

## Windows® 10 IoT Core: to VM or not to VM?

By Sean D. Liming and John R. Malin  
Annabooks – [www.annabooks.com](http://www.annabooks.com)

July 2018

Microsoft has worked with several silicon and board manufacturers to get Windows 10 IoT Core running on different silicon and platforms such as MinnowBoard Turbot/Max, Raspberry Pi2/3, NXP, DragonBoard 410c, etc. There have been questions on the Forum about running IoT Core in a Virtual Machine (VM). An obvious question since Windows 10 Enterprise easily runs on a VM. IoT Core is intended for single function small devices that interact with the low-speed bus features of a microcontroller (GPIO, SPI, I2C, UART), thus having hardware is usually required. The only scenario for using a VM that makes sense is when you don't have access to hardware, for example, flying at 30,000 ft (9,144 meters) for several hours, and you just want to test the application you have been working on. Taking out strange hardware to work with might be a concern to the flight crew, other passengers, and any air marshals who might be onboard. There might be other scenarios, but in most cases using real-hardware for development is best if not mandatory. The question does remain, can IoT Core run in a VM? This paper looks to answer that basic question.

### UEFI is the Key: Setup and Installation

The key to getting IoT Core up and running on any Intel Architecture platform is UEFI support in the firmware. Virtual Machines are catching up and adding pure UEFI boot support options, thus we need a solution that boots via pure UEFI to install the IoT Core FFU image.

The first thing we need is the IoT Core FFU image.

1. Open your Internet browser and go to [Windows IoT Core site](#).
2. Click on Downloads.
3. Scroll down and select a pre-built image "For MinnowBoard Turbot/MAX". This will download an ISO image.
4. Mount the ISO image and run the installer. The FFU image will be installed to C:\Program Files (x86)\Microsoft IoT\FFU\MinnowBoardMax\_x64. This is a 64-bit image, so the VM will need to be 64-bit.

Next, we will use a WinPE image to help us install the FFU into the VM.

1. In your Internet browser go to the [ADK download site](#).
2. Download the last ADK.
3. Run the installer and choose the defaults. Be patient. This will take some time.
4. Once the ADK has been installed, click on the Start button and go to the Windows Kits->Deployment and Imaging Tools Environment. This will open a command prompt.
5. Run the following command to create a 64-bit WinPE image:

```
copy c:\WinPE_IoTCore64
```

**Warning:** the destination directory should NOT exist before running the command.

Note: The final destination directory is up to you. I used e:\WinPE-Windows10.

6. This will create the destination directory with three folders underneath: fwfiles, media, and mount. Copy efisys.bin in the \fwfiles folder to the \media\boot folder. The efisys.bin file is the bootloader for booting a WinPE ISO image in a pure UEFI system.
7. Copy the flash.FFU file from C:\Program Files (x86)\Microsoft IoT \FFU\MinnowBoardMax\_x64 to the root of \media.

- In the root of \media, create a batch file called FFU-install.bat with the following:

```

Dism.exe /Apply-Image /ImageFile:"d:\Flash.ffu" /ApplyDrive:\\.\PhysicalDrive0
/SkipPlatformCheck

```

The batch file calls DISM to install the FFU onto the virtual disk. You can use the same technique to install on any x86/amd64 system.

- Outside the WinPE folder, create another batch file called CreateISO.bat with the following to run the oscdimg.exe utility:

```

"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment
Tools\amd64\Oscdimg\oscdimg.exe" -u2 -udfver102 -pEF -h -o -b"c:\WinPE_IoTCore64
\media\Boot\efisys.bin" "c:\WinPE_IoTCore64\media"
"c:\WinPE_IoTCore64\media\W10IoTcore.iso"

```

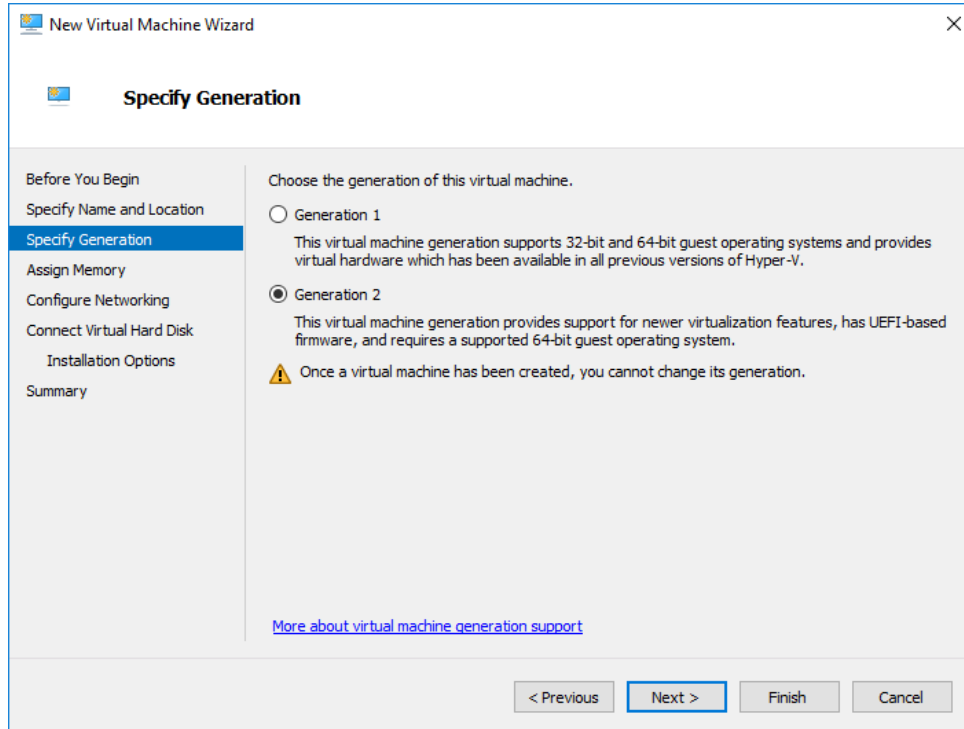
The options for the above command creates an ISO image with UDF file system (-u2) that boots on a system with UEFI (-pEF and -b efisys.bin). More information on the oscdimg.exe command line options can be found [here](#).

- Run the CreateIOS.bat file. The final W10lotcore.ISO will be in the root of \media

The final step is to create a VM that supports UEFI boot.

### Installation for Hyper-V:

- Create a new virtual machine, and when you get to the point in the Wizard to select Generation 1 or Generation 2, select Generation 2. The VM will have UEFI boot support.

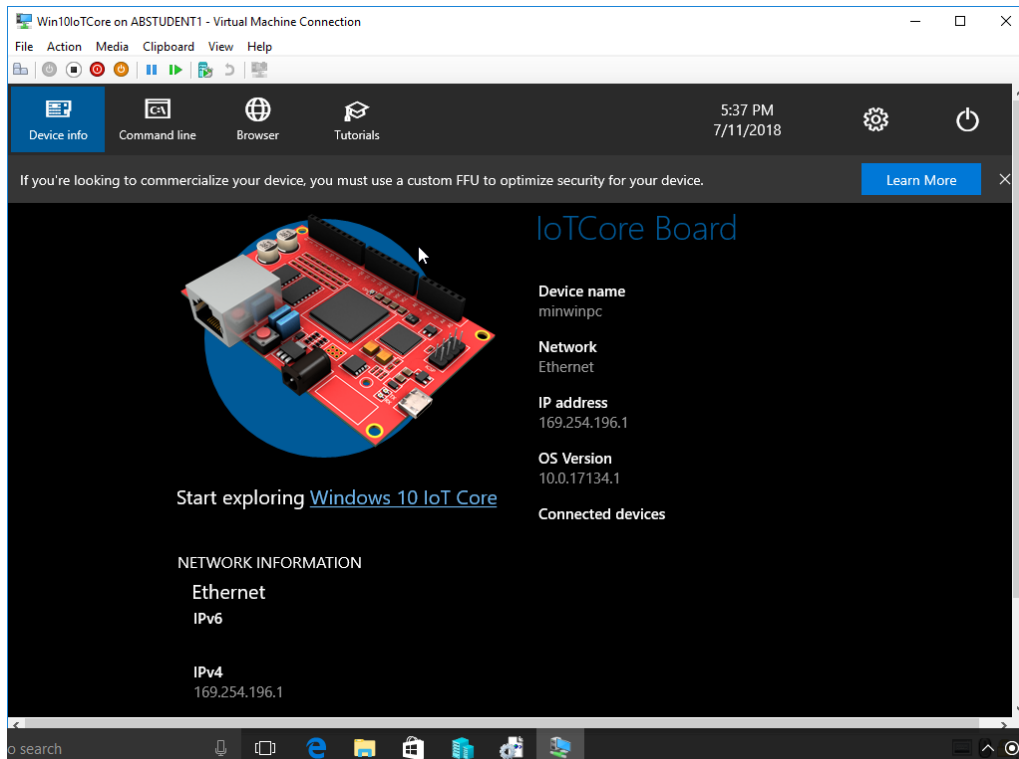


- Set the RAM memory to 2GB and the hard drive should be made fixed for 15GB.
- Attach the W10lotcore.ISO as the DVD to the VM and boot the VM. The system should boot to WinPE in the ISO.

4. In the command window, change to the D drive, and run the FFU-install.bat. This will install the flash.FFU on the VM drive.
5. Once the installation is complete, shut down the system by running the following command:

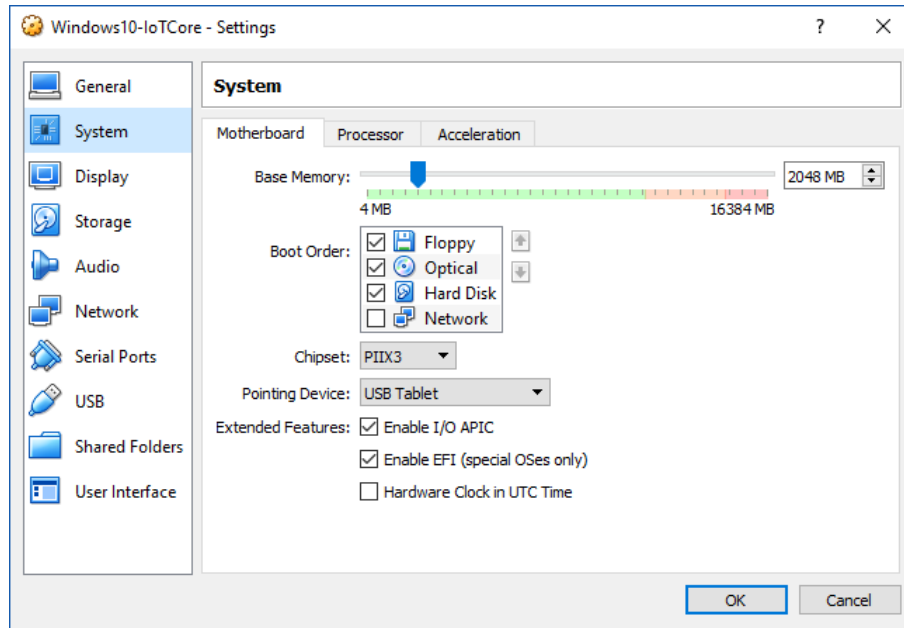
```
wpeutil shutdown
```

6. Remove the ISO from the VM.
7. Start the VM, and the image will eventually boot to the IoT Core image:

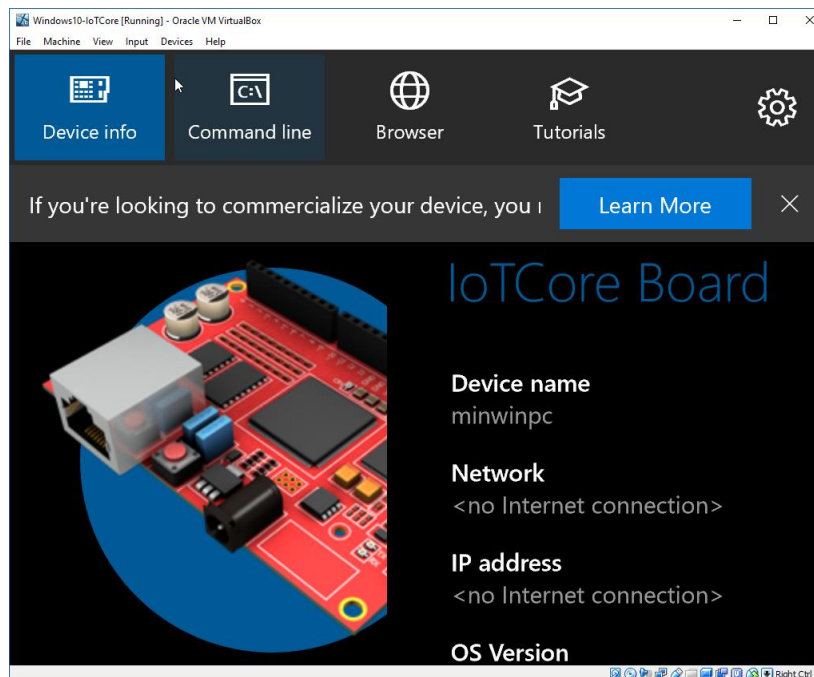


### Installation for VirtualBox:

1. Create a new Windows 10 64-bit VM.
2. Set the RAM memory to 2GB and the hard drive should be made fixed for 15GB.
3. Once the VM has been created, go to Settings->System->Motherboard, and check "Enable EFI (special Oses only)". This will enable UEFI boot.



4. Attach the W10lotcore.ISO as the DVD to the VM and boot the VM. The system should boot to WinPE in the ISO.
5. In the command window, change to the D drive, and run the FFU-install.bat. This will install the flash.FFU on the VM drive.
6. Once the installation is complete, shut down the system by running the following command:  
`wpeutil shutdown`
7. Remove the ISO from the VM.
8. Start the VM, and the image will eventually boot to the IoT Core image:



### **The Items that are Missing**

The FFU image is a stock image designed for the MinnowBoard Turbot/Max, thus the image only has driver support for that platform. The Hyper-V VM has what appears to be a network driver loaded since an IP address is shown, but a default IP address is one that appears when there is no network connection. The VirtualBox VM has no IP address so there is no way to connect to the network. In both cases, IoT Dashboard will not find either VM on the network, nor will you be able to do remote Visual Studio development. We did make attempts to use the various VM network connection options, but there were no settings that made for a successful connection. Network device drivers are needed in the image. The VirtualBox VM has the additional issue of a missing video driver as the screen appears to be cut off.

Further investigation into the missing drivers is needed, and a custom amd64 image with missing device drivers will have to be created for the VM to be usable for development.

### **Summary: Focus on Using Real Hardware**

The answer to the questions is “Yes, IoT Core can run in a virtual machine”, but there are limitations around device driver support. The stock amd64 image for the MinnowBoard Turbot/Max is not enough to support remote development on a VM. Even if an image could be created, your applications will not be able to access low-speed bus ports as these ports are not available in either VM environment. Working with real hardware gives you the best development environment for developing for IoT Core.

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