



QUICK START GUIDE

VIA ARTiGO A950

Android 10.0 EVK



Copyright

Copyright © 2022 VIA Technologies Incorporated. All rights reserved.

No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise without the prior written permission of VIA Technologies, Incorporated.

Trademarks

All brands, product names, company names, trademarks and service marks are the property of their respective holders.

Disclaimer

VIA Technologies makes no warranties, implied or otherwise, in regard to this document and to the products described in this document. The information provided in this document is believed to be accurate and reliable as of the publication date of this document. However, VIA Technologies assumes no responsibility for the use or misuse of the information (including use or connection of extra device/equipment/add-on card) in this document and for any patent infringements that may arise from the use of this document. The information and product specifications within this document are subject to change at any time, without notice and without obligation to notify any person of such change.

VIA Technologies, Inc. reserves the right to make changes to the products described in this manual at any time without prior notice.



Revision History

Version	Date	Remarks
1.00	20/10/2022	Initial release

Table of Contents

1.	Introduction.....	1
1.1	EVK Package Contents.....	1
1.1.1	Firmware Folder Contents	1
1.1.2	Document Folder Contents.....	1
1.1.3	Tool Folder Contents.....	1
1.2	Version Information and Supported Features.....	2
2.	Image Installation	3
2.1	Installing with the Fastboot Mode.....	3
3.	Hardware Functions	7
3.1	Using the Debug Console.....	7
3.2	Changing the Kernel Debug Level	7
3.3	Checking the BSP Version.....	7
3.4	DVFS.....	7
3.5	Display.....	8
3.6	MTK NeuroPilot AI APU Hardware Acceleration	8

1. Introduction

This Quick Start Guide provides an overview of how to boot the Android image for the VIA ARTiGO A950 and configure the supported hardware functions in the build.

The VIA ARTiGO A950 Android 10.0 EVK is developed based on the MediaTek Android 10.0 BSP, and it enables the hardware features of the VIA ARTiGO A950.

1.1 EVK Package Contents

There are three folders in the package listed as below.

Firmware folder		Description
VIA_ARTiGO_A950_Android_10.0_EVK.zip		Android image
Document folder		Description
VIA_ARTiGO_A950_Android_10.0_EVK_Quick_Start_Guide.pdf		Quick Start Guide
Tool folder		Description
VIA_Android_USB_Driver.zip		VIA USB driver

1.1.1 Firmware Folder Contents

VIA_ARTiGO_A950_Android_10.0_EVK.zip: Contains the precompiled Android image for evaluating the VIA ARTiGO A950.

1.1.2 Document Folder Contents

VIA_ARTiGO_A950_Android_10.0_EVK_Quick_Start_Guide.pdf: This Quick Start Guide provides an overview on how to boot the Android image for the VIA ARTiGO A950 and configure the supported hardware functions in the build.

1.1.3 Tool Folder Contents

VIA_Android_USB_Driver.zip: VIA USB driver.

1.2 Version Information and Supported Features

- Kernel version: 4.14.141
- Evaluation image: Android 10.0
- Development based on MediaTek Android 10.0 BSP
- Supports eMMC boot
- Supports HDMI display
- Supports HDMI audio output
- Supports COM1 as RS-232 mode (TX/RX) and COM as debug port
- Supports two 10/100Mbps Ethernet ports
- Supports MediaTek MT6358 Headphone and Mic-in
- Supports MediaTek MT7668 Wi-Fi 802.11ac and Bluetooth 5.0
- Supports VIA EMIO-2574 (SIM7600JC-H) 4G LTE mobile broadband miniPCle module
- Supports MediaTek NeuroPilot AI APU hardware acceleration

2. Image Installation

This section explains the setup requirements for booting from the eMMC.

The precompiled images are provided in the "Firmware" folder.

2.1 Installing with the Fastboot Mode

The first step is to prepare the "fastboot tool" and install the "VIA USB driver" into your Windows 10 host machine.

To get the "fastboot tool" for Windows, please refer to the "SDK Platform Tools release notes" chapter on the Android developer website (<https://developer.android.com/studio/releases/platform-tools>).

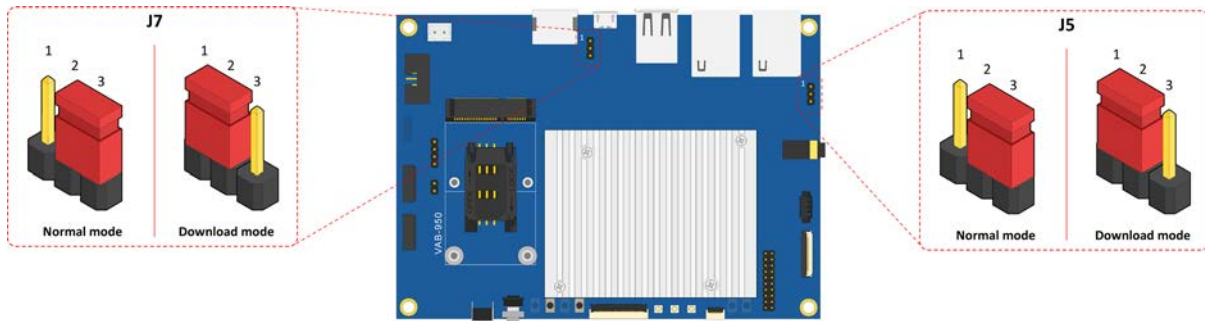
To install the "VIA USB driver", right-click on the mouse button to install the "android_winusb". When finished, it will show "The operation completed successfully".

Then connect the Windows 10 host machine and the VIA ARTiGO A950 through the Micro USB 2.0 port using the Micro USB cable.



Micro USB 2.0 port diagram

Next, on the VIA ARTiGO A950, set the two OS image jumpers (J7 and J5) to download mode as shown in the diagram below.



OS image jumpers diagram

J5 and J7 Settings	Pin 1	Pin 2	Pin 3
Normal mode (default)	Open	Short	Short
Download mode	Short	Short	Open

OS image jumper settings

Plug in the AC-to-DC power adapter to power on the VIA ARTiGO A950.

The VIA ARTiGO A950 will stop at "fastboot: processing commands" as shown in the debug console below.

```
[1202] [USB] HS is detected
[1202] [USB] USB: speed HS!!
[1202] [USB] [INTR] Reset
[1591] [USB] USB: speed HS!!
[1591] fastboot: processing commands
```

Fastboot Mode



Note:

Make sure the Windows 10 host machine detects the VIA ARTiGO A950 as an Android device.

Extract the VIA_ARTiGO_A950_Android_10.0_EVK.zip file on your Windows 10 host machine.

To install the image, use the following command:

```
D:\VIA_ARTiGO_A950_Android_10.0_EVK>set PATH=D:\platform-tools;%PATH%
D:\VIA_ARTiGO_A950_Android_10.0_EVK>viaflash.exe
+++++
VIAFlash version v1.0.0
(built: Mar 16 2022 18:03:28 UTC)
+++++
To let this tool work well, please
ensure the system is at fastboot mode!

If yes, please press any key to continue.
+++++
>> Image OS is "Android"
+++++
MBR : OK
preloader_tb8788p1_64_wifi.bin : OK
logo.bin : OK
vbmeta.img : OK
vbmeta_system.img : OK
vbmeta_vendor.img : OK
mdlimg.img : OK
spmfw.img : OK
scp.img : OK
sspm.img : OK
cam_vpu1.img : OK
cam_vpu2.img : OK
cam_vpu3.img : OK
lk.img : OK
boot.img : OK
dtbo.img : OK
tee.img : OK
super.img : OK
userdata.img : OK
+++++
=====
Start to flash images --> Android
=====
Sending 'preloader_a' (255 KB) OKAY [ 0.032s]
Writing 'preloader_a' OKAY [ 0.014s]
Finished. Total time: 0.071s

Erasing 'pgpt' OKAY [ 0.014s]
Finished. Total time: 0.024s

Sending 'pgpt' (17 KB) OKAY [ 0.026s]
Writing 'pgpt' OKAY [ 0.007s]
Finished. Total time: 0.051s

Sending 'logo' (2801 KB) OKAY [ 0.090s]
Writing 'logo' OKAY [ 0.074s]
Finished. Total time: 0.194s
```



```

Sending 'vbmeta_a' (4 KB)                                OKAY [ 0.026s]
Writing 'vbmeta_a'                                       OKAY [ 0.006s]
Finished. Total time: 0.052s

Sending 'vbmeta_system_a' (4 KB)                         OKAY [ 0.025s]
Writing 'vbmeta_system_a'                               OKAY [ 0.006s]
Finished. Total time: 0.053s

Sending 'vbmeta_vendor_a' (4 KB)                        OKAY [ 0.025s]
Writing 'vbmeta_vendor_a'                              OKAY [ 0.006s]
Finished. Total time: 0.054s

Sending 'md1img_a' (0 KB)                                OKAY [ 0.026s]
Writing 'md1img_a'                                       OKAY [ 0.006s]
Finished. Total time: 0.051s

Sending 'spmfw_a' (47 KB)                                OKAY [ 0.027s]
Writing 'spmfw_a'                                       OKAY [ 0.008s]
Finished. Total time: 0.055s

Sending 'scp_a' (608 KB)                                 OKAY [ 0.038s]
Writing 'scp_a'                                         OKAY [ 0.022s]
Finished. Total time: 0.088s

Sending 'sspm_a' (480 KB)                                OKAY [ 0.036s]
Writing 'sspm_a'                                       OKAY [ 0.019s]
Finished. Total time: 0.077s

Sending 'cam_vpu1_a' (1610 KB)                           OKAY [ 0.059s]
Writing 'cam_vpu1_a'                                   OKAY [ 0.044s]
Finished. Total time: 0.126s

Sending 'cam_vpu2_a' (9887 KB)                           OKAY [ 0.242s]
Writing 'cam_vpu2_a'                                   OKAY [ 0.220s]
Finished. Total time: 0.492s

Sending 'cam_vpu3_a' (135 KB)                            OKAY [ 0.028s]
Writing 'cam_vpu3_a'                                   OKAY [ 0.012s]
Finished. Total time: 0.062s

Sending 'lk_a' (774 KB)                                  OKAY [ 0.041s]
Writing 'lk_a'                                         OKAY [ 0.026s]
Finished. Total time: 0.091s

Sending 'dtbo_a' (8192 KB)                               OKAY [ 0.200s]
Writing 'dtbo_a'                                       OKAY [ 0.186s]
Finished. Total time: 0.410s

Sending 'tee_a' (139 KB)                                 OKAY [ 0.028s]
Writing 'tee_a'                                         OKAY [ 0.011s]
Finished. Total time: 0.062s

Sending 'userdata' (860 KB)                              OKAY [ 0.043s]
Writing 'userdata'                                     OKAY [ 0.157s]
Finished. Total time: 0.225s

-----
Platform will be rebooted and continue soon,
please wait for flash completed
-----

Rebooting into bootloader                                OKAY [ 0.004s]
Finished. Total time: 0.006s

< waiting for any device >
Sending 'boot_a' (33792 KB)                              OKAY [ 0.905s]
Writing 'boot_a'                                       OKAY [ 0.750s]
Finished. Total time: 1.723s

```

```

-----
Platform will be rebooted and continue soon,
please wait for flash completed
-----
Rebooting into bootloader          OKAY [ 0.003s]
Finished. Total time: 0.006s

< waiting for any device >
Sending sparse 'super' 1/12 (131068 KB)    OKAY [ 3.209s]
Writing 'super'                          OKAY [ 3.036s]
Sending sparse 'super' 2/12 (128270 KB)    OKAY [ 2.956s]
Writing 'super'                          OKAY [ 4.464s]
Sending sparse 'super' 3/12 (127597 KB)    OKAY [ 2.943s]
Writing 'super'                          OKAY [ 3.553s]
Sending sparse 'super' 4/12 (131068 KB)    OKAY [ 3.046s]
Writing 'super'                          OKAY [ 3.092s]
Sending sparse 'super' 5/12 (131068 KB)    OKAY [ 2.929s]
Writing 'super'                          OKAY [ 2.916s]
Sending sparse 'super' 6/12 (131071 KB)    OKAY [ 3.183s]
Writing 'super'                          OKAY [ 11.019s]
Sending sparse 'super' 7/12 (131068 KB)    OKAY [ 2.979s]
Writing 'super'                          OKAY [ 3.179s]
Sending sparse 'super' 8/12 (125124 KB)    OKAY [ 2.920s]
Writing 'super'                          OKAY [ 2.940s]
Sending sparse 'super' 9/12 (118861 KB)    OKAY [ 2.806s]
Writing 'super'                          OKAY [ 3.327s]
Sending sparse 'super' 10/12 (125744 KB)   OKAY [ 2.847s]
Writing 'super'                          OKAY [ 2.995s]
Sending sparse 'super' 11/12 (131068 KB)   OKAY [ 3.032s]
Writing 'super'                          OKAY [ 3.103s]
Sending sparse 'super' 12/12 (73272 KB)   OKAY [ 1.712s]
Writing 'super'                          OKAY [ 1.783s]
Finished. Total time: 80.451s

=====
Finished. Re-Starting system ...
Rebooting                          OKAY [ 0.003s]
Finished. Total time: 0.006s

D:\VIA_ARTiGO_A950_Android_10.0_EVK>

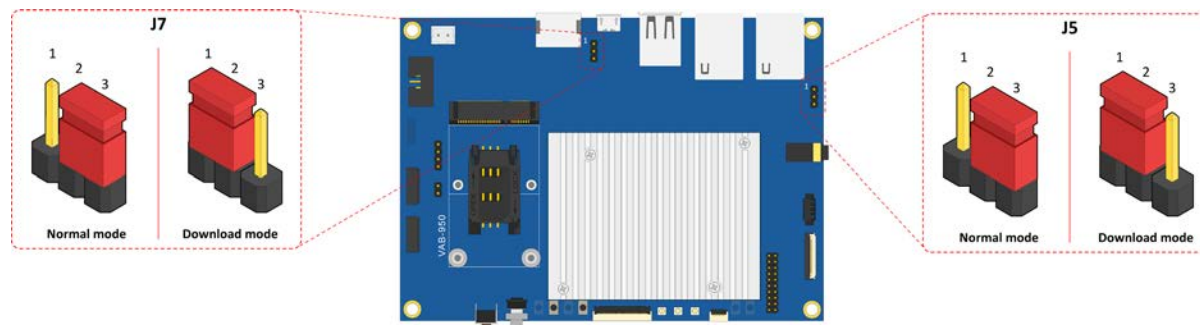
```

If you have already installed the image, use the following command to enter the Fastboot Mode on your Windows 10 host machine.

```
E:\platform-tools>adb reboot bootloader
```

Unplug the AC-to-DC power adapter to power off the VIA ARTiGO A950.

Set the two OS image jumpers (J7 and J5) back to the normal mode setting.



OS image jumpers diagram

Unplug the Micro USB cable, press the Power Button for 2 seconds, then release it to power on the VIA ARTiGO A950.

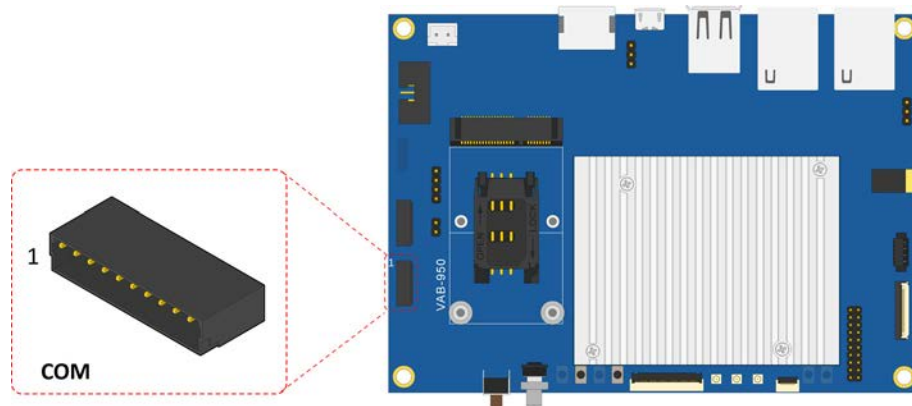
When the boot process has completed, you will see the Android 10.0 desktop.

3. Hardware Functions

This section explains how to enable and test the hardware functions precompiled in the Android 10.0 EVK including using the debug console.

3.1 Using the Debug Console

The first step is to connect the host machine and the VIA ARTiGO A950 through the COM connector labeled as "COM". Use a serial port communication program such as PuTTY or Tera Term to connect the debug console. Set the console Baud Rate to "921600".



COM connector diagram

Next, power on the VIA ARTiGO A950 to initiate the boot process.

When the boot process has completed, log in to the debug console.

3.2 Changing the Kernel Debug Level

To disable the kernel messages, modify the debug level using the following command:

```
console:/ # su
console:/ # echo 3 > /proc/sys/kernel/printk
```

3.3 Checking the BSP Version

To check the BSP version, use the following command:

```
console:/ # cat /proc/version
```

3.4 DVFS

To verify the DVFS (Dynamic Voltage Frequency Scaling) function and list all the supported features, use the following commands:

```
console:/ # ls -l /sys/devices/system/cpu/cpu0/cpufreq/
total 0
-r--r--r-- 1 root root 4096 2020-04-06 06:34 affected_cpus
-r----- 1 root root 4096 2020-04-06 06:34 cpuinfo_cur_freq
-r--r--r-- 1 root root 4096 2020-04-06 06:34 cpuinfo_max_freq
-r--r--r-- 1 root root 4096 2020-04-06 06:34 cpuinfo_min_freq
-r--r--r-- 1 root root 4096 2020-04-06 06:34 cpuinfo_transition_latency
```

```
-r--r--r-- 1 root root 4096 2020-04-06 06:34 related_cpus
-r--r--r-- 1 root root 4096 2020-04-06 06:34 scaling_available_frequencies
-r--r--r-- 1 root root 4096 2020-04-06 06:34 scaling_available_governors
-r--r--r-- 1 root root 4096 2020-04-06 06:34 scaling_cur_freq
-r--r--r-- 1 root root 4096 2020-04-06 06:34 scaling_driver
-rw-rw---- 1 system system 4096 2010-01-01 00:00 scaling_governor
-rw-rw---- 1 system system 4096 2010-01-01 00:00 scaling_max_freq
-rw-rw-r-- 1 system system 4096 2010-01-01 00:00 scaling_min_freq
-rw-r--r-- 1 root root 4096 2020-04-06 06:34 scaling_setspeed
console:/ #
```

To check the supported and current CPU frequency, use the following commands:

```
:~# cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
1989000 1924000 1846000 1781000 1716000 1677000 1625000 1586000 1508000 1417000 1326000
1248000 1131000 1014000 910000 793000
:~# cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq
793000
```

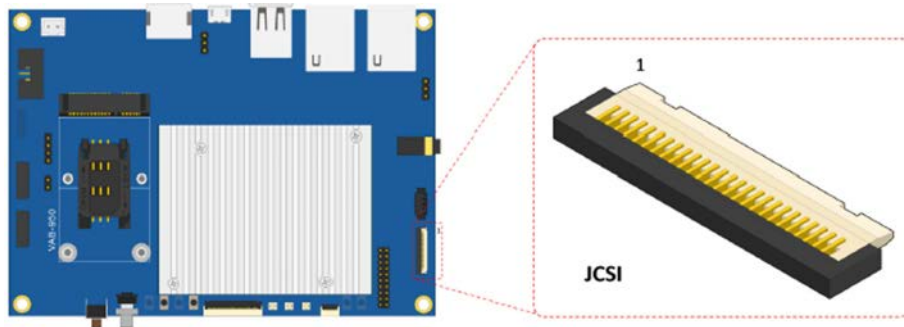
3.5 Display

There is no need to set the display device. If you connect an HDMI display, HDMI output will be automatically enabled after booting. If you connect an LCD panel, LCD panel output will be automatically enabled after booting.

3.6 MTK NeuroPilot AI APU Hardware Acceleration

We recommend that you install the third-party APK "image_classification_MobileNet_app-debug.apk" for verification.

Connect the CSI camera before booting the VIA ARTiGO A950 after image installation.



MIPI CSI connector diagram

To enlarge the logcat buffer to 64MB, use the following command:

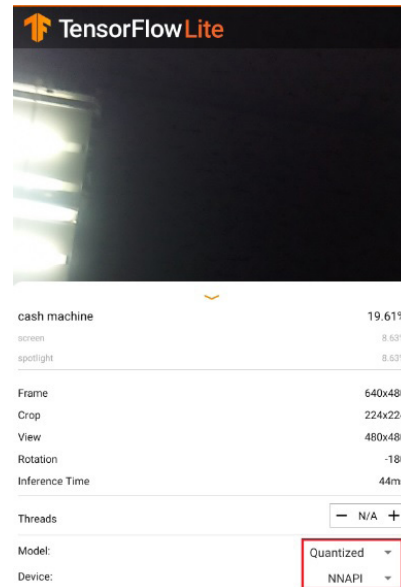
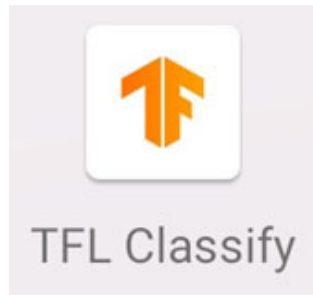
```
console:/ # logcat -G 64M
```

To enable the NeuroPilot relative log for getting the NN relative log in logcat, use the following command:

```
console:/ # su
console:/ # setprop debug.nn.vlog 1
console:/ # setprop debug.neuropilot.apu_nn.log 1
console:/ # setprop debug.neuropilot.gpu_nn.log 1
console:/ # setprop debug.neuropilot.arm_nn.log 1
```

Run the third-party APK "image_classification_MobileNet_app-debug.apk".

Then choose "Model: Quantized" and "Device: NNAPI".



To check that the NeuroPilot AI APU hardware acceleration is enabled, use the following command:

```
console:/ # logcat | grep findBestDeviceForEachOperation
11-05 04:31:15.502 3728 6893 I ExecutionPlan:
ModelBuilder::findBestDeviceForEachOperation(CONV_2D) = 0 (apunn)
11-05 04:31:15.502 3728 6893 I ExecutionPlan:
ModelBuilder::findBestDeviceForEachOperation(DEPTHWISE_CONV_2D) = 0 (apunn)
11-05 04:31:15.502 3728 6893 I ExecutionPlan:
ModelBuilder::findBestDeviceForEachOperation(CONV_2D) = 0 (apunn)
11-05 04:31:15.502 3728 6893 I ExecutionPlan:
ModelBuilder::findBestDeviceForEachOperation(AVERAGE_POOL_2D) = 0 (apunn)
11-05 04:31:15.502 3728 6893 I ExecutionPlan:
ModelBuilder::findBestDeviceForEachOperation(CONV_2D) = 0 (apunn)
11-05 04:31:15.502 3728 6893 I ExecutionPlan:
ModelBuilder::findBestDeviceForEachOperation(RESHAPE) = 0 (apunn)
11-05 04:31:15.503 3728 6893 I ExecutionPlan:
ModelBuilder::findBestDeviceForEachOperation(SOFTMAX) = 0 (apunn)
```


Note:

When the value 0 is displayed, it means that the APU hardware acceleration is enabled.



Taiwan Headquarters

1F, 531 Zhong-zheng Road,
Xindian Dist., New Taipei City 231
Taiwan

Tel: 886-2-2218-5452
Fax: 886-2-2218-9860
Email: embedded@via.com.tw



USA

940 Mission Court
Fremont, CA 94539,
USA

Tel: 1-510-687-4688
Fax: 1-510-687-4654
Email: embedded@viatech.com



Japan

3-15-7 Ebisu MT Bldg. 6F,
Higashi, Shibuya-ku
Tokyo 150-0011
Japan

Tel: 81-3-5466-1637
Fax: 81-3-5466-1638
Email: embedded@viatech.co.jp



China

Tsinghua Science Park Bldg. 7
No. 1 Zongguancun East Road,
Haidian Dist., Beijing, 100084
China

Tel: 86-10-59852288
Fax: 86-10-59852299
Email: embedded@viatech.com.cn



Europe

Email: embedded@via-tech.eu