ARM926EJ-S™ Based

32-bit Microprocessor

NuDesign NK-980ETH2P
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 NUC980 microprocessor 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

Table of Contents

1 Overview 3
2 NuDesign NK-980ETH2P Demo Quick Start 4
2.1 Install BSP 4
2.2 Build U-Boot 5
2.3 Build Application 5
2.4 Build Linux Kernel 5
2.5 Program Kernel and U-Boot to SPI Flash 6
2.6 Boot Linux Kernel 9
2.7 Hardware Setup 10
2.8 Execute Sample Code 11
3 Conclusion 15
4 Revision History 16

1 Overview

Nuvoton's NuDesign NK-980ETH2P demo board is based on Nuvoton NUC980DK61Y, including 2 sets 10/100Mbps Ethernet port and 8 sets UART port. The NuDesign NK-980ETH2P demo board can be used as remote control multiple serial device for industrial automation data transmission and distal environmental monitoring.

User can bring up the NuDesign NK-980ETH2P board following the instruction of this document.

2 NuDesign NK-980ETH2P Demo Quick Start

2.1 Install BSP

Linux BSP contains three directories. Content of each directory listed in following table:

Directory Name	Content
BSP	A tar ball contains cross compiler, root file system, and pre-build tool chain.
Documents	BSP related documents.
Tools	NuWriter tool and its driver for Windows and SD writer tool.

Table 2-1 BSP Content

Source code could be downloaded using repo tool. Below list the steps of doing so.

Make sure you have a bin/ directory in your home directory and that is included in your path.

```
$ mkdir ~/bin
$ export PATH=~/bin:$PATH
```

Download the repo tool and ensure it is executable.

```
$ curl https://storage.googleapis.com/git-repo-downloads/repo >
~/bin/repo
$ chmod a+x ~/bin/repo
```

Create an empty directory to hold working directory.

```
$ mkdir WORKING_DIR
$ cd WORKING_DIR
```

Configure git with your real name and email address.

```
$ git config --global user.name "Your Name"
$ git config --global user.email "you@example.com"
```

Use one of following commands to download manifest file for NUC980 BSP. The first command download from Github, and the second command download from Gitee. You can use the command to select the site with faster download speed.

```
$ repo init -u git://github.com/OpenNuvoton/manifest.git -b nuc980-
2019.09 -m github.xml
```

Or.

```
$ repo init -u https://gitee.com/OpenNuvoton/manifest.git -b nuc980-
2019.09 -m gitee.xml
```

Then download source code.

```
$ repo sync
```

After downloading the source code, please copy the tar ball under BSP directory to Linux machine and use following command to extract the file.

```
$ tar zxvf nuc980bsp.tar.gz
```

After entering nuc980bsp directory, execute the installation script install.sh. This script requires the administrator privilege to execute. You can use “su” command to switch to root and execute the installation script.

```
$ su
Password: (Enter password of root)
```

```
# ./install.sh
```

Or execute this script as root by using sudo command. (This method works for those distributions do not open the root account privilege, such as Ubuntu.)

```
# sudo ./install.sh
```

Below is the console output during installation. The path input should be the same as the WORKING_DIR set previously.

```
Now installing arm_linux_4.8 tool chain to /usr/local/
Setting tool chain environment
Installing arm_linux_4.8 tool chain successfully
Install rootfs, applications, u-boot and Linux kernel
Please enter absolute path for installing(eg:/home/<user name>) :
BSP will be installed in /<path you input>/nuc980bsp
/home/someone
Extract rootfs and pre-build images
...
...
NUC980 BSP installion complete
```

2.2 Build U-Boot

User can compile NuDesign NK-980ETH2P U-Boot code with following steps.

Enter U-Boot folder.

```
# cd u-boot-2016.11
```

Config U-Boot for NuDesign NK-980ETH2P demo voard.

```
$ make nuc980_eth2uart_defconfig
```

Compile U-Boot.

```
$ make
......
  COPY    u-boot.bin
  SYM     u-boot.sym
./scripts/check-config.sh u-boot.cfg \
                ./scripts/config_whitelist.txt . 1>&2
```

2.3 Build Application

User can compile NuDesign NK-980ETH2P application code with following steps.

Enter application folder.

```
# cd applications
```

Type “make eth2uart” to build application code and copy generated files to rootfs folder.

```
$ make eth2uart
```

2.4 Build Linux Kernel

User can build kernel image with following steps.

Enter kernel folder.

```
# cd linux-4.4.y
```

Config kernel for NuDesign NK-980ETH2P demo board.

```
# make nuc980_eth2uart_defconfig
```

Build kernel.

```
$ make uImage
......
    Kernel: arch/arm/boot/Image is ready
cp arch/arm/boot/Image   ../image/980image
  Kernel: arch/arm/boot/Image is ready
cp arch/arm/boot/Image   ../image/980image
  GZIP    arch/arm/boot/compressed/piggy.gzip
  AS      arch/arm/boot/compressed/piggy.gzip.o
  LD      arch/arm/boot/compressed/vmlinux
  OBJCOPY arch/arm/boot/zImage
  Kernel: arch/arm/boot/zImage is ready
  Kernel: arch/arm/boot/Image is ready
cp arch/arm/boot/Image   ../image/980image
  Kernel: arch/arm/boot/zImage is ready
  UIMAGE  arch/arm/boot/uImage
Image Name:   Linux-4.4.115
Created:      Tue Dec 25 16:15:24 2018
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:    8039480 Bytes = 7851.05 kB = 7.67 MB
Load Address: 00008000
Entry Point:  00008000
  Image arch/arm/boot/uImage is ready
cp arch/arm/boot/uImage   ../image/980uimage
$ ls ../image/
980image 980uimage
```

2.5 Program Kernel and U-Boot to SPI Flash

This section introduces how to program U-Boot and kernel to SPI Flash.

A. Install NuWrite Driver. (Please refer to “NUC980 NuWriter User Manual”)

B. Set SW1 (Power On Setting) to Boot from USB (shown in Table 2-2 and Figure 2-1). Connect USB0 to PC and connect UARTconsole to PC.

SW1.2/SW1.1 Switch State	Function
ON/ON	Boot from USB
ON/OFF	Boot from SD/eMMC
OFF/OFF	Boot from SPI Flash

Table 2-3 Power On Setting

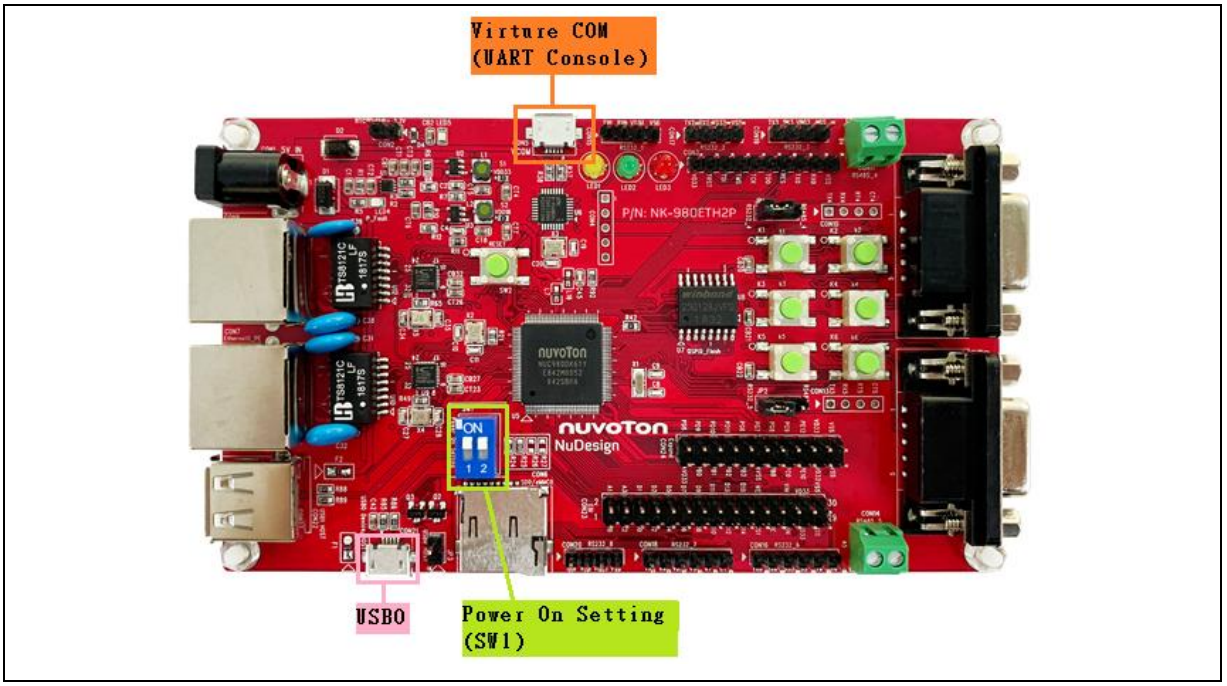

Figure 2-2 Burn Kernel to SPI Flash

C. Open NuWriter. Select target chip as “NUC980 series” and select DDR parameter is “NUC980DK61Y.ini”. Then, click “Continue” button.

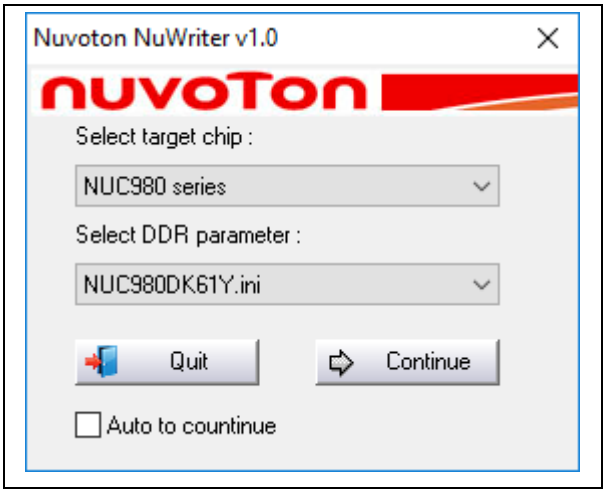

Figure 2-3 Nu-Writer Setting

D. Program u-boot.bin in “uboot_v2016.11” folder:

1. Select “SPI” mode
2. Input parameter
   - **Image Name :** u-boot.bin (refer to section 2.2)
   - **Image Type :** Loader
   - **Image execute address :** 0xe00000
3. Click “Program”

4. Wait program finish
5. Click “Verify” to check correctness of programmed data.

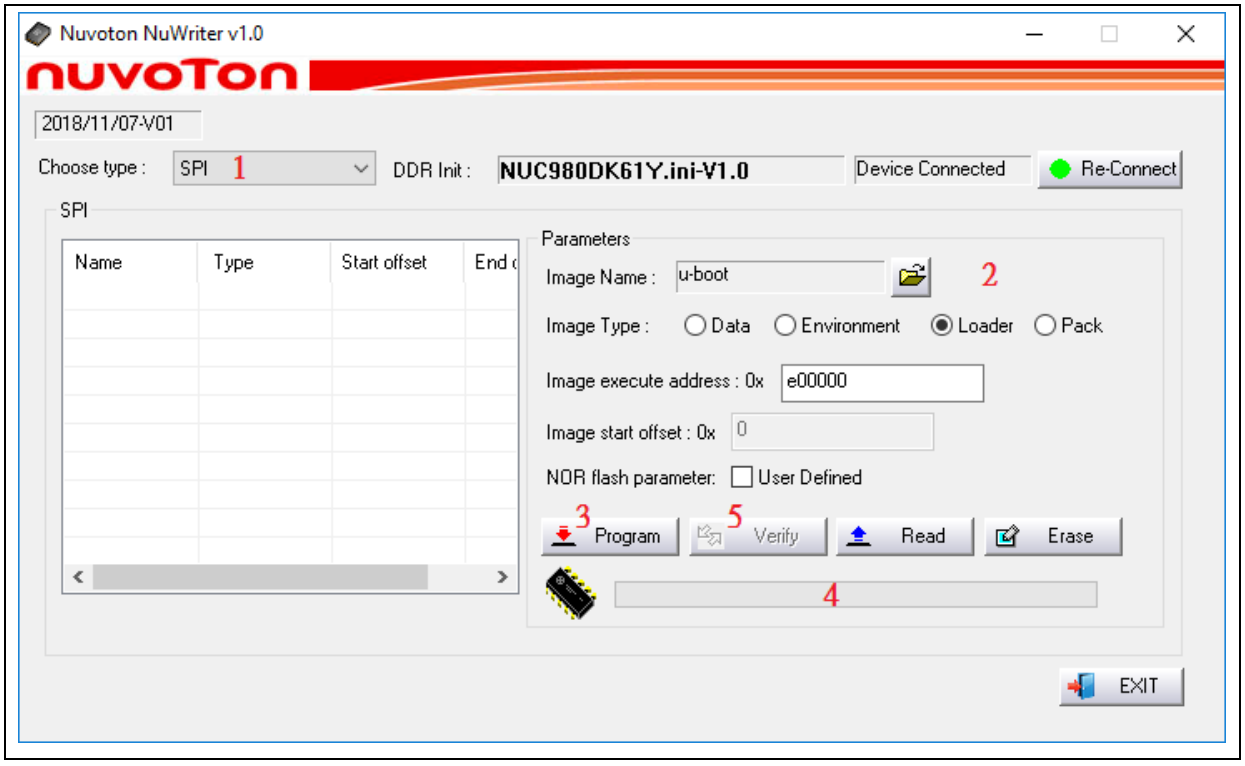

Figure 2-4 Program u-boot

E. Program kernel image:

1. Select “SPI” mode
2. Input parameter: Image Name, Image Type and Image start offset. (shown as below Fig.)
3. Click “Program”
4. Wait program finish
5. Click “Verify” to check correctness of programmed data.

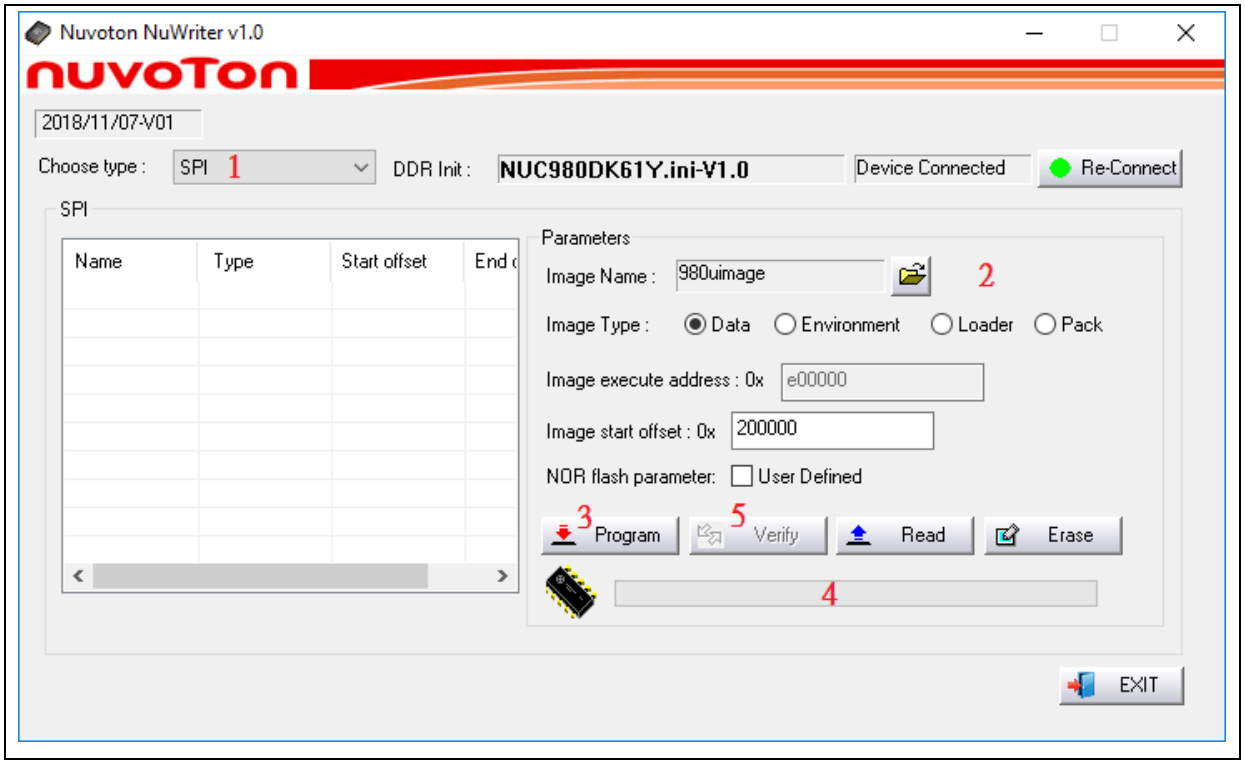

Figure 2-5 Program Kernel Image

2.6 Boot Linux Kernel

This section describes how to boot Linux kernel.

A. Set SW1(Power On Setting) to Boot from QSPI 0 Flash(refer to Table 2-4 and Figure 2-6).

B. Click Reset button on demo board. From console user can find system enter U-Boot. The user can use following commands to launch Linux kernel after enter U-Boot shell.

- Type “sf probe 0 18000000” to set SPI speed (optinoal)
- Type “sf read 0x7FC0 0x200000 0x760000” to read kernel image from SPI Flash to DDR.
- Type “bootm 0x7FC0” to boot Linux kernel image.

```
U-Boot 2016.11-g9618a94-dirty (Dec 25 2018 - 08:46:04 +0800)


CPU: NUC980
Board: NUC980
DRAM:  64 MiB
NAND:  NAND Flash not found !
NUC980 NAND CONTROLLER IS NOT SUPPORT THE PAGE SIZE. (0, 0)
0 MiB
SF: Detected W25Q128BV with page size 256 Bytes, erase size 4 KiB,
total 16 MiB
*** Warning - bad CRC, using default environment


In:    serial
Out:   serial
```

```
Err:   serial
Net:   Net Initialization Skipped
No ethernet found.
=> sf probe 0 18000000
SF: Detected W25Q128BV with page size 256 Bytes, erase size 4 KiB,
total 16 MiB
=> sf read 0x7FC0 0x200000 0x760000
device 0 offset 0x200000, size 0x760000
SF: 7733248 bytes @ 0x200000 Read: OK
=> bootm 0x7FC0
## Booting kernel from Legacy Image at 00007fc0 ...
   Image Name:   Linux-4.4.115+
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    7573624 Bytes = 7.2 MiB
   Load Address: 00008000
   Entry Point:  00008000
   Verifying Checksum ... OK
   XIP Kernel Image ... OK

Starting kernel ...
```

C. After booting Linux kernel image, user can see following information from UART console.

```
Freeing unused kernel memory: 5456K
[Mount JFFS2]: /dev/mtdblock0 --> /mnt/mtdblock0
nuc980-emac0 nuc980-emac0: eth0 is OPENED
nuc980-emac1 nuc980-emac1: eth1 is OPENED
random: arm-linux-light: uninitialized urandom read (8 bytes read, 7
bits of entropy available)



BusyBox v1.22.1 (2016-02-03 14:11:04 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.


~ #
```

D. For the detailed kernel compile and setting, please refer to “NUC980 Linux BSP User Manual”.

2.7 Hardware Setup

A. Set SW1(Power On Setting) to Boot from QSPI 0 Flash(refer to Table 2-5 and Figure 2-7).

B. Connect UART console port

C. Connect Ethernet0 to PC and connect UART1~8 to other UART device (ex:PC COM port).

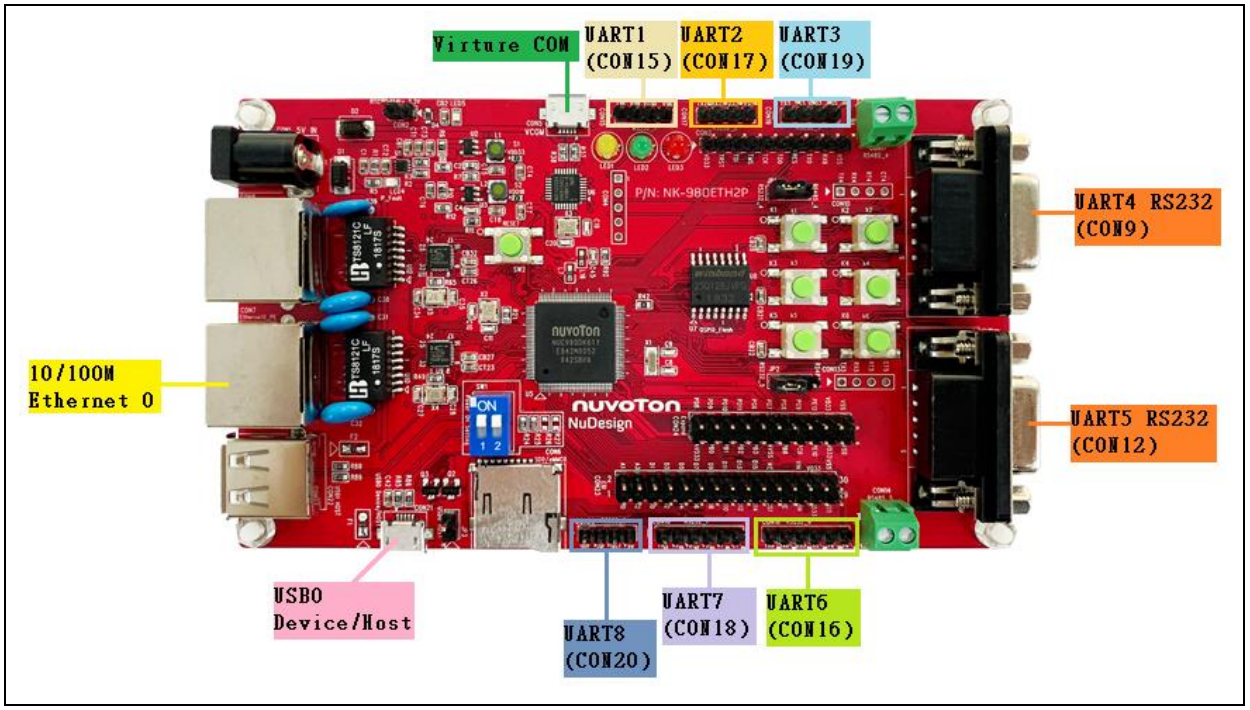

Figure 2-8 NuDesign NK-980ETH2P Board Setup

2.8 Execute Sample Code

First, make sure UART console connect to PC and use Terminal tool, such as Putty or Teraterm to open the serial COM port. The COM port configuration is 115200bps, 8-bit data length and no-parity.

User also needs to ensure that the PC Ethernet port is connected to the demo board Ethernet0 (or Ethernet1) port. For PC Internet settings, refer to the figure below.

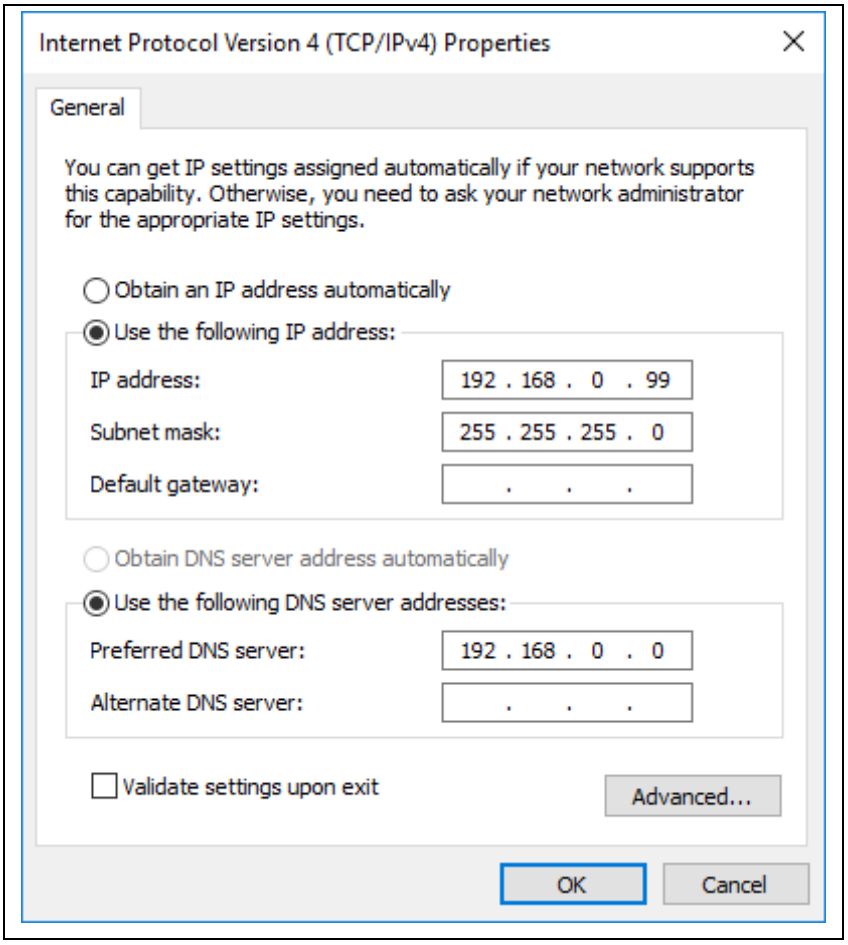

Figure 2-9 Serial COM Port

Power ON or push the Reset key and boot system (refer to section 2.6) .

```
Freeing unused kernel memory: 5456K
[Mount JFFS2]: /dev/mtdblock0 --> /mnt/mtdblock0
nuc980-emac0 nuc980-emac0: eth0 is OPENED
nuc980-emac1 nuc980-emac1: eth1 is OPENED
random: arm-linux-light: uninitialized urandom read (8 bytes read, 7
bits of entropy available)



BusyBox v1.22.1 (2016-02-03 14:11:04 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.


~ #
```

Use Terminal tool to open serial COM port (from UART1 to UART8)

And use Terminal tool open TCP/IP connection. Ethernet 0 IP Address is 192.168.0.100, Port number from 50001 to 50008. Ethernet 1 IP Address is 192.168.10.100, Port number from 50001 to 50008. Where Transmitting and receiving of port numbers 50001~50008 maps to to

UART1~8 respectively.

Below is an example transmit data from Ethernet to UART. When type “123” in TCP/IP connection window which port number is 50001. The UART1 serial COM port window will show “123”.

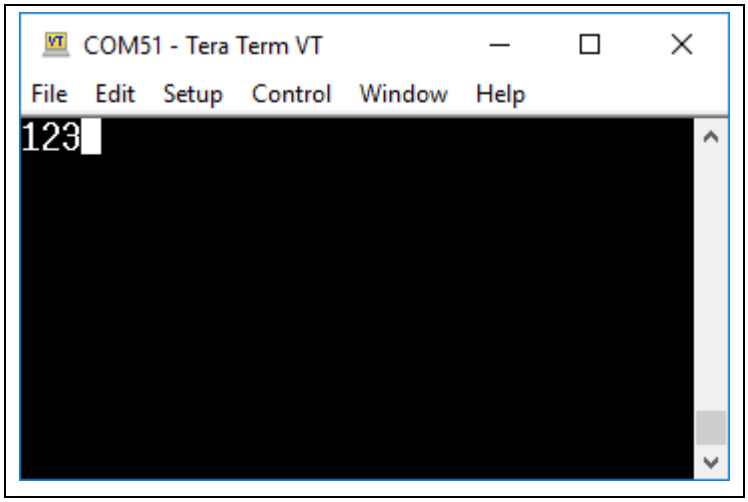

Figure 2-10 Serial COM Port

The following is an example with other direction, transmit data from UART to Ethernet. When typing “123” in UART1 serial COM port. The TCP/IP connection window which port number is 50001 will show “123”.

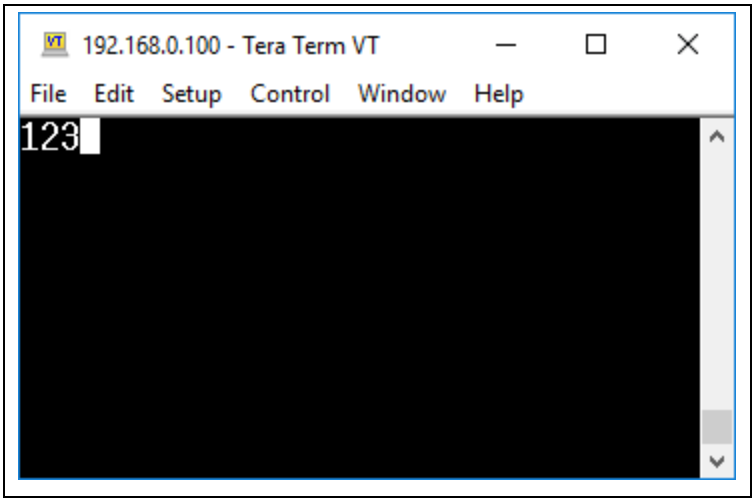

Figure 2-11 TCP/IP connection window

User can configure UART port via browser with following steps.

1. Use browser connect to http://192.168.0.100 (Ethernet 0) or http://192.168.10.100 (Ethernet 1).
2. Configure UARTs’ attribute. Including UART port, baud-rate, data length, parity, stop bit, flow control, enable/diable RS485.
3. Click Submit

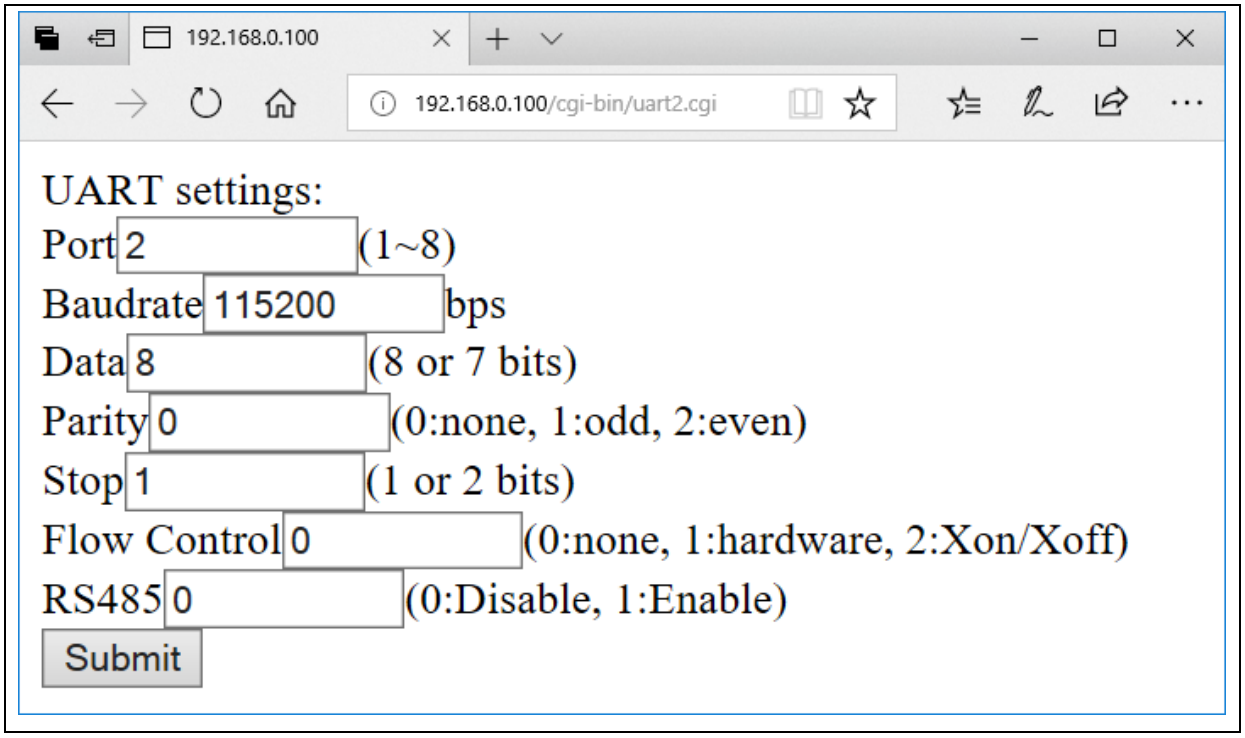

Figure 2-12 UART Setting Web Page

3 Conclusion

Nuvoton's NuDesign NK-980ETH2P demo code can transfer data received by UART to the Ethernet, and can also transfer the data received by Ethernet to UART.

Users can develop applications based on this demoe code, especially in industrial automation data transmission.

4 Revision History

Version	Date	Description
1.00	Sept. 25, 2018	Initial release.
1.01	Mar. 21, 2019	Updated manifest file download branch.
1.02	Apr. 22, 2019	Editorial Change.

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.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.