
Chapter 3: Installing RIVA on Apollo.....	13
Appendix A: Apollo libraries.....	15
Pre-packaged and ready to use applications.....	15
Appendix B: Troubleshooting.....	16
Hardware issues.....	16
Software issues.....	17

Topics:

- [Features and key benefits](#)
- [Specifications](#)
- [Programmable buttons](#)
- [Apollo pin header layouts](#)
- [About the OLED display](#)

Apollo is an audio/video AI engineering kit based around the NVIDIA® Jetson Xavier™ NX computing module, enabling developers to build applications with image, conversational, and audio AI capabilities. It includes built-in microphones, speaker terminals, a camera module, and an OLED display.

- Prepackaged with NVIDIA DeepStream and NVIDIA RIVA
- Preloaded with NLP examples and hardware usage examples.
- Two programmable buttons

1.1 Features and key benefits

Apollo is an engineering kit specialized in audio and video, which also includes the software necessary to provide an end-to-end AI development platform.

At Apollo's core is NVIDIA's Jetson Xavier NX, an embedded Linux processor with an integrated GPU enabling the deployment of multiple AI models. Apollo's built-in hardware, in conjunction with RIVA and DeepStream, enables you to configure and run various audio and vision models.

Apollo also includes two programmable buttons, Ethernet and USB ports, and a Mini DisplayPort. In case of additional hardware needs, Apollo also provides a GPIO header.



CAUTION: Exercise caution while handling Apollo's pin header because it is sharp-pointed.

Table 1: What's inside the box?

Item	Units	Highlights
Apollo Engineering Kit	1	Onboard audio-visual sensors with the following components: <ul style="list-style-type: none">• 4 microphones• 2 speaker outputs• An 8MP camera module• An OLED display• 2 programmable buttons
Power cord	1	Connects the kit to an external power source
Adaptor	1	Use A/C power from a standard wall output
Mounting frame	1	The kit is mounted upright for convenience of use.

1.2 Specifications

The following figure shows the Apollo audio/video engineering kit.

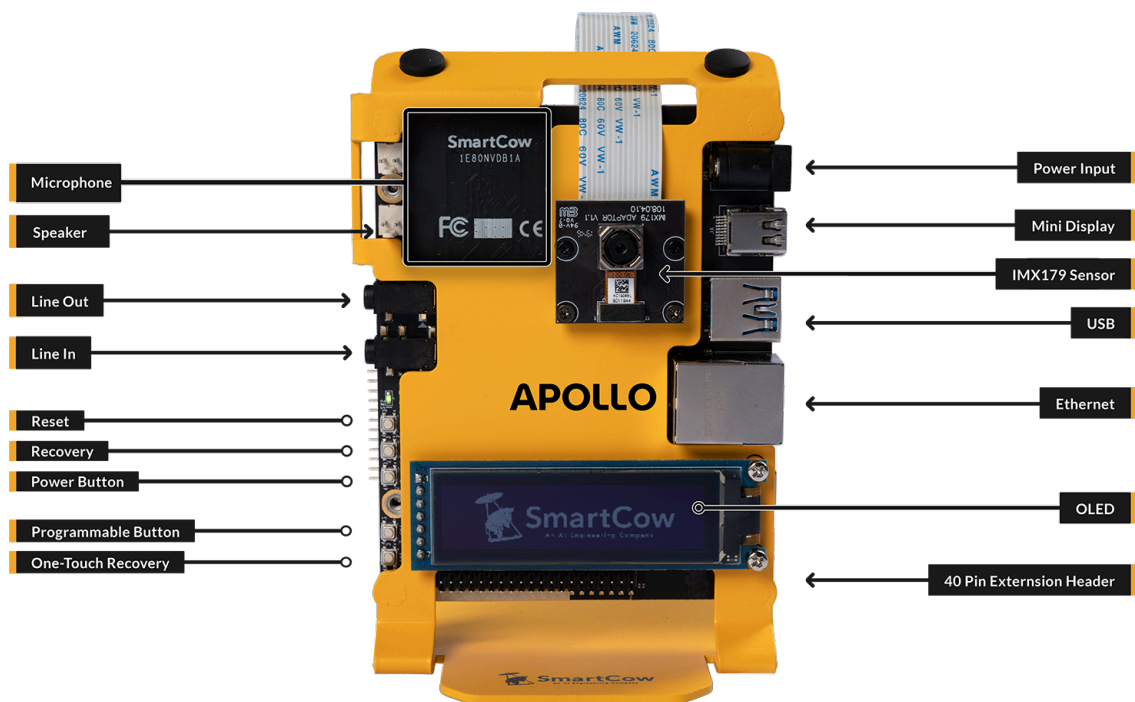


Figure 1: Apollo front side

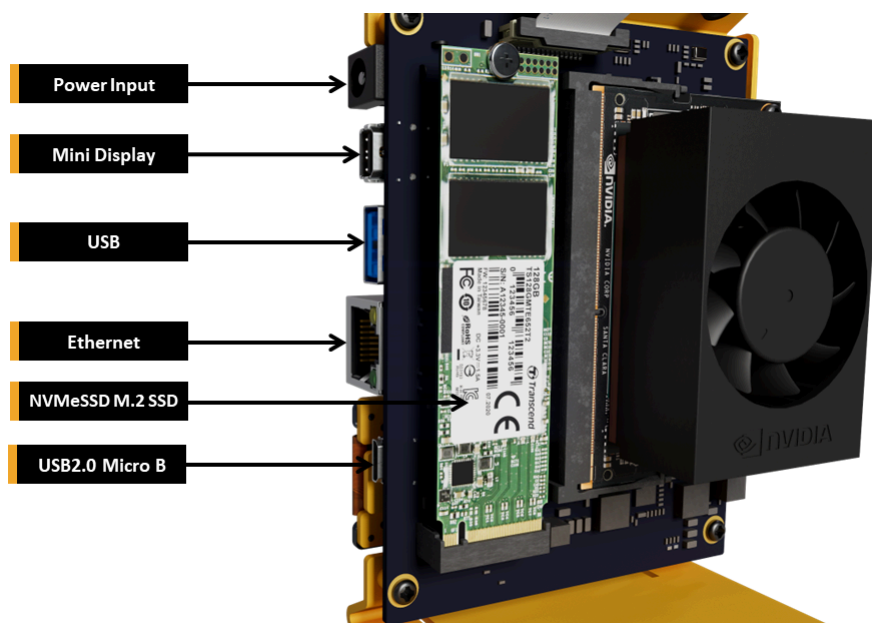


Figure 2: Apollo input/output and expansion slots rear side

The following table lists some of the major parts in the Apollo device and their specifications.

Table 2: Specifications

Item	Description	Specification
NVIDIA® JetsonXavier™ NX	CPU	6-core NVIDIA Carmel ARM®v8.2 64-bit CPU 6MB L2 + 4MB L3 processor
	GPU	NVIDIA Volta™ architecture with 384 NVIDIA CUDA® cores and 48 Tensor cores
	Memory	8GB 128-bit LPDDR4x @ 1600 MHz 51.2GB/s with 128G NVMeSSD M.2 SSD

Item	Description	Specification
		16 GB 128-bit LPDDR4x @ 59.7GB/s with 128G NVMeSSD M.2 SSD
	Storage	16 GB eMMC 5.1
Physical I/O	Display	1 x Mini DP
	Ethernet	1 x RJ45 GbE (10/100/1000)
	Tact Switch	Recovery / reset / power / programmable buttons x 2
	USB	1 x USB3.2 Gen1 Type A 1 x USB2.0 Micro B (OTG only)
	Line in	1 x 3.5mm phone jack
	Line out	1 x 3.5mm phone jack
Internal Connector	Speaker out	2 x pin header (2-pin with pitch 2.54mm)
	Mic in	4 x MEMS microphone
	Camera	1 x 15-pin FPC for MIPI CSI-II 2 Lanes (equipped with 8MP IMX179 MIPI camera)
	2-pin header	1 x RTC with CR2032 battery
	4-pin header	1 x Fan (equipped with Xavier NX Cooler)
	7-pin header	1 x SPI for OLED (equipped with 2.08-inch OLED)
	12-pin header with pitch 2.0mm	1 x UART (Debug only)
		1 x Power LED
		1 x Auto-power-on
		1 x Reset
		1 x Recovery
		1 x Power
	40-pin header with pitch 2.0mm	1 x UART
		2 x I2C
		1 x SPI
	Expansion slots	1x M.2 2280 Key M (equipped with 128G NVMe SSD)
		1 x M.2 2230 Key E
Environment	Power input	12V DC input with DC jack
	Dimension	PCBA with base frame: 81 x 69 x 125 mm (L x W x H)
	Operating temperature	0°C ~ 50°C
	Storage temperature	-25°C ~+80°C
	Storage humidity	95% @ 40 °C (non-condensing)

1.3 Programmable buttons

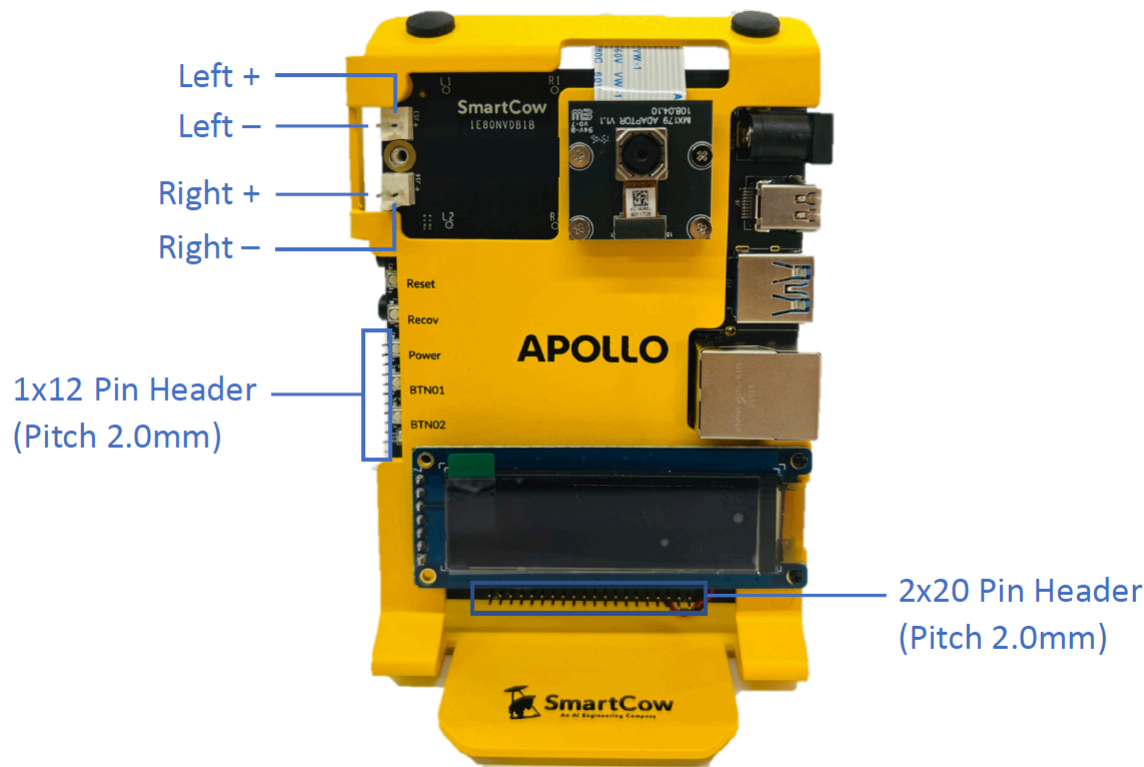
Apollo has two programmable buttons for developers to add their applications and broaden the scope of building potential projects.

1.4 Apollo pin header layouts

This section lists the sets of pin headers of Apollo.

Pin header to connect a speaker

SmartCow recommends that you use a 1Watt, 40hm speaker.



1x12 pin header

The following figure shows the 1X12 pin header of Apollo.

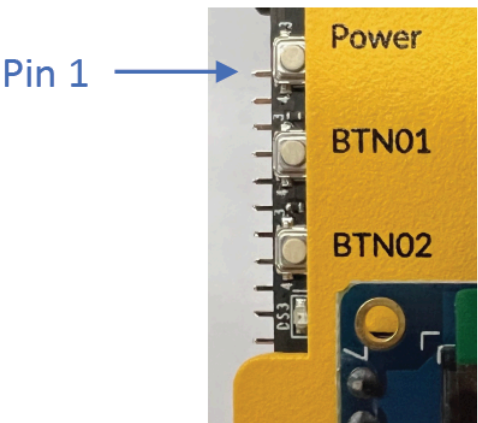


Table 3: 1x12 Pin define

Pin	Pin Define
1	LED-

Pin	Pin Define
2	LED+
3	DebugUART_RX
4	DebugUART_TX
5	Autopower setting, Off=Plug
6	Autopower setting, Off=Plug
7	ResetButton
8	ResetButton
9	ForceRecovery Button
10	ForceRecovery Button
11	PowerButton
12	PowerButton

2x20 pin headers

The following figure shows the 2x20 pin headers of Apollo.

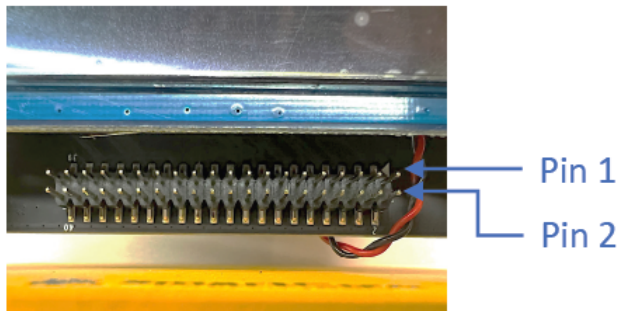


Table 4:

Used by Board Device	Pin Define	Pin	Pin	Pin Define	Used by Board device
	3.3V	1	2	5.0V	
	I2C1_SDA	3	4	5.0V	
	I2C1_SCL	5	6	GND	
I2SCodec	GPIO09	7	8	UART1_TXD	
	GND	9	10	UART1_RXD	
	UART1_RTS	11	12	I2S0_SCLK	I2SCodec
OLED	SPI1_SCK	13	14	GND	
BTN01	GPIO12	15	16	SPI1_CS1	
	3.3V	17	18	SPI1_CS0	OLED
	SPI0_MOSI	19	20	GND	
	SPI0_MISO	21	22	SPI1_MISO	
	SPI0_SCK	23	24	SPI0_CS0	

Used by Board Device	Pin Define	Pin	Pin	Pin Define	Used by Board device
	GND	25	26	SPI0_CS1	
	I2C0_SDA	27	28	I2C0_SCL	
	GPIO01	29	30	GND	
OLED	GPIO11	31	32	GPIO07	OLED
BTN02	GPIO13	33	34	GND	
I2SCodec	I2S0_FS	35	36	UART1_CTS	
OLED	SPI1_MOSI	37	38	I2S0_DIN	I2SCodec
	GND	39	40	I2S0_DOUT	I2SCodec

1.5 About the OLED display

Apollo has an inbuilt 2.08 inch OLED screen with an SPI interface, and can display in white.

For more information about using the OLED display, including setting up, debugging, and customizing the screen with Apollo, check the `/opt/apollo/base/oled` directory on Apollo.

When Apollo is switched on, the OLED displays the SmartCow logo by default.



Note: Do not disassemble the OLED model. Ensure that Apollo is used and stored away from bright lights, extreme temperatures, and humidity.

Topics:

- [Setting up audio peripherals](#)
- [Configuring the camera](#)
- [Enabling the I2S interface and installing and updating libraries](#)

This section describes how to get started with setting up Apollo hardware.

To simplify the setup process, Apollo comes pre-flashed with a customized version of the Linux for Tegra BSP, which includes the necessary drivers and libraries. This eliminates the need to install these yourself, allowing you to quickly get the onboard peripherals and sensors up and running. You only need to configure the onboard microphones, camera, and external speaker to get them to work.



Note: All tutorials are based on Python and use the same program structure and common libraries.

2.1 Setting up audio peripherals

Follow these steps to set up and test the speaker and microphone components of Apollo using your computer.

Before you begin

Ensure that you have the following items on hand.

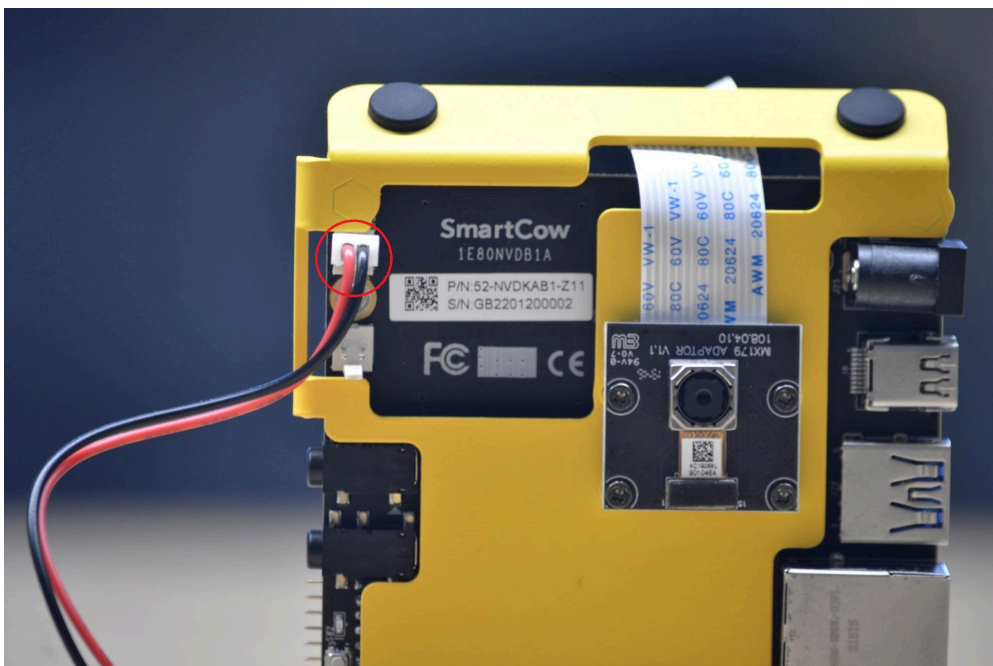
- Computer
- Micro-USB cable
- Ethernet cable
- 1Watt, 40hm speaker

About this task

The inbuilt amplifier can accommodate a variety of speakers. However, we recommend using a 1Watt, 40hm speaker.

Procedure

1. Attach an external speaker to the speaker connector at the top left section of Apollo as shown in the following figure.





CAUTION: Before you proceed, ensure that the speaker's red wire is closer to the edge of Apollo than the black wire.

2. Connect a micro-USB cable between your computer and Apollo.
3. Connect an Ethernet cable to Apollo to provide Internet access.
4. Connect Apollo to the power supply.

A green LED lights up and Apollo starts booting up. Apollo takes a few minutes to boot up. After Apollo boots up, a welcome graphic is displayed on the OLED display.



Note: At start-up, Apollo automatically checks whether a new firmware update is available. If there is an update, you are prompted on the OLED display to confirm if you wish to upgrade Apollo to this new release or to skip the update. If you confirm, Apollo automatically downloads and applies the upgrade, and you are prompted to reboot the Apollo board after completion. You can also disable this update checking procedure for a period of 30 days or re-enable it at any point in time by running the relevant scripts, as described in the README.md file located in the `/etc/apollo/services/onetouch` directory.

5. Open a terminal on your computer and SSH into the device by typing the following command and press **Enter**.

```
ssh -X nvidia@192.168.55.1
```

The `-x` option enables applications running on Apollo to forward their display to your computer.



Tip: Alternatively, you can use the network IP address for the SSH connection or connect a monitor, keyboard, and mouse to Apollo.

You are prompted to type the password.

6. Type the password.

The default password is `nvidia`

7. If you have access to the Internet and you want to update the device to the most recent packages, run the following command and press **Enter**.

```
sudo apt upgrade apollo
```

You are prompted to type the password.

8. Type the password.

The default password is `nvidia`

9. Test the audio peripherals.

- a. To record five seconds of audio from Apollo's built-in microphone, type the following command.

```
arecord -D hw:jetsonxaviernxa,0 -c 2 -d 5 -r 48000 -f S32_LE test.wav
```

where:

- `-D` option selects the sound card and the microphone channel
- `hw:jetsonxaviernxa` is the name of the sound card
- `0` selects the channel zero to set the microphone tune
- `-c 2` sets the number of channels to 2 for stereo recording
- `-d 5` sets the duration of the snippet to 5 seconds
- `-r 48000` is the sample rate of 48kHz
- `-f S32_LE` is the sample format of 32-bit little-endian
- `test.wav` is the output file name

- b. Record the `.wav` file.

- c. To playback the audio recording, type the following command.

```
aplay -D hw:jetsonxaviernxa,1 -c 2 -r 48000 -f S32_LE test.wav
```

The five second audio recording is successfully played through the Apollo speakers.

2.2 Configuring the camera

About this task

Follow these steps to configure and test the Apollo Engineering Kit's onboard 8MP IMX179 camera module.

Procedure

1. To check which resolutions the camera supports, type the following command at the terminal.

```
v4l2-ctl -d /dev/video0 --list-formats-ext
```

2. To create a pipeline to configure the camera and display the live stream using GStreamer, type the following command.

```
gst-launch-1.0 nvarguscamerasrc !  
"video/x-raw(memory:NVMM), width=820, height=616, framerate=30/1, format=NV12" !  
nvvidconv ! xvimagesink -e
```

where:

- `nvarguscamerasrc` is the source CSI camera
- `video/x-raw(memory:NVMM), width=820, height=616, framerate=30/1, format=NV12` is the caps filter to set the resolution of the frame rate
- `nvvidconv` adding `nvvidconv` to the pipeline
- `xvimagesink` to display the live feed
- `-e` to safely terminate the GStreamer pipeline

After the setup is complete, a live feed from the camera is displayed.

2.3 Enabling the I2S interface and installing and updating libraries

Enable the I2S interface to connect to the speakers and install and update libraries.

Procedure

1. Open the terminal and type the following command to open up NVIDIA GUI to configure peripherals.

```
sudo /opt/nvidia/jetson-io/jetson-io.py
```

The system prompts to type the password.

2. Type the password.

The default password is `nvidia`.

Jetson will launch.

3. Select **Configure the Jetson 40pin header** and press **Enter**.
4. Select **Configure header pins manually** and press **Enter**.
5. Select the `i2s4` peripheral and go back.
6. Select **Save pin changes**.
7. Select **Save and reboot**.

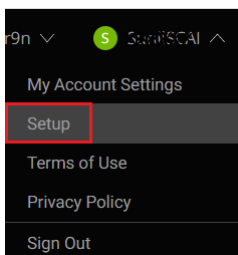
Follow these steps to install NVIDIA RIVA on Apollo.

About this task

RIVA takes approximately 3.5GB to download and install without any active services running.

Procedure

1. Create a new NVIDIA account or sign in to an existing account.
<https://ngc.nvidia.com/signin>
2. Generate an NGC API key.
 - a) Sign in to your NVIDIA account.
 - b) On the top-right corner of the page, click *display_name* > **Setup**.



The *Setup* page is displayed.

- c) Click **Get API Key**.
The *API Key* page is displayed.



Note: Carefully read the on-screen instructions.

- d) Click **Generate API Key**.
The *Generate a New API Key* dialog box is displayed.
- e) Click **Confirm**.
After the API key is successfully generated, the key is displayed on the *API Key* page.



Important: This is the only time your API Key is displayed. Keep your API Key secret. Do not share it or store it in a place where others can see or copy it. If you lose the API key, you can generate it again; however, the old API key becomes invalid.

API

API Information

Generate your own API key to use the NGC service through the Docker client. Anyone with this API Key has access to all services, actions, and resources on your behalf.

Click [Generate API Key](#) to create your own API Key. If you have forgotten or lost your API Key, you can come back to this page to create a new one at any time.

Usage

Use your API key to log in to the NGC registry by entering the following command and following the prompts:

NGC CLI

```
$ nqc config set
```



Docker™

For the username, enter 'soauthtoken' exactly as shown. It is a special authentication token for all users.

```
$ docker login nver.io
```

Username: Southtoken

```

Password: 21'a'8'3'9'a'0'1'7'5'a'2'7'0'7'6'4'5'a'4'0'1'1'7'5'4'7'6'8'7'1'4'6't'2'7'9'4'7'0'0'4'1'7'5'4'g'6'2'0'5'2'6'1'8'2'6'0'a'y'0'7'a

```



API Key generated successfully. This is the only time your API Key will be displayed. Keep your API Key secret. Do not share it or store it in a place where others can see or copy it.

API Key: 281VUz8W00p02r02-T5r0T0F0N0280F0P0F0C0P0S0P0T0F0T022Y02-0V010F0Y02



3. To automatically pull and extract the RIVA containers, run the `./download_riva.sh` script in the `/opt/apollo/sdk/riva` directory.
4. At the Docker container CLI, perform the following configuration steps.

```
jupyter notebook --generate-config
```

```
jupyter notebook password
```



Note: You can specify any password you want.

5. Launch the jupyter notebook session for `asr-python-basics`, `asr-python-boosting`, and `tts-python-basics` by running the following command.

```
jupyter notebook --allow-root --notebook-dir=/work/notebooks
```

You are prompted to launch your web browser using a link in the terminal and also prompted to type the password you previously used.

You should now have access to the notebooks.

Topics:

- [Pre-packaged and ready to use applications](#)

SmartCow provides access to public GitHub repositories containing examples with the RIVA and DeepStream SDKs that make use of Apollo's hardware to run audio and vision models.

This section contains a list of sample libraries and applications that come pre-packaged with Apollo.

Pre-packaged and ready to use applications

PyAudio

PyAudio is a library that enables connection between Python and PortAudio, the cross-platform audio I/O library. With PyAudio, you can use Python to play and record audio on a variety of platforms, including GNU/Linux.

You can use PyAudio for the microphone and the speaker. For the microphone, in particular, PyAudio is capable of both recording `.wav` files and streaming audio, making it a versatile and useful tool. PyAudio can be easily configured to generate mono or stereo files with varying bit depths and bit rates.

Typically, audio AI models are designed to receive 16 bit data. However, the Apollo microphones return 24 useful bits of data. To address this, Apollo software includes an operation that converts microphone data to its 16 bit equivalent while streaming. This enables AI audio models to run efficiently in real time.

The Apollo-Audio repository contains information on getting started with recording and playing `.wav` files with PyAudio, and other applications such as a volume meter and a small audio library to delve into some minor processing tasks. For more information on PyAudio visit <https://people.csail.mit.edu/hubert/pyaudio/>

Natural language processing

Natural language processing (NLP) is allowing machines to interpret and respond to textual data. Common applications include text classification, language translation, and text analytics such as named entity recognition and sentiment analysis. The Apollo Engineering Kit comes pre-packaged with the Apollo-NLP Library to train your own model, and a small demonstration.

Chatbots

Chatbots are NLP-based software applications that can be used to converse with humans or other chatbots. Chatbot complexity varies greatly, with simple bots scanning for key phrases and more complex bots utilizing cutting-edge NLP pipelines. For more information about chatbots, check a small project inside the Apollo NLP library.

Audio recognition

Audio recognition is identifying an entity or phenomenon by sound computation. For more information on audio recognition using spectrograms, Apollo comes with a simple audio recognition program.

Speaker verification

An application of audio recognition is speaker verification. The primary application of speaker verification is to verify a person's identity. A user is asked to identify themselves and then speak. If the system confirms the user's identity, the user is granted access. A typical system would have two stages: registration/enrollment and login/verification. Apollo has a Apollo-Speaker-Verification demo to use as a template.


Topics:


- [Hardware issues](#)
- [Software issues](#)

This section describes the troubleshooting scenarios and frequently asked questions.

Hardware issues

Table 5: Hardware issues

Issue	Workaround
Speaker is not generating sound.	<ol style="list-style-type: none"> 1. Open the NVIDIA GUI to see which peripherals are enabled by running the following command. <pre>sudo /opt/nvidia/jetson-io/jetson-io.py</pre> 2. Ensure that the i2s5 is enabled. If not, configure the 40-pin expansion header and enable i2s5. You are prompted to save and reboot. 3. After you reboot your device, configure the sound card appropriately by running the following command. <pre>amixer -c jetsonxaviernxa sset "I2S5 Mux" ADMAIF2</pre> 4. Run a speaker test by running the following command. <pre>speaker-test -D hw:jetsonxaviernxa,1 -c 2 -r 48000 -F S16_LE -t sine -f 500</pre>
Microphone is not recording.	<ol style="list-style-type: none"> 1. Reconfigure the sound card sub-device appropriately by running the following command. <pre>amixer -c jetsonxaviernxa cset name='ADMAIF1 Mux' I2S3</pre> 2. Record a .wav file by running the following command. <pre>arecord -D hw:jetsonxaviernxa,0 -c 2 -d 10 -r 48000 -f S32_LE test.wav</pre> 3. To put to test the test.wav file on the speaker to ensure that sound was actually recorded, run the following command. <pre>aplay -D hw:jetsonxaviernxa,1 -c 2 -r 48000 -f S32_LE test.wav</pre> <p> Note: The I2S peripheral which connects to the microphones is constantly running, there is no need to address that.</p>
OLED screen is not turning on.	If the OLED screen is not turning on, check out the demo applications in /opt/apollo/base/oled. You can find examples and setup scripts to verify the functionality.

Issue	Workaround
Camera is not working.	<p>If your camera is not functioning after connecting to Apollo, run the following command to verify if your camera is working:</p> <pre>ls /dev/video* nvgstcapture-1.0 --camsrc=0 --cap-dev-node=<N></pre> <p> Note: <N> is the /dev/videoN</p>
The fan is not working, or the device is overheating.	<p>Reset the fan by running the following command.</p> <pre>sudo jetson_clocks --fan</pre>
Other hardware issues	<p>Follow this link to receive assistance for any other hardware issue.</p> <p>https://forms.gle/kJ9HXRYErWMff8rF7</p>

Software issues

If you are having software issues, you can post your query along with the tag #Apollo on NVIDIA Developer forum from the link below:

<https://forums.developer.nvidia.com/c/agx-autonomous-machines/jetson-embedded-systems/70>

The SmartCow team will look into your posted issue and get back to you as soon as possible.

Document control

Document Version	Product Version	Release Date
1.0	1.0	2022-11-28