NUC980/NUC970
Linux Environment on VMware
User Manual

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

NUC980/NUC970 LINUX ENVIRONMENT ON VMWARE USER MANUAL

Table of Contents

List of Figures

1 OVERVIEW

This document describes the Linux Environment Image which provides users with a packaged Linux platform to develop their applications on NUC980/NUC970 series EVBs. It is integrated with relevant kits of NUC980/NUC970 development environment, so users can skip the installation process, and only need to focus on programming applications. A Linux platform is needed to build Linux kernel, U-Boot, and applications using the Linux compiling toolchain. In this user manual, we take NuMaker NUC980 IIoT as the target board for example.

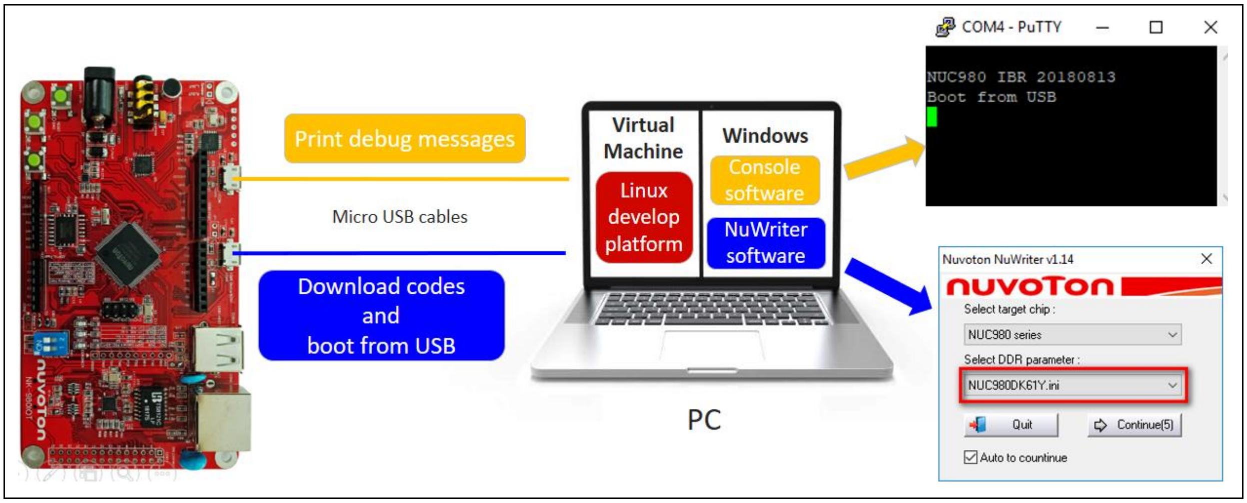
The Linux Environment Image includes the following contents:

- Linux platform: ubuntu-18.04.3-desktop-amd64.
- GCC 4.8.4 crLinuxs compiler with EABI support.
- uClibc-0.9.33
- Binutils-2.24
- Demo program for device drivers, busybox, mtd-util, and other open source applications.
- Linux 4.4 kernel source code and NUC980/ NUC970 device drivers.

U-Boot 2016.11 source code including NUC980 device drivers.

1.1 Development Environment

The Linux Image runs on a virtual machine. PC can communicate with NUC980/NUC970 series via UART, as well as debug port. The above interface can be used to load the binary file to EV board for execution or debugging. The USB interface can be used by NuWriter to program NAND, SPI, and eMMC. Figure1-1 is an example of the development environment.



```
NUC980 IBR 20180813
Boot from USB
```

Select target chip : NUC980 series

Select DDR parameter : NUC980DK61Y.ini

☑ Auto to countinue

Figure 1-1 Development Environment Setup

2 LINUX IMAGE INSTALLATION

2.1 System Requirement

The Linux Environment Image provides a Linux operating system running on VMware.

This chapter introduces how to install VMware virtual machine, and the steps to install the Linux Environment Image.

2.2 Download and Install VMware Virtual Machine

The VMware provides free virtual machine VMware Workstation Player 15.5 for users to download from VMware official website http://www.VMware.com/. User can select **Downloads** → "**Workstation Player**" and then click "**VMware Workstation 15.5 Player for Windows**". Please refer to Figure 2-1 and Figure 2-2.

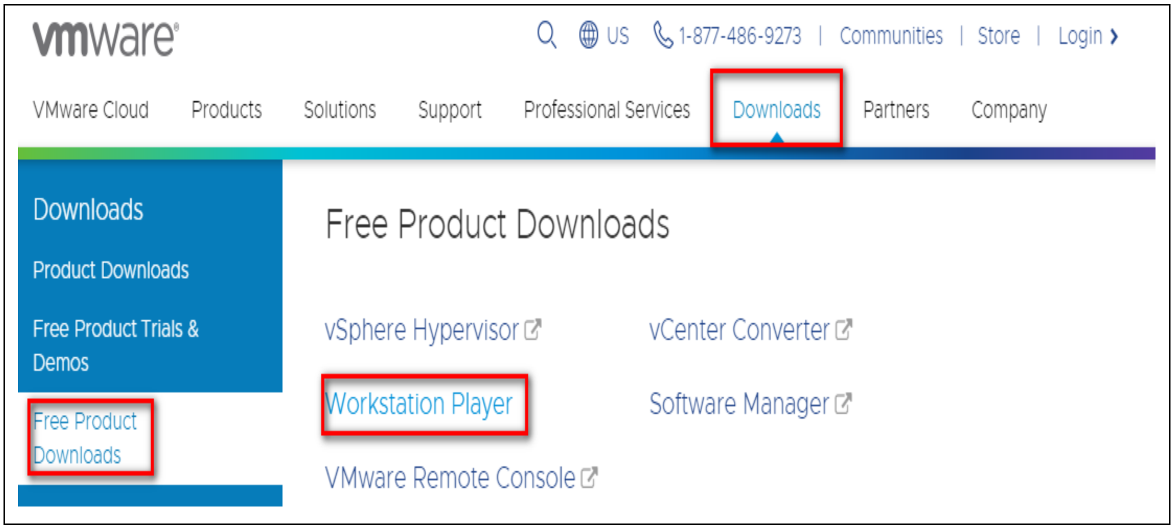
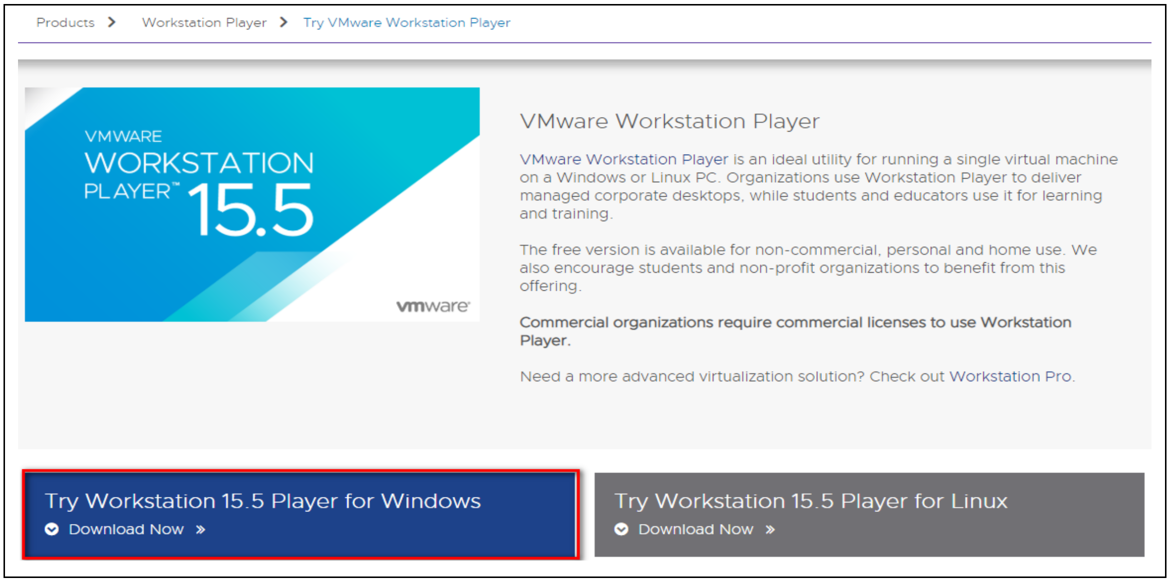


Figure 2-1 Download VMware Workstation Player



VMware Workstation Player

VMware Workstation Player is an ideal utility for running a single virtual machine on a Windows or Linux PC. Organizations use Workstation Player to deliver managed corporate desktops, while students and educators use it for learning and training.

The free version is available for non-commercial, personal and home use. We also encourage students and non-profit organizations to benefit from this offering.

Commercial organizations require commercial licenses to use Workstation Player.

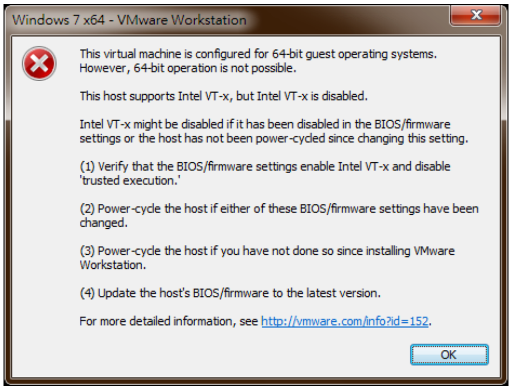
Need a more advanced virtualization solution? Check out Workstation Pro.

Try Workstation 15.5 Player for Windows
Download Now »
Try Workstation 15.5 Player for Linux
Download Now »

Figure 2-2 Download VMware Workstation Player for Windows

After the download is complete, following next steps to install VMware Workstation Player.

Note: Make sure to enable Virtualization of BIOS before installing VMware:



This virtual machine is configured for 64-bit guest operating systems. However, 64-bit operation is not possible.

This host supports Intel VT-x, but Intel VT-x is disabled.

Intel VT-x might be disabled if it has been disabled in the BIOS/firmware settings or the host has not been power-cycled since changing this setting.

(1) Verify that the BIOS/firmware settings enable Intel VT-x and disable 'trusted execution.'

(2) Power-cycle the host if either of these BIOS/firmware settings have been changed.

(3) Power-cycle the host if you have not done so since installing VMware Workstation.

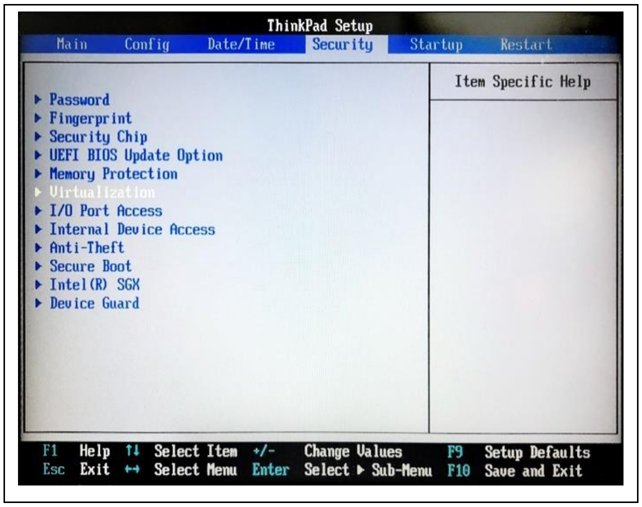
(4) Update the host's BIOS/firmware to the latest version.

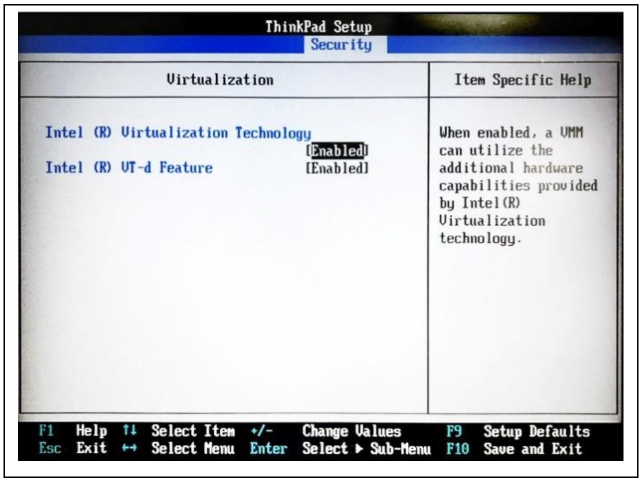
For more detailed information, see http://vmware.com/info?id=152.

Figure 2-3 VMware Installing Fault

Follow the procedure below to solve the problem:

- Enable Virtualization from BIOS before installing VMware, refer to Figure 2-4.
  1. Restart computer.
  2. Log in to the BIOS screen when booting.
  3. Select configuration, then select Intel virtual technology. The system is disabled by default.
  4. Change disabled to enabled.
  5. Save the settings and restart.
- There will be some differences depending on each BIOS.





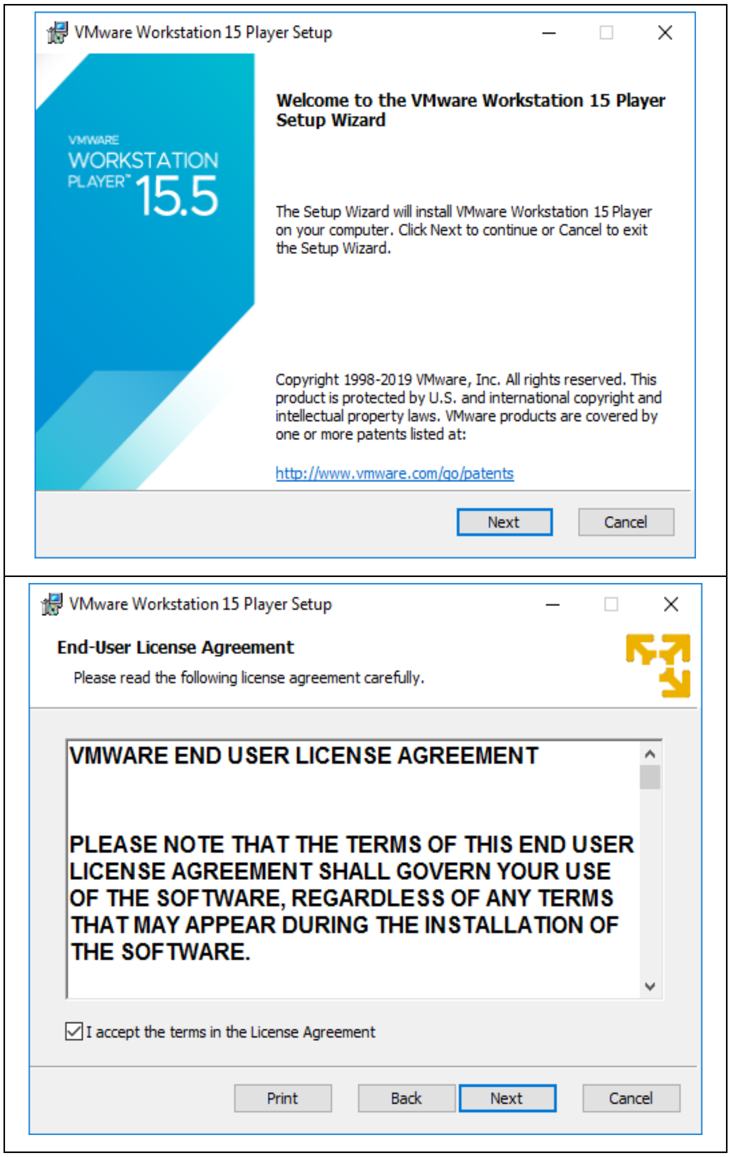
```
                              ThinkPad Setup
                                 Security

            Virtualization                      Item Specific Help

  Intel (R) Virtualization Technology          When enabled, a VMM
                                  [Enabled]    can utilize the
  Intel (R) VT-d Feature          [Enabled]    additional hardware
                                               capabilities provided
                                               by Intel(R)
                                               Virtualization
                                               technology.

 F1   Help  ↑↓  Select Item  +/-   Change Values        F9   Setup Defaults
 Esc  Exit  ↔   Select Menu  Enter Select ▸ Sub-Menu    F10  Save and Exit
```

Figure 2-4 BIOS Setting



VMware Workstation 15 Player Setup

Welcome to the VMware Workstation 15 Player Setup Wizard

The Setup Wizard will install VMware Workstation 15 Player on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Copyright 1998-2019 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at:

http://www.vmware.com/go/patents

VMware Workstation 15 Player Setup

End-User License Agreement

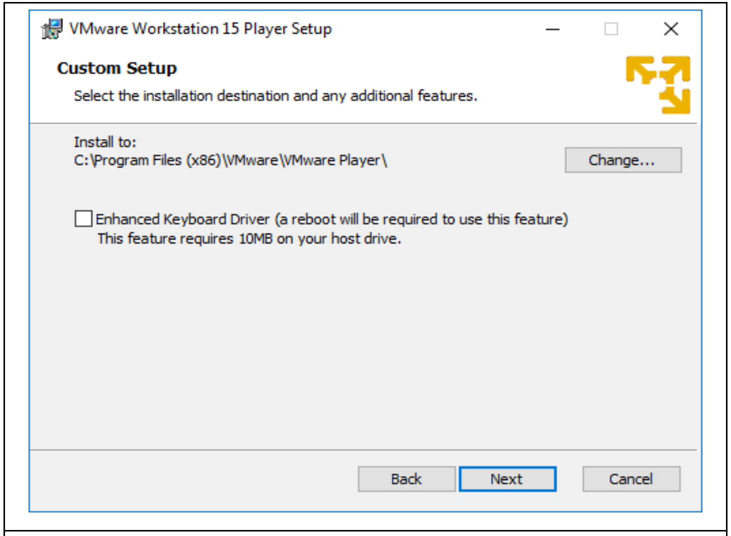
Please read the following license agreement carefully.

VMWARE END USER LICENSE AGREEMENT

PLEASE NOTE THAT THE TERMS OF THIS END USER LICENSE AGREEMENT SHALL GOVERN YOUR USE OF THE SOFTWARE, REGARDLESS OF ANY TERMS THAT MAY APPEAR DURING THE INSTALLATION OF THE SOFTWARE.

☑ I accept the terms in the License Agreement

NUC980/NUC970 LINUX ENVIRONMENT ON VMWARE USER MANUAL



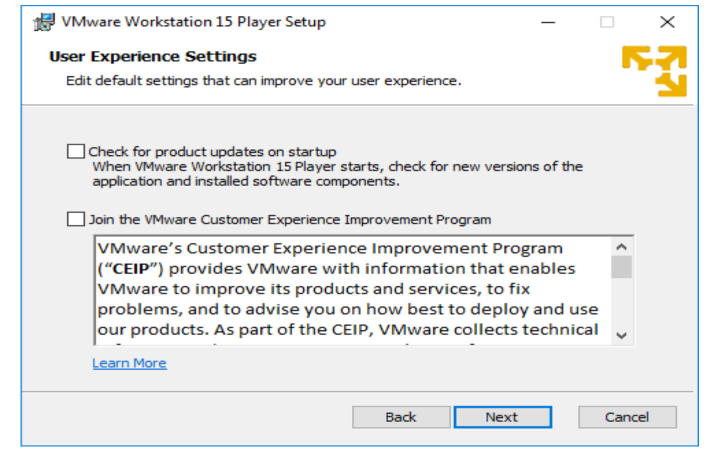
VMware Workstation 15 Player Setup

Custom Setup

Select the installation destination and any additional features.

Install to:
C:\Program Files (x86)\VMware\VMware Player\

□ Enhanced Keyboard Driver (a reboot will be required to use this feature)
This feature requires 10MB on your host drive.



VMware Workstation 15 Player Setup

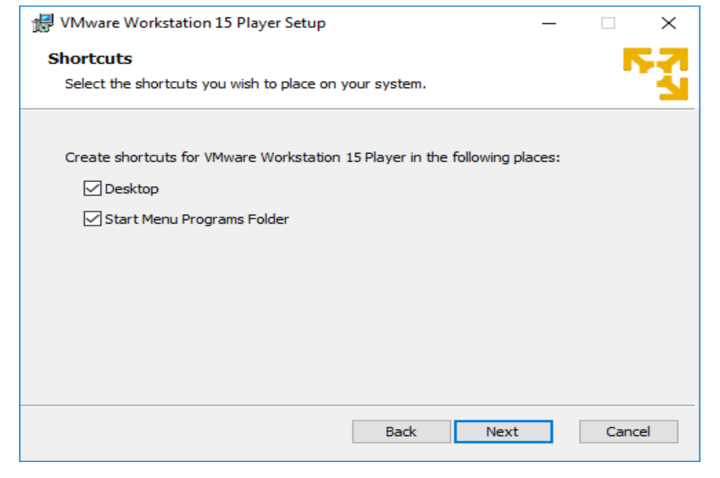
User Experience Settings

Edit default settings that can improve your user experience.

□ Check for product updates on startup
When VMware Workstation 15 Player starts, check for new versions of the application and installed software components.

□ Join the VMware Customer Experience Improvement Program

VMware's Customer Experience Improvement Program ("**CEIP**") provides VMware with information that enables VMware to improve its products and services, to fix problems, and to advise you on how best to deploy and use our products. As part of the CEIP, VMware collects technical



VMware Workstation 15 Player Setup

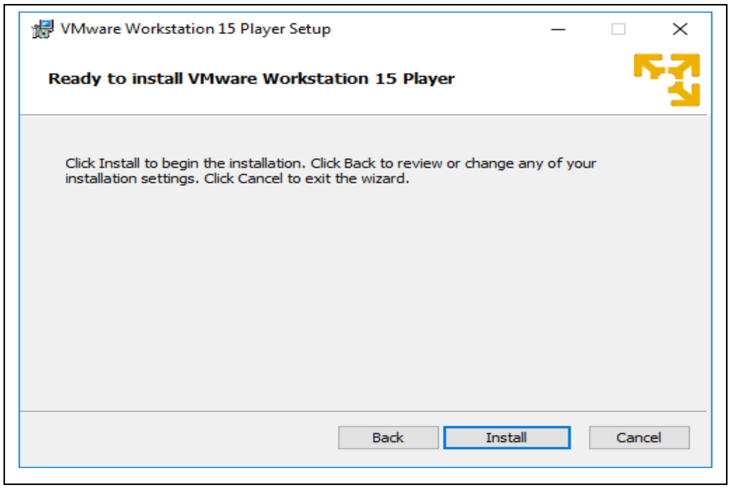
Shortcuts

Select the shortcuts you wish to place on your system.

Create shortcuts for VMware Workstation 15 Player in the following places:

☑ Desktop

☑ Start Menu Programs Folder



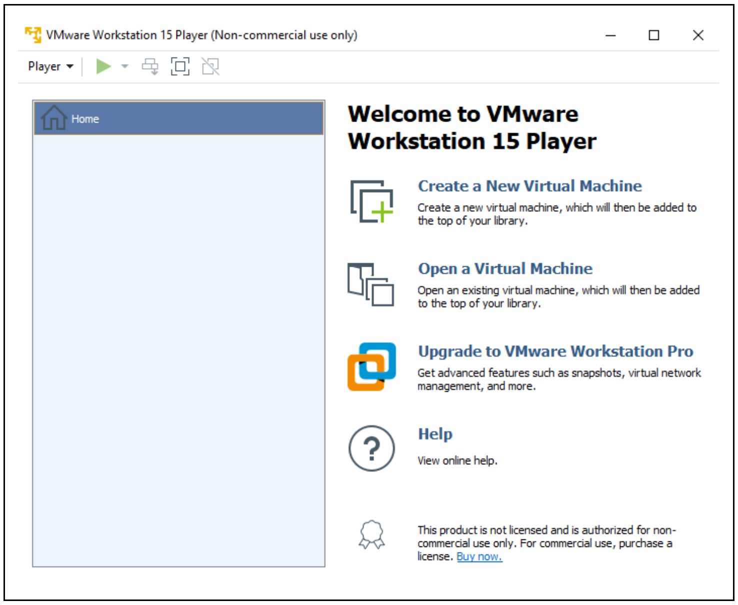
VMware Workstation 15 Player Setup

Ready to install VMware Workstation 15 Player

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

Figure 2-5 Install VMware Workstation Player

Open VMware Workstation 15 Player.



VMware Workstation 15 Player (Non-commercial use only)

Welcome to VMware Workstation 15 Player

Create a New Virtual Machine
Create a new virtual machine, which will then be added to the top of your library.

Open a Virtual Machine
Open an existing virtual machine, which will then be added to the top of your library.

Upgrade to VMware Workstation Pro
Get advanced features such as snapshots, virtual network management, and more.

Help
View online help.

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

Figure 2-6 Virtual Machine Menu

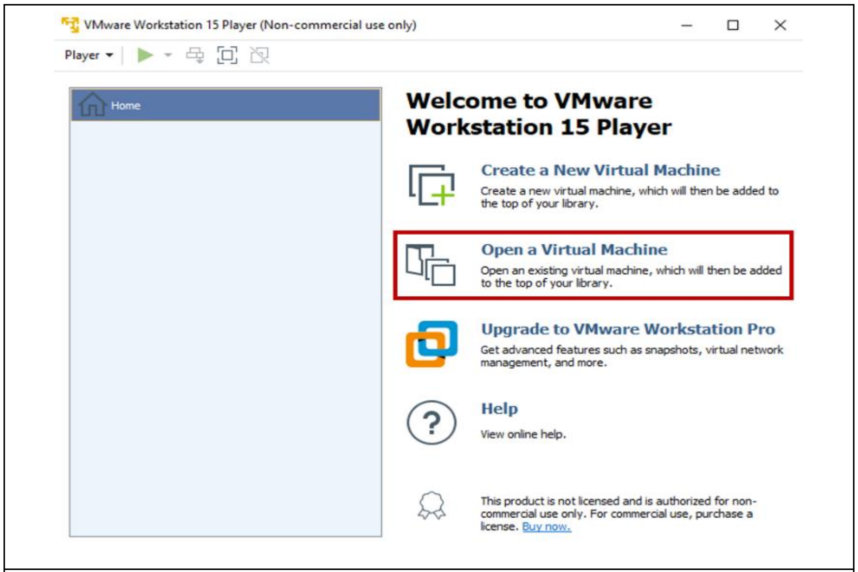
2.3 Download and Open Linux Environment Image

Download the image resource "**NUC980_970 Linux environment on VMware**" at the following URL:

https://www.nuvoton.com/products/microprocessors/arm9-mpus/nuc980-industrial-control-iot-series/?group=Software&tab=2

Decompress **ubuntu_NUC970_980_Linux.zip** to **ubuntu_NUC970_980_Linux** folder, then refer to Figure 2-7 and the procedure below to open Linux Environment Image.

1. Open a Virtual Machine
2. Select "Ubuntu 64-bit_nuvoton.vmx" under **ubuntu_NUC970_980_Linux** folder.
3. The Linux Environment Image is loaded now. Click "Play virtual machine".
4. You will see messages when opening a new image, refer to Figure 2-7 to click the options.



VMware Workstation 15 Player (Non-commercial use only)

Welcome to VMware Workstation 15 Player

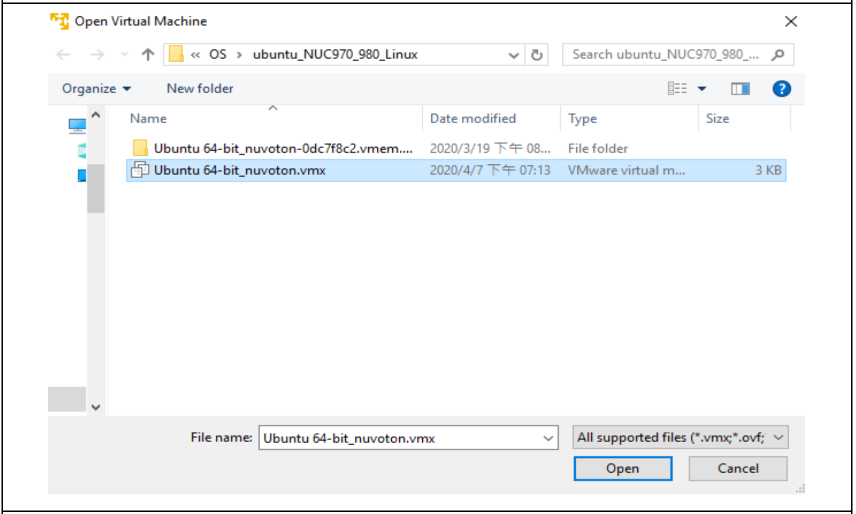
Create a New Virtual Machine
Create a new virtual machine, which will then be added to the top of your library.

Open a Virtual Machine
Open an existing virtual machine, which will then be added to the top of your library.

Upgrade to VMware Workstation Pro
Get advanced features such as snapshots, virtual network management, and more.

Help
View online help.

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

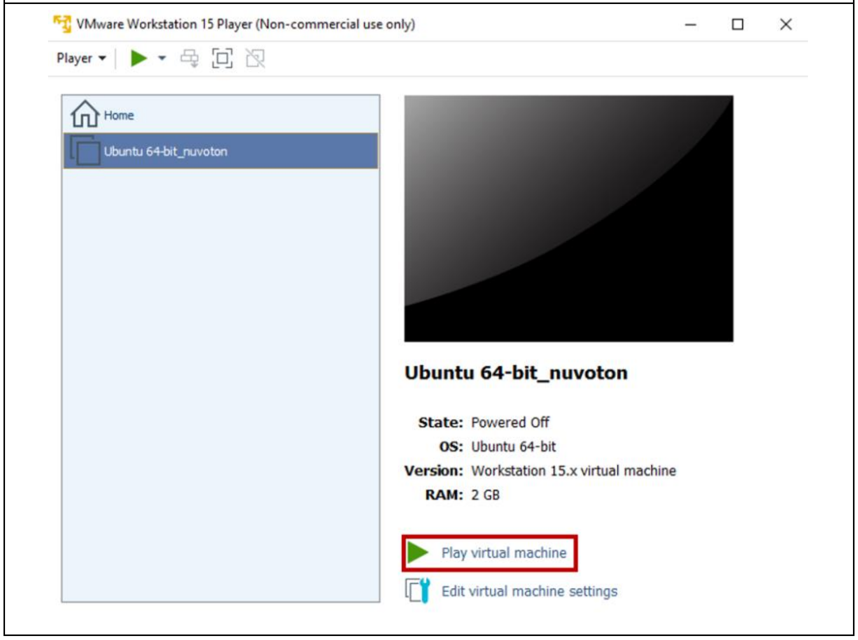


Open Virtual Machine

Name	Date modified	Type	Size
Ubuntu 64-bit_nuvoton-0dc7f8c2.vmem....	2020/3/19 下午 08...	File folder	
Ubuntu 64-bit_nuvoton.vmx	2020/4/7 下午 07:13	VMware virtual m...	3 KB

File name: Ubuntu 64-bit_nuvoton.vmx

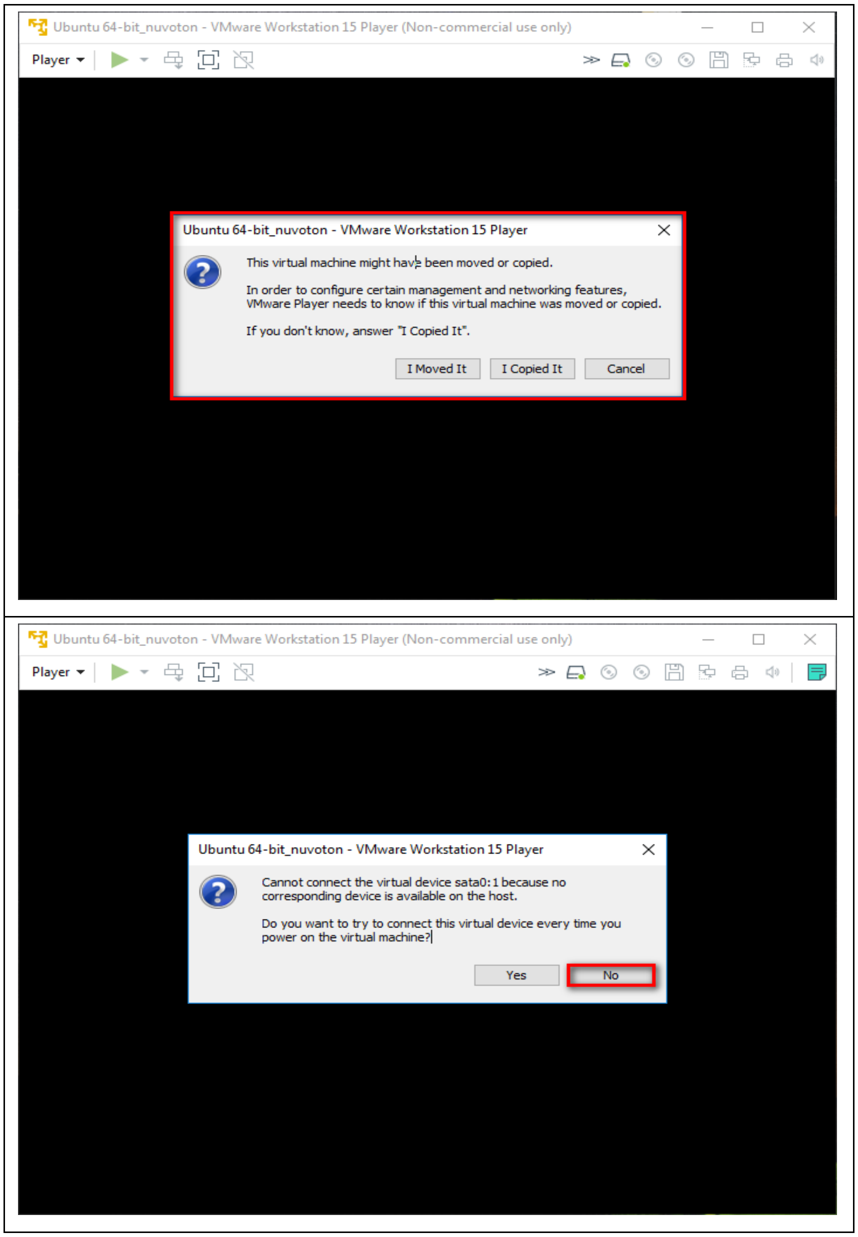
All supported files (*.vmx;*.ovf;



VMware Workstation 15 Player (Non-commercial use only)

Ubuntu 64-bit_nuvoton

State: Powered Off
OS: Ubuntu 64-bit
Version: Workstation 15.x virtual machine
RAM: 2 GB



Ubuntu 64-bit_nuvoton - VMware Workstation 15 Player

This virtual machine might have been moved or copied.

In order to configure certain management and networking features, VMware Player needs to know if this virtual machine was moved or copied.

If you don't know, answer "I Copied It".

Ubuntu 64-bit_nuvoton - VMware Workstation 15 Player

Cannot connect the virtual device sata0:1 because no corresponding device is available on the host.

Do you want to try to connect this virtual device every time you power on the virtual machine?

Ubuntu 64-bit_nuvoton - VMware Workstation 15 Player (Non-commercial use only)
Player
I Moved It
I Copied It
Cancel
Ubuntu 64-bit_nuvoton - VMware Workstation 15 Player (Non-commercial use only)
Player
Yes
No

Figure 2-7 Open Linux Environment Image

An Ubuntu login window will show up after installation is complete. Log in with the username "nuvoton" and the password is "user".

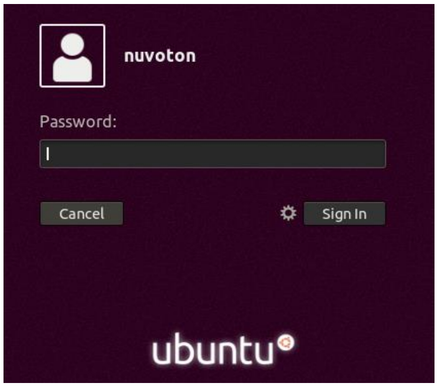
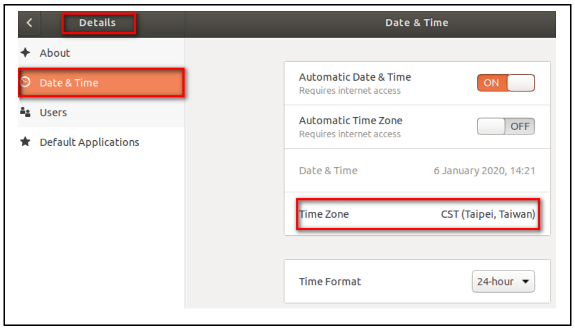


Figure 2-8 Login Linux Virtual Machine

Check if Time Zone is identical to your location.



Automatic Date & Time ON
Requires internet access

Automatic Time Zone OFF
Requires internet access

Date & Time 6 January 2020, 14:21

Time Zone CST (Taipei, Taiwan)

Time Format 24-hour

Figure 2-9 Time Zone Setting

Find **NUC970_Buildroot-master** folder under personal folder "Home".

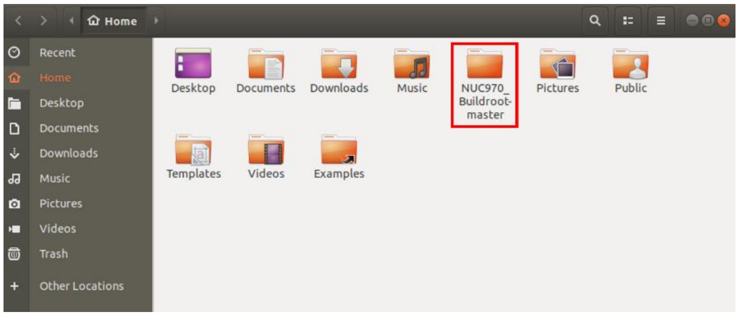


Figure 2-10 NUC970_Buildroot-master Folder

The **NUC970_Buildroot-master** folder contains everything needed to develop Linux kernel, uboot and applications.

Buildroot is a simple, efficient and easy-to-use tool, which is a set of Makefiles and patches that simplifies and automates the process of building a complete and bootable Linux environment for an embedded system through cross-compilation.

NUC980 and NUC970 use identical Linux kernel source code and device drivers. According to what EVB used, set the corresponding configuration file.

If users need to modify buildroot parameters, use command "`make menuconfig`" to set the configuration of buildroot.

Now, the Linux platform is set up successfully. The next section will introduce how to compile and generate executable files in Linux platform.

2.4 Update

If Nuvoton releases new patches on the Internet, follow this procedure to update Linux, uboot and buildroot.

Note: If it is your first time to set up the Linux platform, please update Linux, uboot and buildroot first.

Linux and uboot Update:

Delete original folders and *.gz* files of linux-master and uboot-mater in Linux platform at the location as shown below.

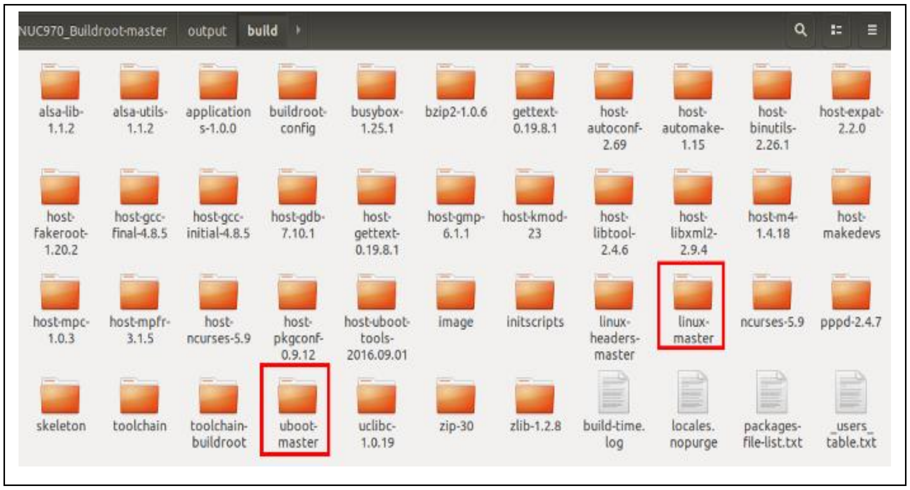


Figure 2-11 File location of linux-master and uboot-master

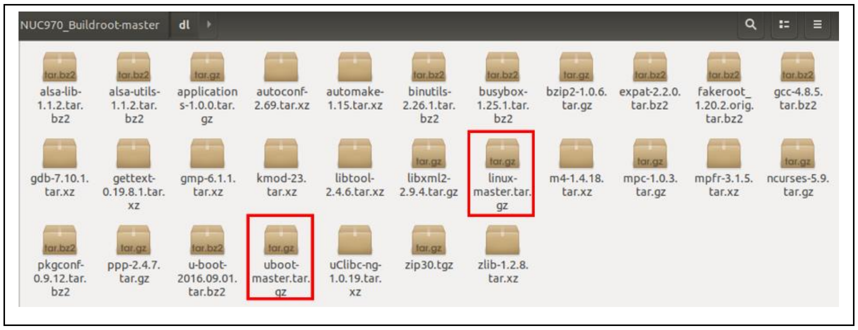
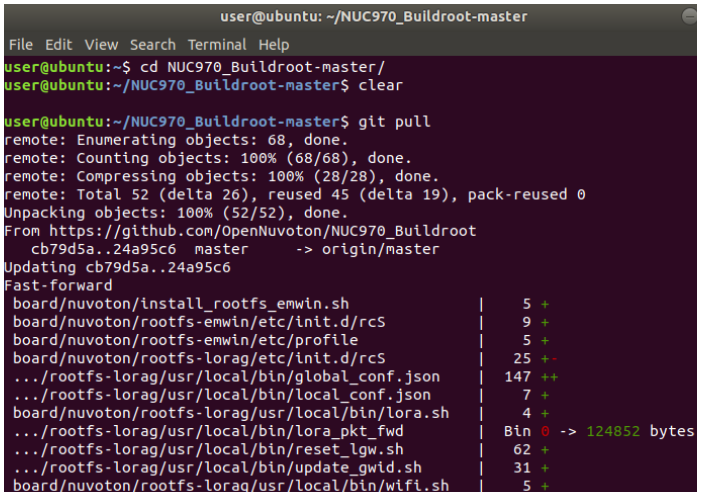


Figure 2-12 File Location of linux-master.gz and uboot-maste.gz

After deleting these files, at the path "*/NUC970_Buildroot-master*" run command "`make`" again, and **NUC970_Buildroot-master** will automatically download the latest patch of Linux and U-boot on the Internet.

Buildroot Update:

At the path "*/NUC970_Buildroot-master*", run command "`git pull`". Then the buildroot will be updated to the latest version, refer to Figure 2-13.



user@ubuntu: ~/NUC970_Buildroot-master
File Edit View Search Terminal Help

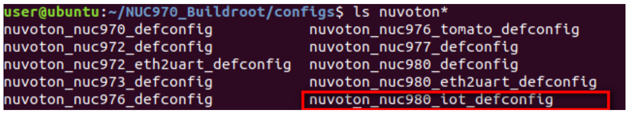
```
user@ubuntu:~$ cd NUC970_Buildroot-master/
user@ubuntu:~/NUC970_Buildroot-master$ clear

user@ubuntu:~/NUC970_Buildroot-master$ git pull
remote: Enumerating objects: 68, done.
remote: Counting objects: 100% (68/68), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 52 (delta 26), reused 45 (delta 19), pack-reused 0
Unpacking objects: 100% (52/52), done.
From https://github.com/OpenNuvoton/NUC970_Buildroot
   cb79d5a..24a95c6  master     -> origin/master
Updating cb79d5a..24a95c6
Fast-forward
 board/nuvoton/install_rootfs_emwin.sh              |    5 +
 board/nuvoton/rootfs-emwin/etc/init.d/rcS          |    9 +
 board/nuvoton/rootfs-emwin/etc/profile             |    5 +
 board/nuvoton/rootfs-lorag/etc/init.d/rcS          |   25 +-
 .../rootfs-lorag/usr/local/bin/global_conf.json    |  147 ++
 .../rootfs-lorag/usr/local/bin/local_conf.json     |    7 +
 board/nuvoton/rootfs-lorag/usr/local/bin/lora.sh   |    4 +
 .../rootfs-lorag/usr/local/bin/lora_pkt_fwd        |  Bin 0 -> 124852 bytes
 .../rootfs-lorag/usr/local/bin/reset_lgw.sh        |   62 +
 .../rootfs-lorag/usr/local/bin/update_gwid.sh      |   31 +
 board/nuvoton/rootfs-lorag/usr/local/bin/wifi.sh   |    5 +
```

Figure 2-13 Buildroot Update

2.5 Compile Linux Kernel and U-boot

First, set the configuration of buildroot. Open a terminal and find board configuration file at the the following path. Take NuMaker NUC980 IIoT for example.

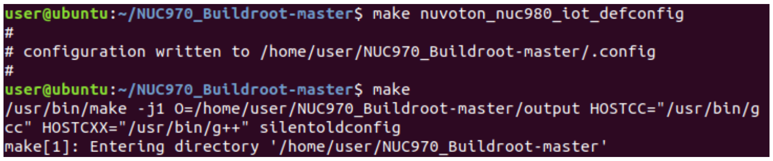


```
user@ubuntu:~/NUC970_Buildroot/configs$ ls nuvoton*
nuvoton_nuc970_defconfig          nuvoton_nuc976_tomato_defconfig
nuvoton_nuc972_defconfig          nuvoton_nuc977_defconfig
nuvoton_nuc972_eth2uart_defconfig nuvoton_nuc980_defconfig
nuvoton_nuc973_defconfig          nuvoton_nuc980_eth2uart_defconfig
nuvoton_nuc976_defconfig          nuvoton_nuc980_iot_defconfig
```

Figure 2-14 Configuration File

Refer to Figure 2-15, run command "`make nuvoton_nuc980_defconfig`" to import configuration file into *.config*, then run command "`make`" to compile Linux kernel and U-boot.

```
# make nuvoton_nuc980_iot_defconfig
# make
```



```
user@ubuntu:~/NUC970_Buildroot-master$ make nuvoton_nuc980_iot_defconfig
#
# configuration written to /home/user/NUC970_Buildroot-master/.config
#
user@ubuntu:~/NUC970_Buildroot-master$ make
/usr/bin/make -j1 O=/home/user/NUC970_Buildroot-master/output HOSTCC="/usr/bin/g
cc" HOSTCXX="/usr/bin/g++" silentoldconfig
make[1]: Entering directory '/home/user/NUC970_Buildroot-master'
```

Figure 2-15 Configuration Setting and Compiling

Note: Base on what target board you use, set the corresponding configuration file.

After compiled, there will be four images used for downloading.

- **Image**: Linux kernel image used for running on DDR.
- **uImage**: Linux kernel image used for booting with uboot.
- **u-boot.bin**: Uboot image.
- **u-boot-spl.bin**: Uboot Secondary Program Loader image", used for booting from SPI NAND.

The images are generated at the following locations:

Image and **uImage** are generated at the path "*NUC970_Buildroot-master/output/images*".

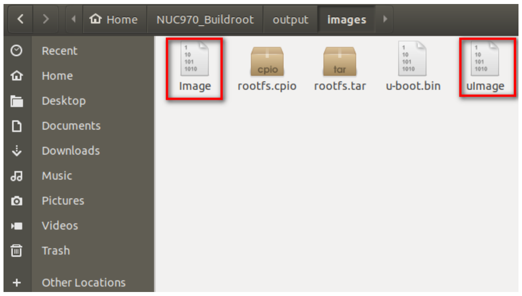


Figure 2-16 Image and uImage

u-boot.bin is generated at the path "*NUC970_Buildroot-master/output/build/uboot-master*".

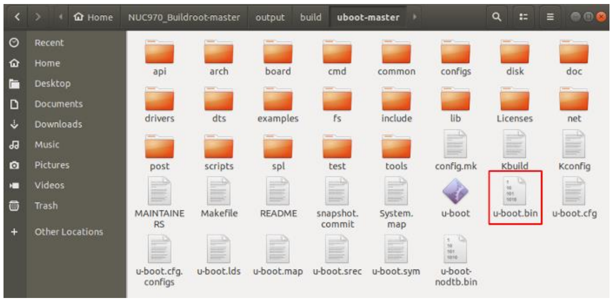


Figure 2-17 u-boot.bin

u-boot-spl.bin is generated at the path "*NUC970_Buildroot-master/output/build/uboot-master/spl*".

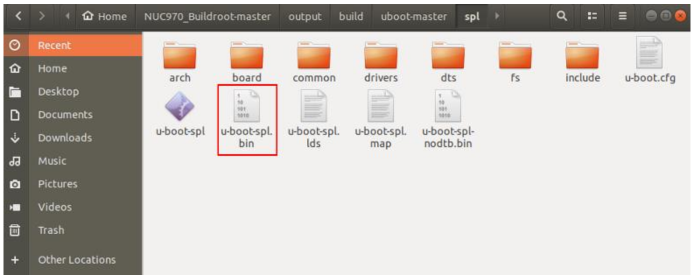


Figure 2-18 u-boot-spl.bin

Download the above four images into your EVB and it will be ready to work.

You can find the download procedure by referring to

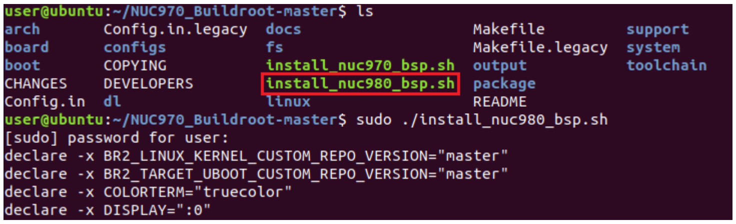
“NuMaker NUC980 IIoT User Manual” (chapter name: NUWRITER TOOL):

URL: https://www.nuvoton.com/export/resource-files/NuMaker_NUC980_IIoT_User_Manual.pdf

2.6 BSP Installation

The LINUX image contains BSP tarball that can be decompressed by shell script "install_nuc980_bsp_sh".

```
# sudo ./install_nuc980_bsp.sh
[sudo] password for user: user
```



```
user@ubuntu:~/NUC970_Buildroot-master$ ls
arch       Config.in.legacy  docs                 Makefile         support
board      configs           fs                   Makefile.legacy  system
boot       COPYING           install_nuc970_bsp.sh  output         toolchain
CHANGES    DEVELOPERS        install_nuc980_bsp.sh  package
Config.in  dl                linux                README
user@ubuntu:~/NUC970_Buildroot-master$ sudo ./install_nuc980_bsp.sh
[sudo] password for user:
declare -x BR2_LINUX_KERNEL_CUSTOM_REPO_VERSION="master"
declare -x BR2_TARGET_UBOOT_CUSTOM_REPO_VERSION="master"
declare -x COLORTERM="truecolor"
declare -x DISPLAY=":0"
```

Figure 2-19 Execute Installation Script

After decompressed, the nuc980bsp folder will be in the NUC970_Buildroot-master folder

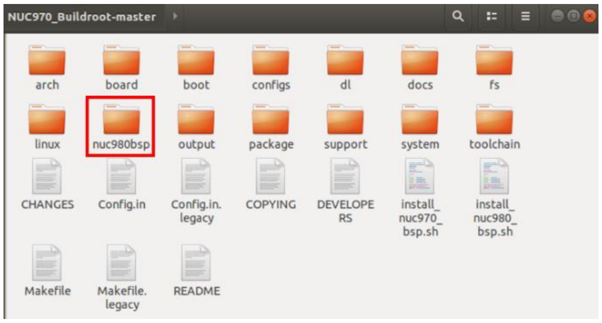


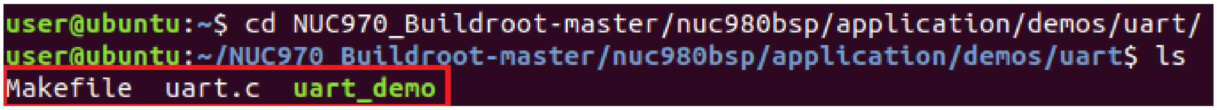
Figure 2-20 nuc980bsp Location

3 LINUX USER APPLICATION

3.1 UART Example

The uart.demo is taken as an example to help users understand how to put their applications into Linux kernel image, compile and download to EVB. We take NuMaker NUC980 IIoT for example.

Enter the uart folder.



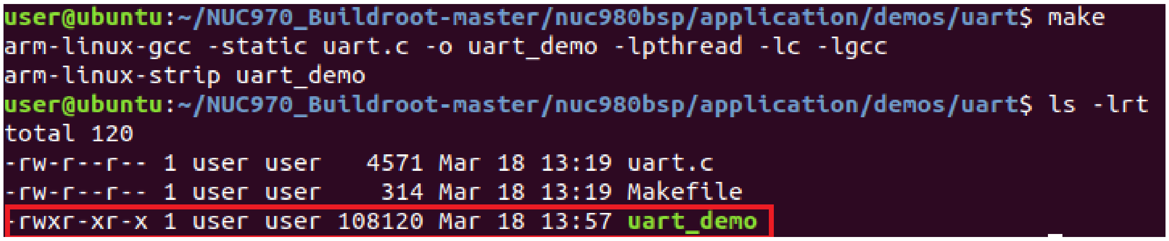
```
user@ubuntu:~$ cd NUC970_Buildroot-master/nuc980bsp/application/demos/uart/
user@ubuntu:~/NUC970_Buildroot-master/nuc980bsp/application/demos/uart$ ls
Makefile  uart.c  uart_demo
```

Figure 3-1 Uart Application Folder

The folder contains three files:

- **Makefile:** Used for cross compiling
- **uart.c:** The source code.
- **uart_demo:** Executable file.

Make the project and generate a new executable file.

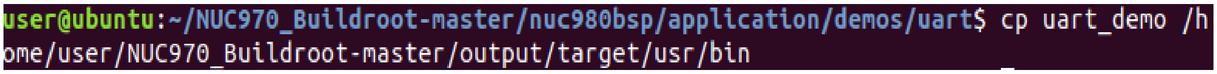


```
user@ubuntu:~/NUC970_Buildroot-master/nuc980bsp/application/demos/uart$ make
arm-linux-gcc -static uart.c -o uart_demo -lpthread -lc -lgcc
arm-linux-strip uart_demo
user@ubuntu:~/NUC970_Buildroot-master/nuc980bsp/application/demos/uart$ ls -lrt
total 120
-rw-r--r-- 1 user user   4571 Mar 18 13:19 uart.c
-rw-r--r-- 1 user user    314 Mar 18 13:19 Makefile
-rwxr-xr-x 1 user user 108120 Mar 18 13:57 uart_demo
```

Figure 3-2 After Compiled

Copy the executable file to the following path:

```
#  cp uart_demo /home/user/NUC970_Buildroot-master/output/target/usr/bin
```



```
user@ubuntu:~/NUC970_Buildroot-master/nuc980bsp/application/demos/uart$ cp uart_demo /h
ome/user/NUC970_Buildroot-master/output/target/usr/bin
```

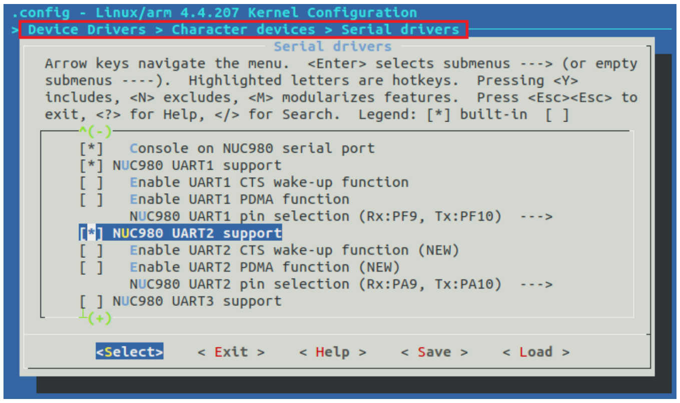
Figure 3-3 Application Placed Location

The application is in the Linux kernel now. If you compile Linux kernel and U-boot, the new generated Image and uImage will have the application inside.

Note: The uart demo needs two sets of UART (UART1 and UART2). Thus, before compiling, user needs to use Linux Kernel Configuration to check if they are enabled or not.

```
#  make linux-menuconfig
```

Go to **Device Drivers —>Character devices —>Serial drivers**, and check **NUC980 UART1, NUC980 UART2** are set to build-in.



```
.config - Linux/arm 4.4.207 Kernel Configuration
> Device Drivers > Character devices > Serial drivers
                              Serial drivers
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>
  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to
  exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]
      ^(-)
      [*]   Console on NUC980 serial port
      [*] NUC980 UART1 support
      [ ]   Enable UART1 CTS wake-up function
      [ ]   Enable UART1 PDMA function
            NUC980 UART1 pin selection (Rx:PF9, Tx:PF10)  --->
      [*] NUC980 UART2 support
      [ ]   Enable UART2 CTS wake-up function (NEW)
      [ ]   Enable UART2 PDMA function (NEW)
            NUC980 UART2 pin selection (Rx:PA9, Tx:PA10)  --->
      [ ] NUC980 UART3 support
      └(+)

        <Select>    < Exit >    < Help >    < Save >    < Load >
```

Figure 3-4 Linux Kernel Configuration

Go to the path "/*NUC970_Buildroot-master*" and run command "`make`" to generate new images. Find new Images, uImage, u-boot, u-boot-spl and download them into NuMaker NUC980 IIoT.

You can find the download procedure by referring to

“NuMaker NUC980 IIoT User Manual” (chapter name: NUWRITER TOOL):

URL: https://www.nuvoton.com/export/resource-files/NuMaker_NUC980_IIoT_User_Manual.pdf

After images downloaded to NuMaker NUC980 IIoT**,** set SW2.1 and SW2.2 to Off, and connect UART1 with UART2.(Refer to Figure 3-5)

Green line: UART1 TX(NU4.2) connect to UART2 RX(CON11.19)

Yellow line: UART1 RX(NU4.1) connect to UART2 TX(CON11.22)

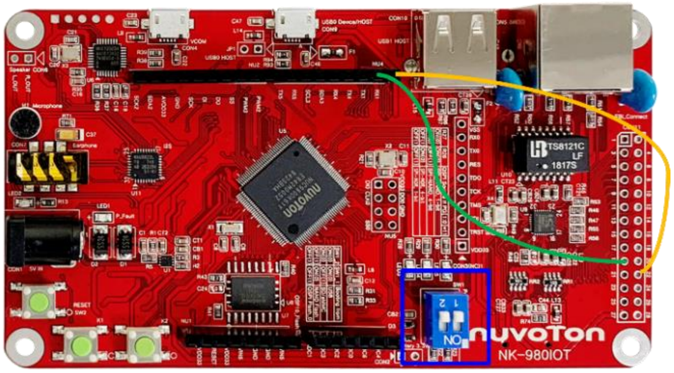
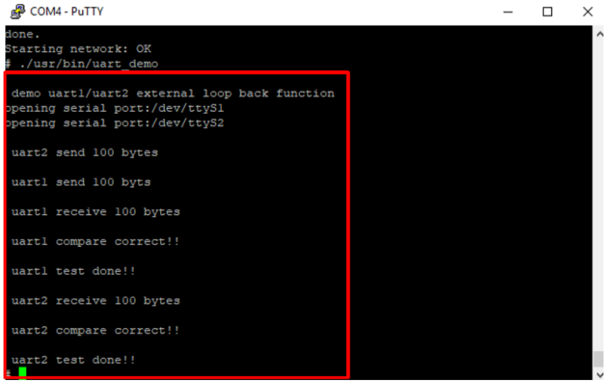


Figure 3-5 UART Demo Hardware Setting

Open the terminal, set Baud rate to 115200 and then boot NuMaker NUC980 IIoT from SPI. After booting is completed, execute uart.demo, You should get the following messages if everything is fine.

```
# ./usr/bin/uart.demo
```



```
done.
Starting network: OK
# ./usr/bin/uart_demo

 demo uart1/uart2 external loop back function
opening serial port:/dev/ttyS1
opening serial port:/dev/ttyS2

 uart2 send 100 bytes

 uart1 send 100 byts

 uart1 receive 100 bytes

 uart1 compare correct!!

 uart1 test done!!

 uart2 receive 100 bytes

 uart2 compare correct!!

 uart2 test done!!
# 
```

Figure 3-6 UART Execution Result

NUC980/NUC970 LINUX ENVIRONMENT ON VMWARE USER MANUAL

4 REVISION HISTORY

Date	Revision	Description
2020.05.20	1.00	1. Initial version

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, “Insecure Usage”.

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer’s risk, and in the event that third parties lay claims to Nuvoton as a result of customer’s Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.