

# Has Windows 11 IoT Enterprise on ARM Finally Arrived?

By Sean D. Liming and John R. Malin

Annabooks – [www.annabooks.com](http://www.annabooks.com)

August 31, 2026

OS: Windows 11 IoT Enterprise LTSC 2024 for ARM64

Target Platform: Radxa® Dragon Q6A with Qualcomm Dragonwing™ QCS6490

Support for Windows IoT Enterprise running on an ARM platform has been a long time in coming. Windows IoT Enterprise first supported the NXP i.MX 8 processor, but the custom Windows build process was overly complicated, and the performance was too slow. The Qualcomm Snapdragon was selected to run on Surface tablets. Qualcomm has been expanding its product lineup with the Dragonwing series to support embedded systems, so adding Windows 11 IoT Enterprise support was not going to be difficult. With the chip shortage, boards featuring Dragonwing are slowly becoming available to test and run Windows 11, and one of the most interesting boards is the Radxa Dragon Q6A. Using the Dragon Q6A as the target, this paper demonstrates how to create a custom Windows 11 IoT Enterprise on ARM image, shares the results for power and performance, and discusses application development to access I/O.

## 1.1 Introduction to the Radxa Dragon Q6A

The Dragonwing QCS6490 was first made available with the Qualcomm RB3 Gen 2 Core Kit and the Advantech AOM-2721 Development Kit. Both kits support a Windows 11 build and can be used to develop applications. How to create a custom Windows 11 IoT Enterprise build was not obvious from the documentation.

The platform that caught my eye was the Radxa Dragon Q6A. The Q6A is a Raspberry Pi equivalent platform that supports different Linux distros and Windows 11. The design is open-source, so you can download the schematics and layout information. Best of all, the board supports UEFI firmware. Booting to install any Linux distro or Windows from a USB flash disk is just like installing an OS on any IA PC board.



|                     |                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| SoC                 | Qualcomm® QCS6490                                                                                                                     |
| RAM                 | 4GB / 6GB / 8GB / 12GB / 16GB LPDDR5                                                                                                  |
| Storage             | SPI Flash (bootloader)<br>eMMC Module: 16 / 32 / 64 GB<br>UFS Module: 128 / 256 / 512 GB<br>MicroSD card slot ×1<br>M.2 2230 NVMe SSD |
| 40-pin Header       | UART / I2C / SPI / GPIO / PWM                                                                                                         |
| Ethernet / Wireless | Gigabit Ethernet and Wi-Fi6                                                                                                           |
| USB                 | 1× USB 3.1 Type-A OTG<br>3× USB 2.0 Type-A HOST<br>1× USB 2.0 Type-C Power                                                            |

## 1.2 Windows IoT Enterprise on ARM Development Process for Q6A

Since there is a UEFI firmware on the board, the Windows 11 IoT Enterprise development process is the same process covered in the [Starter Guide for Windows® 11 IoT Enterprise](#) book for IA platforms. There are some little quirks because device drivers have not been signed. The following sections discuss the process to create a custom Windows 11 IoT Enterprise image.

What is needed:

- [Radxa Dragon Q6A](#) with 12GB or 16GB RAM and 128GB UFS or M.2 NVMe SSD
- Windows IoT Enterprise LTSC 2024 ARM 64 ISO
- System Image Manager (SIM) from the Windows ADK installed
- 32GB USB flash disk
- USB keyboard / mouse
- HDMI display
- Rufus (rufus.ie) for creating a bootable USB flash disk
- Dragon Q6A Windows driver package and Wi-Fi and Bluetooth Driver (<https://docs.radxa.com/en/dragon/q6a/other-system/windows#31-prepare-the-driver-package>)

### 1.2.1 Follow the Radxa Windows Instructions

Radxa has notes about disabling standby in the power management. I thought it best to walk through [Install Windows on ARM on Radxa Dragon Q6A](#) instructions to see Windows running. The only difference was that I followed my instructions to set up Rufus and copy the Windows installer files to the NTFS partition on the USB flash disk. Windows installed from the USB flash disk just like any PC platform. A custom power plan was then created to disable standby, and a security template was created to disable the password timeout. The power plan file and security INF were put on a USB flash disk to bring them over to the development system.

### 1.2.2 Setup for Windows on ARM development

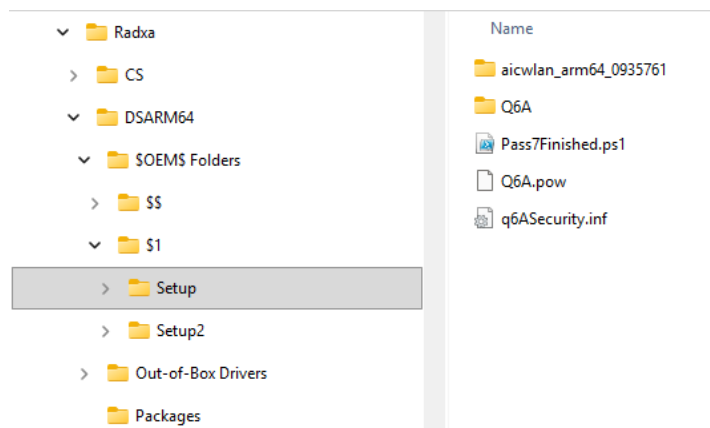
As outlined in the book, System Image Manager from the Windows ADK was used to create the custom image. The first step was to generate the catalog (.CLG) file from the Install.wim. The resulting name was “install\_Windows 11 IoT Enterprise LTSC 2024.clg” which is the same as IA x64. It might be a good idea to rename the file to “install\_Windows 11 IoT Enterprise LTSC 2024 Arm.clg” just to tell the files apart.

Following the steps in Chapter 5 of *Starter Guide for Windows® 11 IoT Enterprise*, the install.wim was updated with the latest cumulative updates. The latest updates can be found on the Microsoft Catalog Update site, using the following search string:

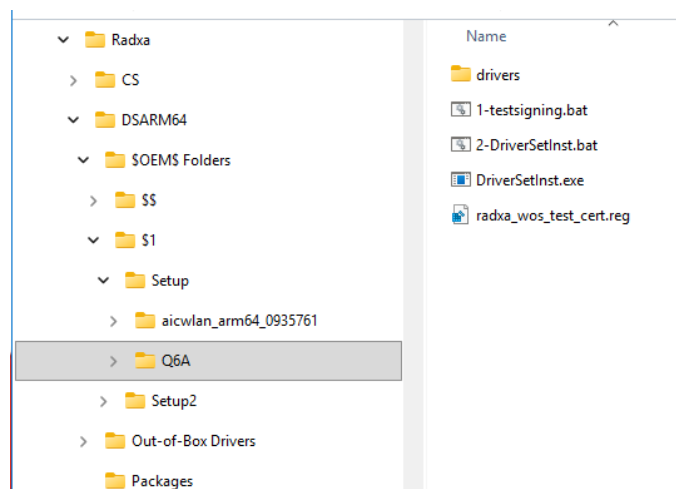
-Dynamic Cumulative Update for Windows 11 Version 24H2 for ARM64

Here is a link to the search results where you can download the security and .NET Framework cumulative updates: [Windows 11 IoT Enterprise Update for ARM64](#).

The final step was to create the distribution share and put all the files in place. The power plan file and the security INF get placed in the DSARM64\SOEM\$ Folders\1\Setup folder.

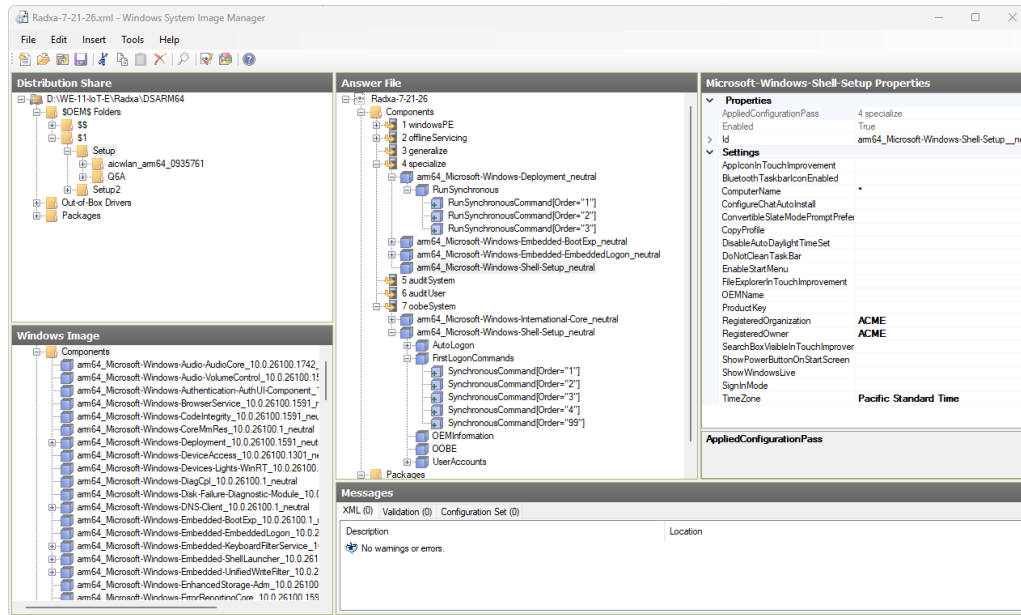


Since the drivers are not signed, the driver packages were placed in the \DSARM\SOEM\$ Folders\1\setup\Q6A folder. Synchronous commands were used to configure Windows for test mode and install the device drivers.



### 1.2.3 Create the Answer File

The next step was to create the answer file. The steps to create the answer file for ARM processors are the same for IA. The only difference is that the component names start with arm64 rather than amd64.



Here are the components and settings that went into the answer file.

| Component                                                 | Configuration Pass | Component Setting                                                                                                  | Possible Value / Notes                        |
|-----------------------------------------------------------|--------------------|--------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| arm64_Microsoft-Windows-International-Core-WinPE__neutral | 1 WindowsPE        | InputLocale                                                                                                        | en-US                                         |
|                                                           | 1 WindowsPE        | SystemLocale                                                                                                       | en-US                                         |
|                                                           | 1 WindowsPE        | UILanguage                                                                                                         | en-US                                         |
|                                                           | 1 WindowsPE        | UILanguageFallback                                                                                                 | en-US                                         |
|                                                           | 1 WindowsPE        | UserLocale                                                                                                         | en-US                                         |
| arm64_Microsoft-Windows-Setup__neutral                    | 1 WindowsPE        | UserData: AcceptEULA                                                                                               | True                                          |
|                                                           | 1 WindowsPE        | UserData : ProductKey : Key                                                                                        | <Enter the license test key from the OEM kit> |
|                                                           | 1 WindowsPE        | UserData : ProductKey: WillShowUI                                                                                  | OnError                                       |
|                                                           | 1 WindowsPE        | ImageInstall: OSImage : InstallFrom: MetaData: Key<br>(Right-click on "InstallFrom" to create the MetaData option) | /IMAGE/INDEX                                  |
|                                                           | 1 WindowsPE        | ImageInstall: OSImage : InstallFrom: MetaData: Value                                                               | 2                                             |
| arm64_Microsoft-Windows-LUA-Settings__neutral             | 2 offlineServices  | EnableLUA                                                                                                          | false                                         |
| arm64_Microsoft-Windows-Embedded-BootExp                  | 4 Specialize       | HideBootLogo                                                                                                       | 1                                             |
|                                                           | 4 Specialize       | HideBootStatusMessage                                                                                              | 1                                             |
| arm64_Microsoft-Windows-Embedded-EmbeddedLogon__neutral   | 4 Specialize       | HideAutoLogonUI                                                                                                    | 1                                             |
|                                                           | 4 Specialize       | HideFistLogonAnimation                                                                                             | 0                                             |
|                                                           | 4 Specialize       | NoLockScreen                                                                                                       | 1                                             |
| arm64_Microsoft-Windows-Shell-Setup__neutral              | 4 Specialize       | ComputerName                                                                                                       | *                                             |
|                                                           | 4 Specialize       | RegisteredOrganization                                                                                             | ACME                                          |
|                                                           | 4 Specialize       | RegisteredOwner                                                                                                    | ACME                                          |
|                                                           | 4 Specialize       | TimeZone                                                                                                           | Pacific Standard Time                         |
| arm64_Microsoft-Windows-International-Core__neutral       | 7 OOBE System      | InputLocale                                                                                                        | en-US                                         |
|                                                           | 7 OOBE System      | SystemLocale                                                                                                       | en-US                                         |
|                                                           | 7 OOBE System      | UILanguage                                                                                                         | en-US                                         |
|                                                           | 7 OOBE System      | UserLocale                                                                                                         | en-US                                         |
| arm64_Microsoft-Windows-Shell-Setup__neutral              | 7 OOBE System      | AutoLogon: Username                                                                                                | User1                                         |
|                                                           | 7 OOBE System      | AutoLogon: Enabled                                                                                                 | True                                          |
|                                                           | 7 OOBE System      | AutoLogon: LogonCount                                                                                              | 4294967294                                    |
|                                                           | 7 OOBE System      | AutoLogon: Password: Value                                                                                         | password                                      |
|                                                           | 7 OOBE System      | OEMInformation : Manufacture                                                                                       | ACME                                          |
|                                                           | 7 OOBE System      | OEMInformation : Model                                                                                             | ARM1                                          |
|                                                           | 7 OOBE System      | OOBE : HideEULAPage                                                                                                | True                                          |
|                                                           | 7 OOBE System      | OOBE : HideLocalAccountScreen                                                                                      | True                                          |
|                                                           | 7 OOBE System      | OOBE : HideOEMRegistrationScreen                                                                                   | True                                          |
|                                                           | 7 OOBE System      | OOBE : HideOnlineAccountScreen                                                                                     | True                                          |
|                                                           | 7 OOBE System      | OOBE : HideWirelessSetupInOOBE                                                                                     | True                                          |
|                                                           | 7 OOBE System      | OOBE : ProtectYourPC                                                                                               | 3                                             |

| Component | Configuration Pass | Component Setting                                                                                        | Possible Value / Notes |
|-----------|--------------------|----------------------------------------------------------------------------------------------------------|------------------------|
|           | 7 OOBE System      | OOBE :<br>UnattendEnabledREtailD<br>Emo                                                                  | false                  |
|           | 7 OOBE System      | UserAccounts :<br>LocalAccounts :<br>LocalAccount : Name<br>(Right-click to insert new<br>local account) | User1                  |
|           | 7 OOBE System      | UserAccounts :<br>LocalAccounts :<br>LocalAccount :<br>DisplayName                                       | User1                  |
|           | 7 OOBE System      | UserAccounts :<br>LocalAccounts :<br>LocalAccount :Group                                                 | Administrators         |
|           | 7 OOBE System      | UserAccounts :<br>LocalAccounts :<br>LocalAccount : Password<br>: Value                                  | password               |

The Windows Foundation package was added to enable the lockdown features.

| Package                                    | Windows Feature Selection | Value / Notes |
|--------------------------------------------|---------------------------|---------------|
| arm64_Microsoft-Windows-Foundation-Package | Client-DeviceLockdown     | Enabled       |
|                                            | Client-EmbeddedBootExp    | Enabled       |
|                                            | Client-EmbeddedLogon      | Enabled       |

Since the device drivers have not been signed, the synchronous commands were used to help put Windows into test mode and install the drivers. Radxa required a reboot after running the first batch file and installing the certificate .REG file. These first steps were be put into a Pass 4 Synchronous commands so the reboot can take place before Pass 7 Synchronous commands run the driver installation.

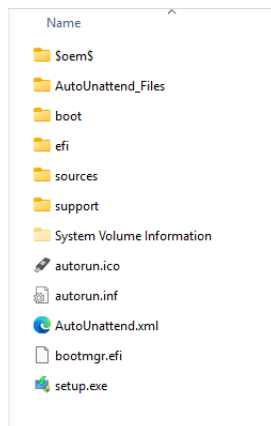
| No | Description                            | Pass 4 Sync Command                                              |
|----|----------------------------------------|------------------------------------------------------------------|
| 1  | Enable Test signing for device drivers | <code>bcdedit /set "{default}" testsigning on</code>             |
| 2  | Disable integrity checks               | <code>bcdedit /set "{default}" nointegritychecks on</code>       |
| 3  | Install the test certificates          | <code>regedit.exe /S c:\setup\Q6A\radxa_wos_test_cert.reg</code> |

The Pass 7 synchronous commands will install the power plan, security template, and device drivers.

| No | Description                                                       | Pass 7 Sync Command                                                                        |
|----|-------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| 1  | Import the power plan file                                        | <code>powercfg.exe /IMPORT c:\setup\Q6A.pow 88f4870f-c158-4980-8a5d-f46f1205f6cb</code>    |
| 2  | Set the power plan as the default plan                            | <code>powercfg.exe /S 88f4870f-c158-4980-8a5d-f46f1205f6cb</code>                          |
| 3  | Install the security template that disables the password timeout. | <code>secedit.exe /configure /db temp.sdb /cfg c:\setup\q6ASecurity.inf</code>             |
| 4  | Call the batch file to install the device drivers                 | <code>cmd.exe /C c:\setup\Q6A\2-DriverSetInst.bat</code>                                   |
| 99 | Signal the user that Pass 7 sync commands are finished            | <code>powershell.exe -ExecutionPolicy RemoteSigned -File c:\setup\Pass7Finished.ps1</code> |

### 1.2.4 Install on the Target

The final step was to create the configuration set and add the configuration set to the Windows install files on the USB flash disk that was prepared using Rufus.



Plug the USB flash disk into the Q6A 3.1 USB port, boot the target, and hit F12. Select the USB flash disk from the boot selection. The system booted to the USB, and the disk drive setup screen appeared. Delete any and all partitions on the target drive, and with the empty drive selected, click Next. The installation takes some time to install. Once the system comes up to the Explorer Shell, the device drivers start to install. There was a dialog asking to accept the installation of the drivers. Once the drivers finished installing, the system rebooted. The final step was to manually install the wireless Bluetooth driver via Device Manager.

| System Information   |                             |                                                                            |
|----------------------|-----------------------------|----------------------------------------------------------------------------|
| File Edit View Help  |                             |                                                                            |
| System Summary       | Item                        | Value                                                                      |
| Hardware Resources   | OS Name                     | Microsoft Windows 11 IoT Enterprise LTSC                                   |
| Components           | Version                     | 10.0.26100 Build 26100                                                     |
| Software Environment | Other OS Description        | Not Available                                                              |
|                      | OS Manufacturer             | Microsoft Corporation                                                      |
|                      | System Name                 | ACMEACM-1UDL4MD                                                            |
|                      | System Manufacturer         | Radxa Computer Co., Ltd.                                                   |
|                      | System Model                | Radxa Dragon Q6A                                                           |
|                      | System Type                 | ARM64-based PC                                                             |
|                      | System SKU                  | Q6A                                                                        |
|                      | Processor                   | Dragonwing(TM) QCS6490, 2707 Mhz, 8 Core(s), 8 Logical Processor(s)        |
|                      | BIOS Version/Date           | Qualcomm Technologies, Inc. 6.0.251230.BOOT.MXF.1.0.1-00549-KODIAKWP-1,... |
|                      | SMBIOS Version              | 3.3                                                                        |
|                      | Embedded Controller Version | 255.255                                                                    |
|                      | BIOS Mode                   | UEFI                                                                       |
|                      | BaseBoard Manufacturer      | Radxa Computer Co., Ltd.                                                   |
|                      | BaseBoard Product           | Radxa Dragon Q6A                                                           |
|                      | BaseBoard Version           | 1.0                                                                        |
|                      | Platform Role               | Desktop                                                                    |

There are some quirks with the system:

- The wireless doesn't work. The Radxa has notes about this. Wired Ethernet works just fine. Playing internet videos works without any problems.
- Double-boot – I noticed that the system starts to boot and then does another reboot and comes up fine. One time I noticed a watchdog timer issue, but it only happened one time. Again, the device drivers are not signed, so there is probably some more work to do.

## 1.3 Performance Qualcomm QCS6490 versus Intel Atom N100

There is the big question: “What is the performance?” Just interacting with the system, there are no noticeable performance issues. Using two different boot drive types, I did a boot time check and ran a Passmark test to do a comparison with an Intel Atom N100 platform.

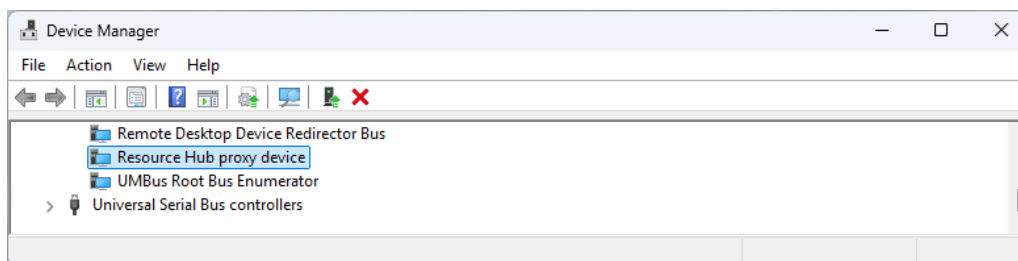
| System                            | Boot Time    | Passmark Score |
|-----------------------------------|--------------|----------------|
| Q6A with 128 UFS                  | 67 seconds   | 971.7          |
| Q6A with 128GB M.2 NVMe SSD       | 60 seconds   | 926.1          |
| UP Squared 7100 N100 with eMMC    | 33.2 seconds | 967.0          |
| UP Squared 7100 N100 with M.2 SSD | 22.9 seconds | 1321.4         |

Because of the quirky double-boot, I estimated that the Q6A could possibly boot in 48 seconds. Passmark tests were not designed for Windows on ARM, but the results match what I was experiencing. The performance was way better than the NXP i.MX8 and comparable to the Intel Atom N100. The biggest difference was power consumption. The Q6A doesn't require a lot of power.

| System               | Power at Windows Idle |
|----------------------|-----------------------|
| Q6A                  | 4 watts               |
| UP Squared 7100 N100 | 17 watts              |

## 1.4 RHPROXY Supported for UART / I2C / SPI / GPIO

Like the Windows 10 running on Raspberry PI, Minnow Board Max, and UP boards with Intel Atom E3800 processor, the Q6A supports RHPROXY, which allows .NET 10 applications to be written that can access the UART, I2C, SPI, and GPIO on the 40 Pin header.



.NET 10 applications can be written to access the UART, I2C, SPI, and GPIO on the 40 Pin header. The trick is to set the Target OS and Supported OS version to a newer Windows version in order to get access to the Windows.Devices namespace. You can also limit the output to the ARM64 processor. More content on low-powered bus support will be covered in future publications.

### Application

General



Output type

Specifies the type of application to build.

Windows Application



Target framework

Specifies the version of .NET that the application targets. This option can have different values depending on which versions of .NET are installed on your computer.

.NET 10.0

[Install other frameworks](#)



Target OS

Specifies the operating system that this project will target.

Windows



Target OS version

Specifies the version of the operating system this project will target.

10.0.22621.0



Supported OS version

Specifies the minimum OS version that the project will run on. When unspecified, the target OS version value is implied. Using an earlier version here requires code to add guards around later version APIs.

10.0.22621.0



Platform target

Specifies the processor to be targeted by the output file. Choose 'Any CPU' to specify that any processor is acceptable, allowing the application to run on the broadest range of hardware.

ARM64

## 1.5 A New Option for PC Board Manufacturers

PC board manufacturers are already familiar with UEFI firmware. Creating a QCS6490 board that can be a lower-powered option would fit perfectly in any product lineup. No need for clients to learn a completely new OS development process, since the system looks and feels like a regular PC.

## 1.6 Summary: Windows on ARM is Here

The Q6A platform operates no differently than any other Intel PC platform. You wouldn't really know that there was an ARM processor onboard until you look at the settings. The performance is close to the Intel Atom but with lower power consumption. Best of all, since the Q6A has a UEFI firmware that can boot Windows Installer, the same development process called out in *Starter Guide for Windows® 11 IoT Enterprise* can be used to create a custom Windows 11 IoT Enterprise image. No learning curve to be overcome.

It took some time, but Windows on ARM has arrived. Radxa really did their homework to develop a great little platform. The real credit goes to Qualcomm and Microsoft, who made Windows IoT Enterprise on ARM finally happen. Qualcomm saw the opportunity to expand beyond Surface tablets with more performance and power-efficient ARM solutions.

Windows is a registered trademark of Microsoft Corporation  
All other copyrighted, registered, and trademarked material remains the property of the respective owners,