NuMaker NUC980 Serial Server User Manual

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

www.nuvoton.com

Table of Contents

List of Figures

List of Tables

1 OVERVIEW

NuMaker NUC980 Serial Server can gather information from up to 8 UART sensors and transfer to cloud server over two on board Ethernet port. It can also transfer information from cloud server by Ethernet to UART devices in opposite direction by applications. This serial server is a very popular IIoT device used in industrial control, which can easily transfer an onsite industrial machine control into a remote cloud factory control.

NuMaker NUC980 Serial Server Development Board has two sets of RS232/RS485 transceiver ports on board and six sets of UART function pins. Company with NUC980 high performance DMA channels, the data transfer is in a very high efficiency way between 8 UARTs and 2 Ethernet. This is why NUC980 can easily satisfy most of the high performance serial transfer requirements.

This document provides a quick start guide for the NuMaker NUC980 Serial Server Development Board. Users can understand both software and hardware configurations for the NuMaker NUC980 Serial Server. The platform provides Linux OS and plenty of industrial control protocol for users to implement the Ethernet control applications in a very short time.

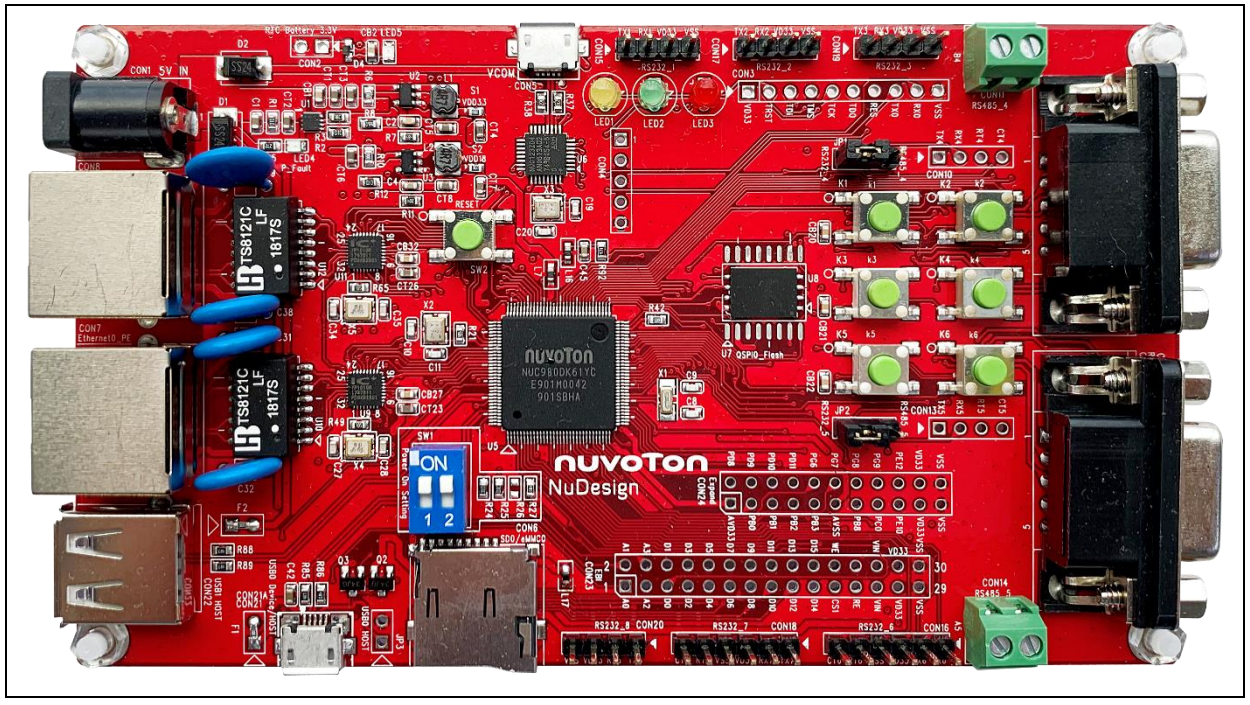


Figure 1-1 NuMaker NUC980 Serial Server Development Board

2 FEATURES

- NUC980DK61Y: LQFP128 pin MCP package with DDR2 (64 MB), which can run up to 300MHz operating speed
- SPI Flash: Quad mode system booting or data storage
- SD0/eMMC0: User SD/eMMC memory card for system booting, data storage or SDIO (Wi-Fi) device
- Provides 9 sets of COM ports
  - UART0: Connected to Virtual COM port for system development, debug message output
  - UART4/UART5: 2 sets of DB9 port with RS232 transceiver
  - UART1~3/UART6~8: 6 sets of pin headers
- JTAG interface provided for software development
- 2 sets of RJ45 port with Ethernet 10/100Mbps MAC
- 3 sets of LED for status indication
- 6 sets of user-configurable push button keys
- USB port-0 that can be used as Device/HOST and USB port-1 that can be used as HOST

  Supports pen drives, keyboards, mouse and printers
- Provides over-voltage and over current protection
- 3.3V I/O power, 1.8V Memory power and 1.2V core power

3 HARDWARE CONFIGURATION

3.1 Front View

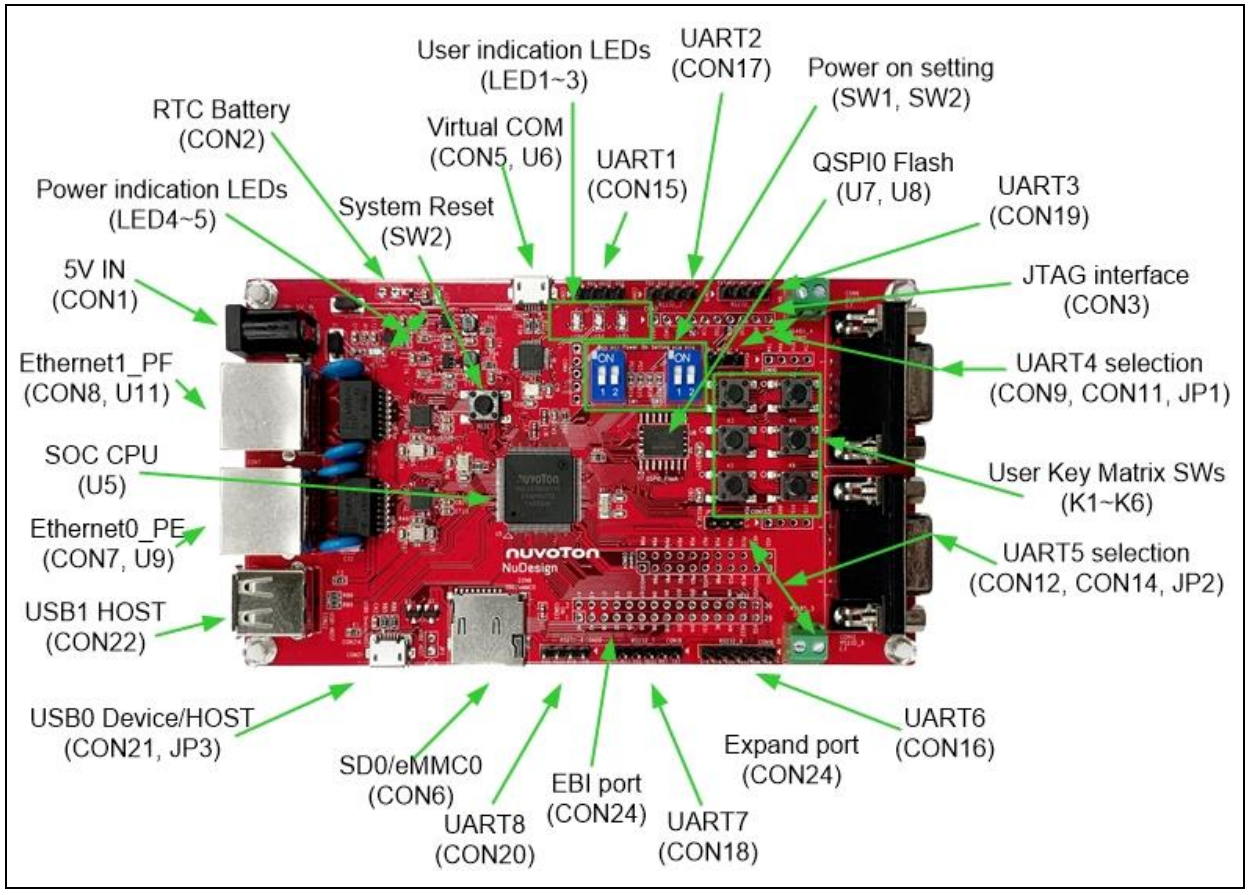


Figure 3-1 Front View of NuMaker NUC980 Serial Server

Figure 3-1 shows the main components from the front view of NuMaker NUC980 Serial Server Development Board

- +5V In (CON1): Power adaptor 5V input

Power Model	CON5 USB Port (Micro-B)	CON21 USB Port (Micro-B)	CON1
Model 1	Connect to PC	-	-
Model 2	-	Connect to PC	-
Model 3	-	-	VDD5V Input

- Power indication LEDs (LED4, LED5):

LED	Color	Descriptions
LED4	Red	The system power will be terminated and LED4 lighting when the input voltage is over 5.7V or the current is over 1.7A.
LED5	Green	Power normal state.

- RTC Battery (CON2): External Battery supply for RTC 3.3V powered
  - CON2.1: Positive (+)
  - CON2.2: Negative (-)
- System Reset (SW2): System will be reset if the SW2 button is pressed
- Virtual COM (CON5, U6): NUC123ZD4AN0 microcontroller (U6), USB micro-B connector (CON5) to PC, for debug message output
- User indication LEDs (LED1, LED2, LED3):

LED	Color	GPIO pin of NUC980
LED1	Yellow	GPG15
LED2	Green	GPB13
LED3	Red	GPF10

- UART1 pin header (CON15)

Connector	GPIO pin of NUC980	Function
CON15.1	GPA1	UART1_TXD
CON15.2	GPA0	UART1_RXD
CON15.3	-	VDD33
CON15.4	-	VSS

- QSPI0 Flash (U7, U8): Use Winbond W25N01GVZE1G 128 MB SPI-NAND (U7) for system booting, only one (U7 or U8) SPI Flash can be used, support dual / quad mode
- UART2 pin header (CON17)

Connector	GPIO pin of NUC980	Function
CON17.1	GPA10	UART2_TXD
CON17.2	GPA9	UART2_RXD
CON17.3	-	VDD33
CON17.4	-	VSS

- UART3 pin header (CON19).

Connector	GPIO pin of NUC980	Function
CON19.1	GPC3	UART3_TXD
CON19.2	GPC4	UART3_RXD
CON19.3	-	VDD33
CON19.4	-	VSS

- JTAG interface and UART0 (CON3)

Connector	GPIO pin of NUC980	Function
CON3.1	-	VDD33
CON3.2	GPG15	nTRST
CON3.3	GPG14	TDI
CON3.4	GPG13	TMS
CON3.5	GPG12	TCK
CON3.6	GPG11	TDO
CON3.7	-	nRESET
CON3.8	GPF12	UART0_TXD
CON3.9	GPF11	UART0_RXD
CON3.10	-	VSS

- UART4 selection (CON9, CON11, JP1):
  - JP1: 1-2 short for RS232 function with RS232 transceiver, and RS232 connected DB9 female (CON9)
  - JP1: 2-3 short for RS485 function with RS485 transceiver, and RS485 connected to 2P terminal (CON11)

Function	GPIO pin of NUC980
UART4_232_TXD/485_D	GPD12
UART4_232_RXD/485_R	GPD13
UART4_232_RTS/485_(/RE&DE)	GPD14
UART4_232_CTS	GPD15

- User Key Matrix SWs (K1~K6)

Key	Function	GPIO pin of NUC980
K1	Row0	GPC13
	Column0	GPC1
K2	Row0	GPC13
	Column1	GPC2
K3	Row1	GPC14
	Column0	GPC1
K4	Row1	GPC14
	Column1	GPC2
K5	Row2	GPC15
	Column0	GPC1
K6	Row2	GPC15
	Column1	GPC2

- UART5 selection (CON12, CON14, JP2):
  - JP2: 1-2 short for RS232 function with RS232 transceiver, and RS232 connected DB9 female (CON12)
  - JP2: 2-3 short for RS485 function with RS485 transceiver, and RS485 connected to 2P terminal (CON14)

Function	GPIO pin of NUC980
UART5_232_TXD/485_D	GPG14
UART5_232_RXD/485_R	GPG13
UART5_232_RTS/485_(/RE&DE)	GPG12
UART5_232_CTS	GPG11

- UART6 pin header (CON16)

Connector	GPIO pin of NUC980	Function
CON16.1	GPA5	UART6_TXD
CON16.2	GPA4	UART6_RXD
CON16.3	-	VDD33
CON16.4	-	VSS
CON16.5	GPA3	UART6_RTS
CON16.6	GPA2	UART6_CTS

- Expand port for user use (CON24)

Connector	GPIO pin of NUN980	Function
CON24.1	-	ADC VSS
CON24.2	GPD8	SPI0_SS0
CON24.3	GPB0	ADC_AIN[0]
CON24.4	GPD9	SPI0_CLK
CON24.5	GPB1	ADC_AIN[1]
CON24.6	GPD10	SPI0_DO
CON24.7	GPB2	ADC_AIN[2]
CON24.8	GPD11	SPI0_DI
CON24.9	GPB3	ADC_AIN[3]
CON24.10	GPG6	PWM10
CON24.11	-	ADC VDD33
CON24.12	GPG7	PWM11
CON24.13	GPB8	CAN2_RXD
CON24.14	GPG8	PWM12
CON24.15	GPC0	CAN2_TXD
CON24.16	GPG9	PWM13
CON24.17	GPE10	I2C0_SDA
CON24.18	GPE12	I2C0_SCL
CON24.19	-	VDD33
CON24.20	-	VDD33
CON24.21	-	VSS
CON24.22	-	VSS

- UART7 pin header (CON18)

Connector	GPIO pin of NUC980	Function
CON18.1	GPB6	UART7_TXD
CON18.2	GPB4	UART7_RXD
CON18.3	-	VDD33
CON18.4	-	VSS
CON18.5	GPB5	UART7_RTS
CON18.6	GPB7	UART7_CTS

- EBI port for user use (CON23)

Connector	GPIO pin of NUN980	Function
CON23.1	GPG0	EBI_ADDR0
CON23.2	GPG1	EBI_ADDR1
CON23.3	GPB2	EBI_ADDR2
CON23.4	GPG3	EBI_ADDR3
CON23.5	GPC0	EBI_DATA0
CON23.6	GPC1	EBI_DATA1
CON23.7	GPC2	EBI_DATA2
CON23.8	GPC3	EBI_DATA3
CON23.9	GPC4	EBI_DATA4
CON23.10	GPC5	EBI_DATA5
CON23.11	GPC6	EBI_DATA6
CON23.12	GPC7	EBI_DATA7
CON23.13	GPC8	EBI_DATA8
CON23.14	GPC9	EBI_DATA9
CON23.15	GPC10	EBI_DATA10
CON23.16	GPC11	EBI_DATA11
CON23.17	GPC12	EBI_DATA12
CON23.18	GPC13	EBI_DATA13
CON23.19	GPC14	EBI_DATA14
CON23.20	GPC15	EBI_DATA15
CON23.21	GPA6	EBI_nCS1
CON23.22	GPA7	EBI_nWE
CON23.23	GPA8	EBI_nRE
CON23.24	-	-
CON23.25	-	VIN
CON23.26	-	VIN
CON23.27	-	VDD33
CON23.28	-	VDD33
CON23.29	-	VSS
CON23.30	-	VSS

- UART8 pin header (CON20)

Connector	GPIO pin of NUC980	Function
CON20.1	GPA12	UART8_TXD
CON20.2	GPA11	UART8_RXD
CON20.3	-	VDD33
CON20.4	-	VSS

- SD0/eMMC0 (CON6): Use Micro SD/eMMC memory card for system booting, data storage or SDIO (Wi-Fi) device
- Power on setting (SW1, SW2)

Switch	Status	Function	GPIO pin of NUC980
SW1.2/SW1.1	ON/ON	Boot from USB	GPG1/GPG0
SW1.2/SW1.1	ON/OFF	Boot from SD/eMMC	GPG1/GPG0
SW1.2/SW1.1	OFF/OFF	Boot from QSPI0 Flash	GPG1/GPG0

Resistance	Status	Function	GPIO pin of NUC980
R24	Solder R	Watchdog Timer OFF	GPG3
R24	Remove	Watchdog Timer ON	GPG3

Resistance	Status	Function	GPIO pin of NUC980
R25	Solder R	UART0 debug message ON	GPG5
R25	Remove	UART0 debug message OFF	GPG5

If SW1.2/SW1.1 status is ON / OFF

Resistance	Status	Function	GPIO pin of NUC980
SW2.2/SW2.1	ON/ON	SD0/eMMC0 boot from GPC group	GPG9/GPG8

If SW1.2/SW1.1 status is OFF / OFF

Switch	Status	Function	GPIO pin of NUC980
SW2.2/SW2.1	ON/ON	SPI-NAND Flash boot with 1-bit mode	GPG9/GPG8
SW2.2/SW2.1	ON/OFF	SPI-NAND Flash boot with 4-bit mode	GPG9/GPG8

SW2.2/SW2.1	OFF/ON	SPI-NOR Flash boot with 4-bit mode	GPG9/GPG8
SW2.2/SW2.1	OFF/OFF	SPI-NOR Flash boot with 1-bit mode	GPG9/GPG8

- USB0 Device/HOST (CON21, JP3): USB0 Device/HOST Micro-B connector, By JP3 status or defined by the ID pin of the USB cable
- USB1 HOST (CON22): USB1 for USB HOST with type-A connector
- Ethernet0_PE (CON7, U9): For Ethernet port, the NUC980 support RMII interface which add one Ethernet PHY IP101GR to RJ45 connector with LED indicator
- SOC CPU: NUC980DK61Y (U5)
- Ethernet1_PF (CON8, U11): For Ethernet port, the NUC980 support RMII interface which add one Ethernet PHY IP101GR to RJ45 connector with LED indicator

3.2 Rear View

Figure 3-2 shows the main components from the rear view of NuMaker NUC980 Serial Server Development Board

- RS232-4/5 transceivers with SN75C3232E (U13 and U15)
- RS485-4/5 transceivers with SN65HVD10 (U14 and U16)

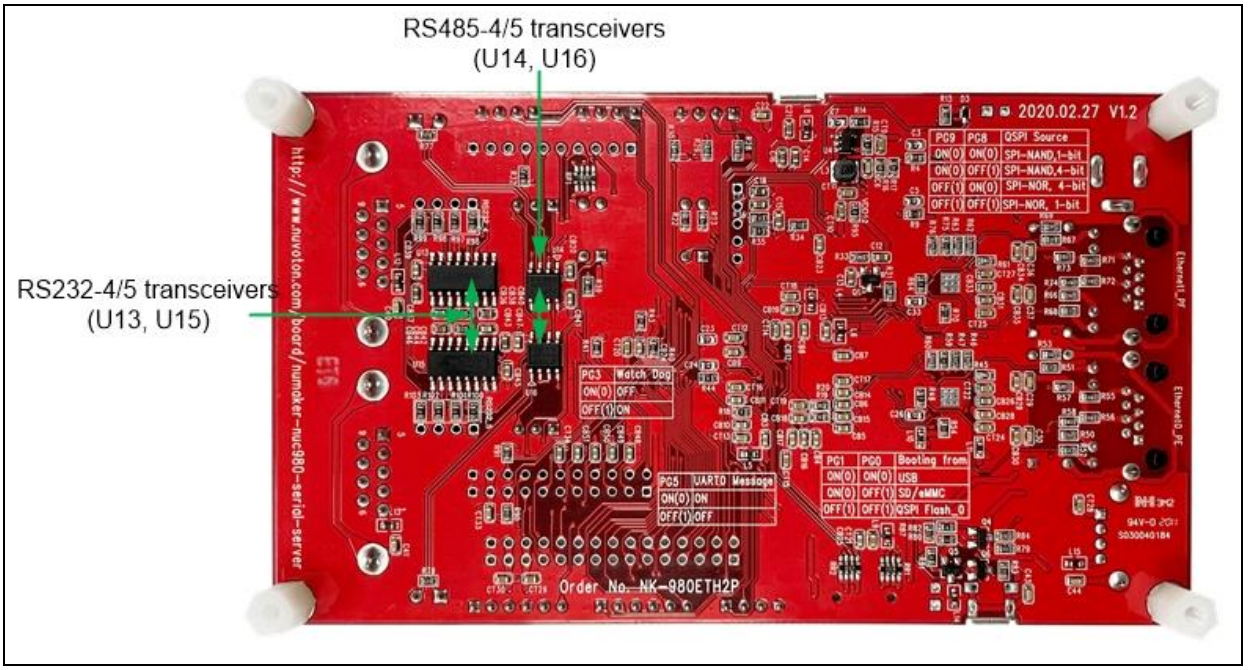


Figure 3-2 Rear View of NuMaker NUC980 Serial Server

4 QUICK START

4.1 BSP Download

The burning tool requires a NuWriter driver to be installed on PC first. Please follow the steps below to install the driver.

Please visit nuvoTon's NuMicro™ website https://www.nuvoton.com/products/iot-solution/iot-platform/numaker-server-nuc980/?group=Software&tab=2) to download the "NUC980_Linux-4.4_BSP_v1.02.001". Run the "**WinUSB4NuVCOM.exe**" before the USB cable is plugged in. The "**WinUSB4NuVCOM.exe**" can be found in the "Tool" directory. Power on the NUC980 Series MPU EVB and plug the USB cable into PC, the Windows shall find a new device and then request to install its driver. Simply follow the installation and optional steps to install USB Driver, included VCOM driver.

4.2 Driver Installation

The programming tool requires a Nuvoton USB driver to be installed on PC first. Please follow the steps below to install the WinUSB driver.

Run the "WinUSB4NuVCOM.exe" before the USB cable is plugged in. The "WinUSB4NuVCOM.exe" can be found in the "Tool" directory. Power on the NUC980 Series MPU EVB and plug the USB cable into PC, the Windows shall find a new device and request to install the driver.

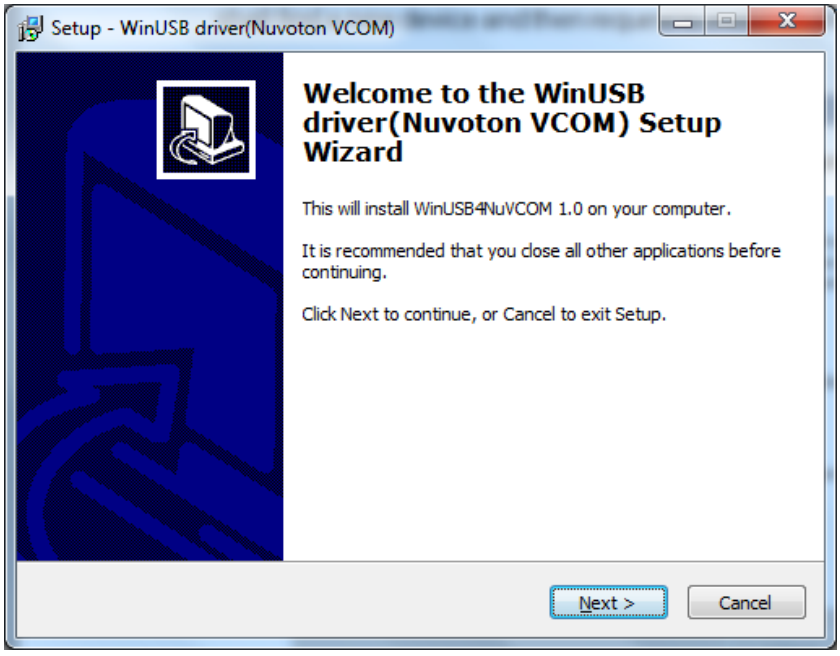
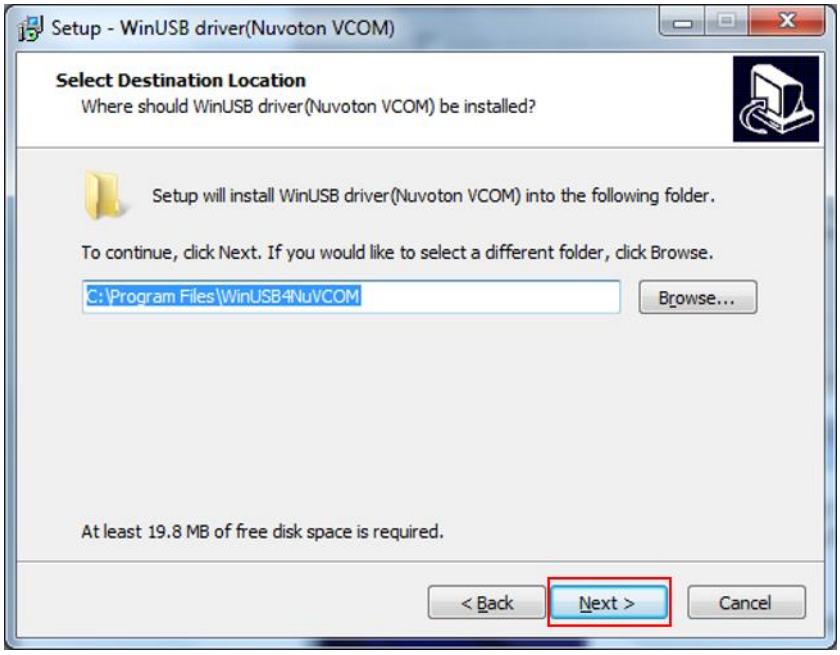
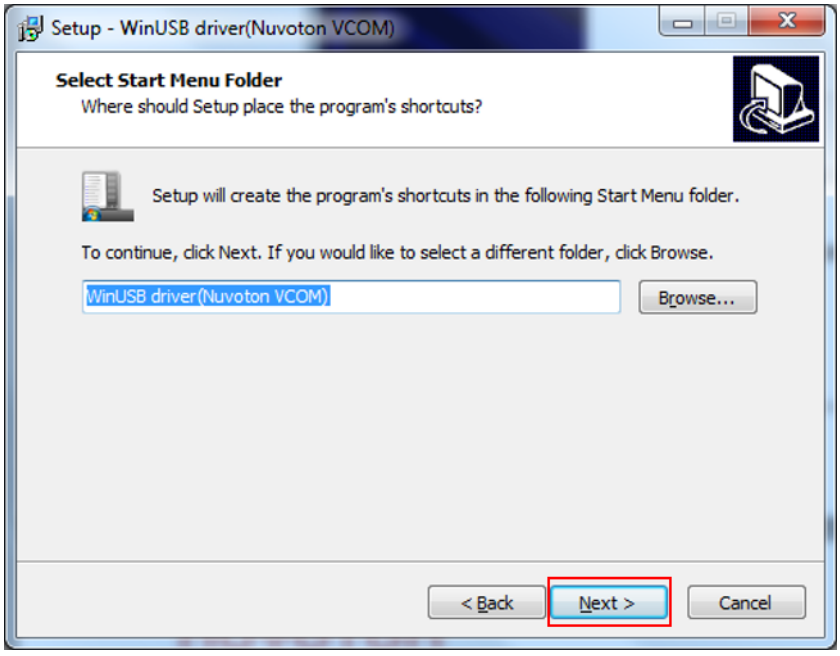


Figure 4-1 Nuvoton USB Driver Installation Setup

Click "**Next**". The WinUSB driver Setup Wizard will be started.


Setup - WinUSB driver(Nuvoton VCOM)
Select Destination Location
Where should WinUSB driver(Nuvoton VCOM) be installed?
Setup will install WinUSB driver(Nuvoton VCOM) into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Program Files\WinUSB4NuVCOM
Browse...
At least 19.8 MB of free disk space is required.
< Back
Next >
Cancel


Setup - WinUSB driver(Nuvoton VCOM)
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
WinUSB driver(Nuvoton VCOM)
Browse...
< Back
Next >
Cancel

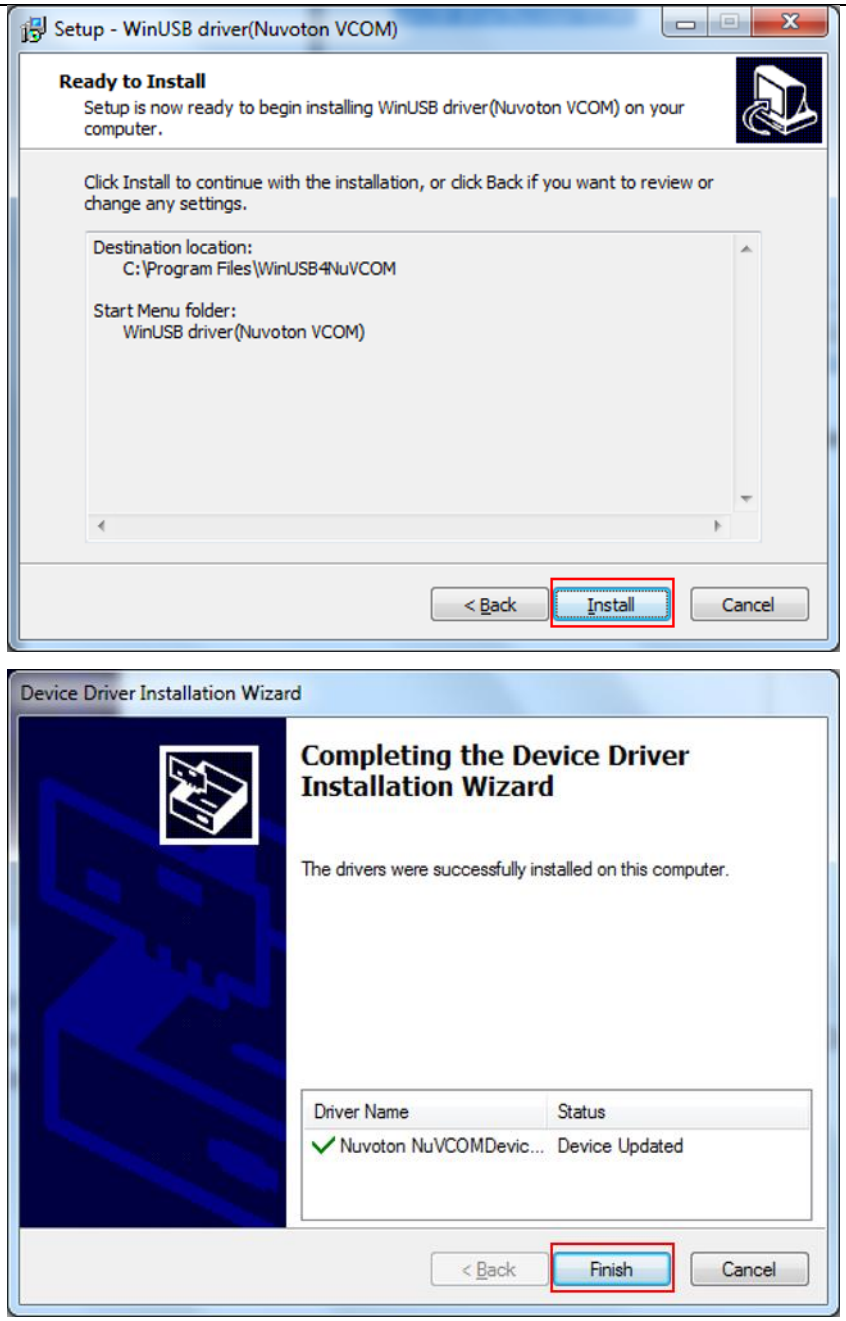


Figure 4-2 Nuvoton USB Driver Installation

The USB serial port function is used to print some messages on PC API, such as SecureCRT, through the standard UART protocol to help user to debug program.

Please download USB CDC driver ” TomatoUSB CDC driver” from Nuvoton’s official webpage, executing the “NuvotonCDC_V1.00.001_Setup.exe” to install the driver:

https://www.nuvoton.com/products/iot-solution/iot-platform/numaker-tomato/?group=Software&tab=2

4.3 Hardware Setting

1. Connect the USB micro-B port (CON5) to the PC HOST.

   The PC HOST will supply 5V power to the NuMaker NUC980 Serial Server and will recognize

the board as a USB composite device.

The VCOM port function is used to print some messages on some Terminal Tools, such as Tera Term, PuTTY, etc. It is through the standard UART protocol to help user to debug.

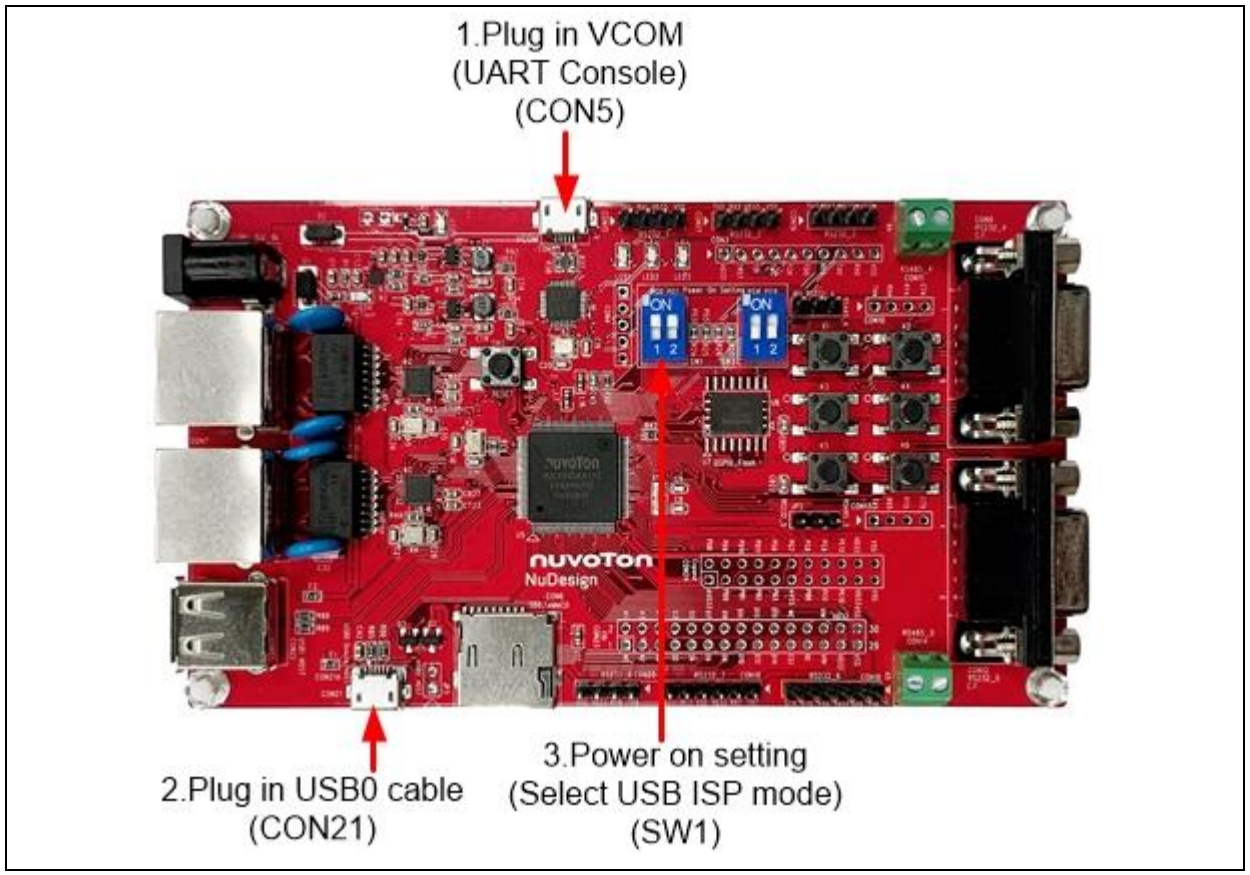


Figure 4-3 Hardware Setting

2. Plug in the USB0 cable (CON21)

   If the installation is successful, a virtual COM port named “WinUSB driver (Nuvoton VCOM)” can be found in the “Device Manager”.

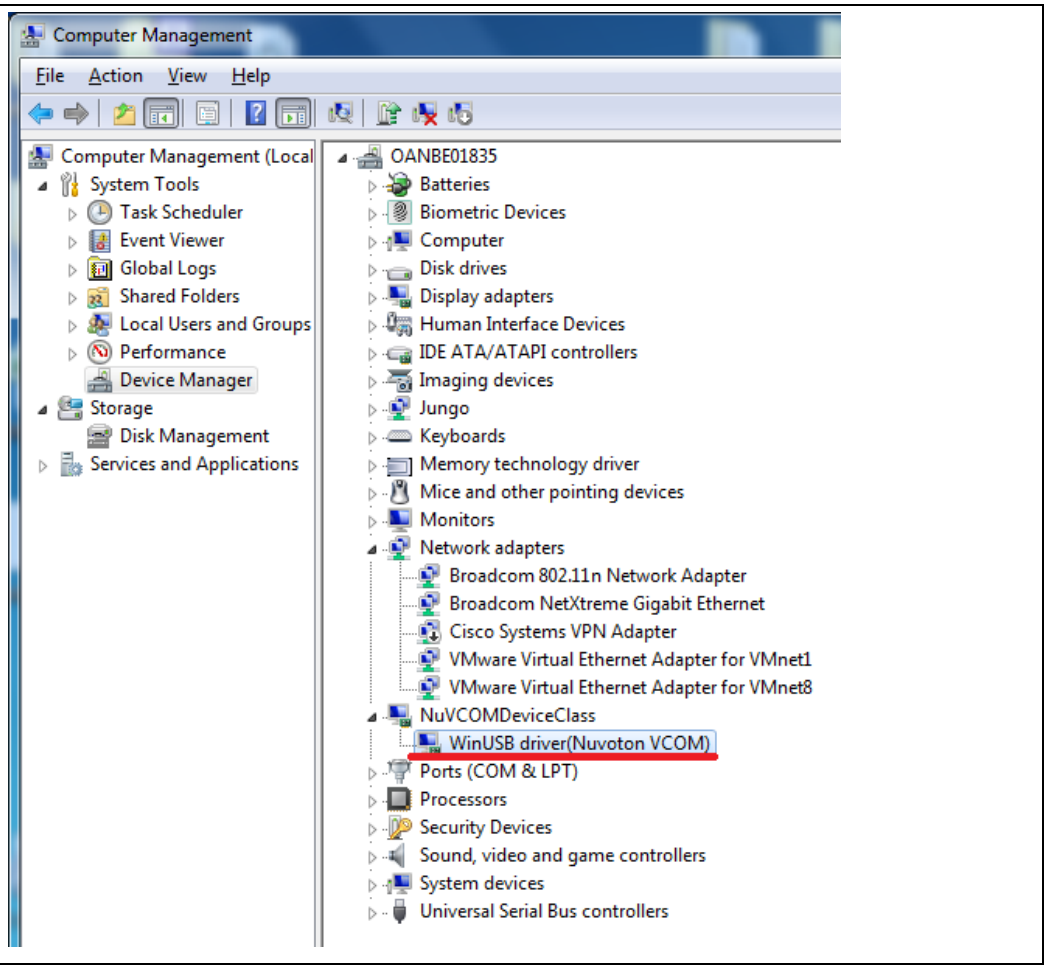


Figure 4-4 Nuvoton VCOM

3. Select the USB mode(SW1) Plug in the USB to UART cable (CON2)

   Set power on setting(SW1) to ON/ON to Boot from USB.

SW	Description (Status and Function)	GPIO pin of NUC980
SW1.2/ SW1.1	**Power on setting** ON/ON = Boot from USB. ON/OFF = Boot from SD/eMMC. OFF/OFF = Boot from QSPI0 Flash.	GPG1/GPG0

Table 4-1 Power On Setting

4. Open the Serial Port Terminal and Reset

   After pressing the reset button(SW2), the MPU will reprogram the application and print out debug message on the terminal.

For detailed NuMaker NUC980 Serial Server introduction, please refer to "**NuDesign NK-980ETH2P User Manual**" in the "Documents" directory.

4.4 Programing Kernel and U-Boot to SPI NAND Flash

1. Install NuWriter Driver. (Please refer to "**NUC980 NuWriter User Manual**")
2. Set SW1(Power On Setting) to Boot from USB(shown in Table 4-1 and Figure 4-3). Connect USB0 to PC and connect UART console to PC.
3. double click "**NuWriter.exe**" on PC. Select target chip as "NUC980 series" and select DDR parameter is "NUC980DK61Y.ini". And then, press "**Continue**" button.

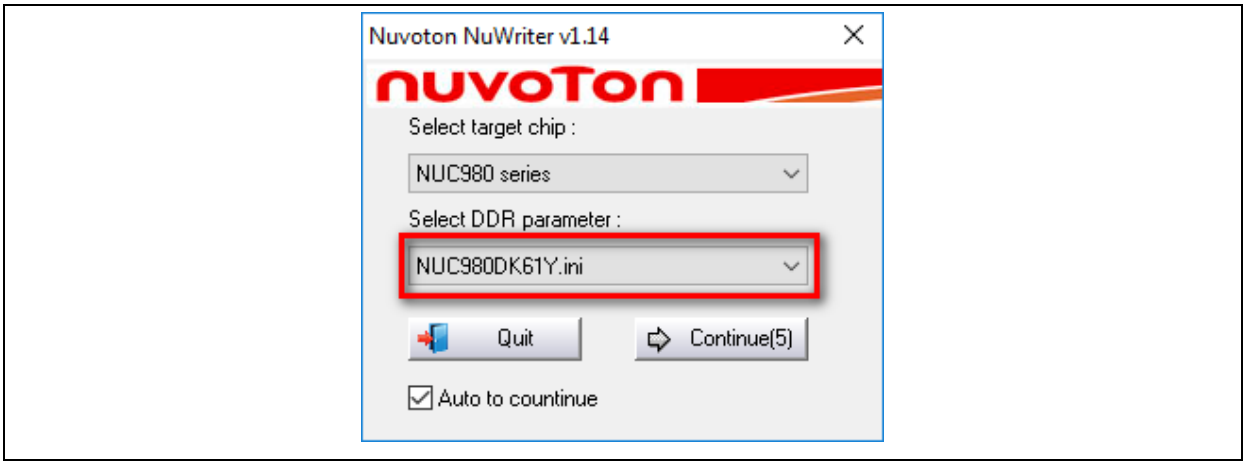


Figure 4-5 NuWriter Setting

4. According to Figure 4-6, following the steps below to program u-boot-spl.bin:
   a. Select the "**SPI NAND**" type.
   b. Fill in the image information：
      - Image Name: u-boot-spl.bin
      - Image Type: Loader
      - Image execute address: 0x200
   c. Click "**Program**".
   d. Waiting for the progress bar to be finished.
   e. After "**Program**" the image, click the "**Verify**" button to read back the image data to make sure the burning status.

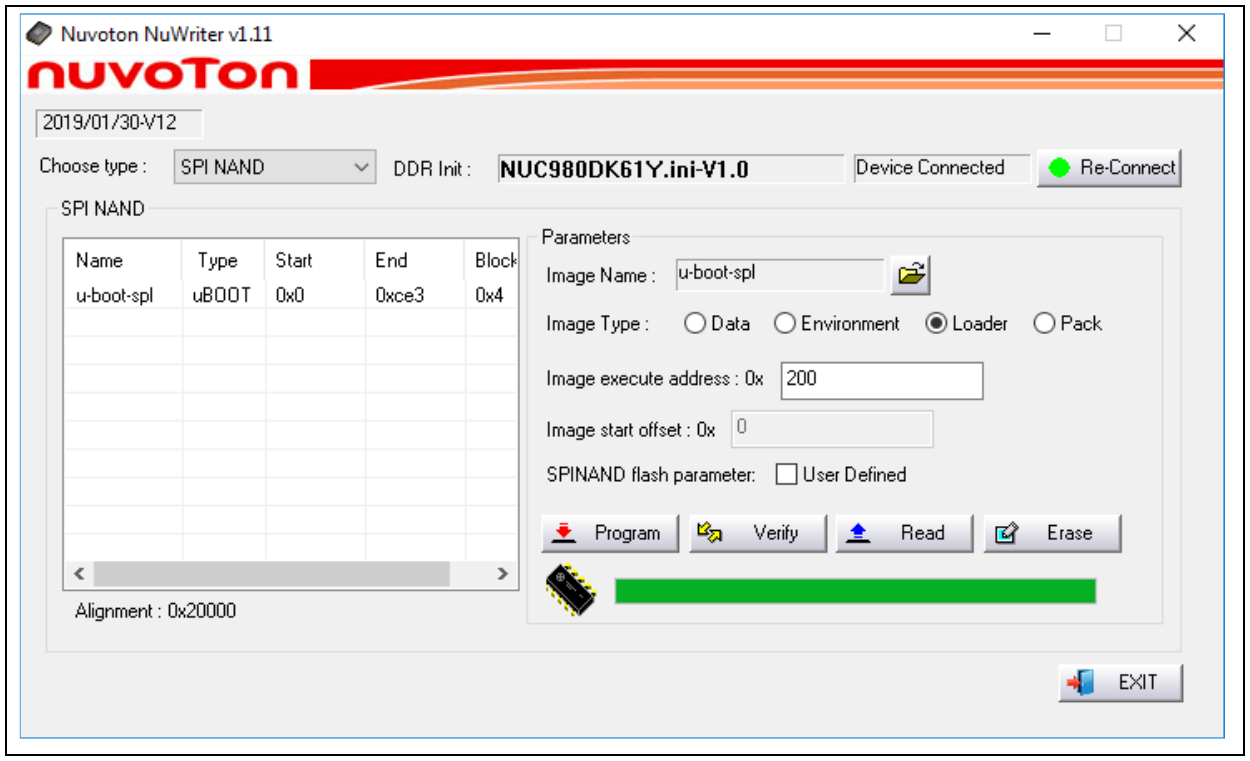


Figure 4-6 Program u-boot-spl

5. According to Figure 4-7, following the steps to program u-boot.bin in the “uboot_v2016.11” folder:

   a. Select the “**SPI NAND**” type.
   b. Fill in the image information：
      - Image Name: u-boot.bin
      - Image Type: Data
      - Image execute address: 0x100000
   c. Click “**Program**”.
   d. Waiting the progress bar to be finished.
   e. After “**Program**” the image, click the “**Verify**” button to read back the image data to make sure the burning status.

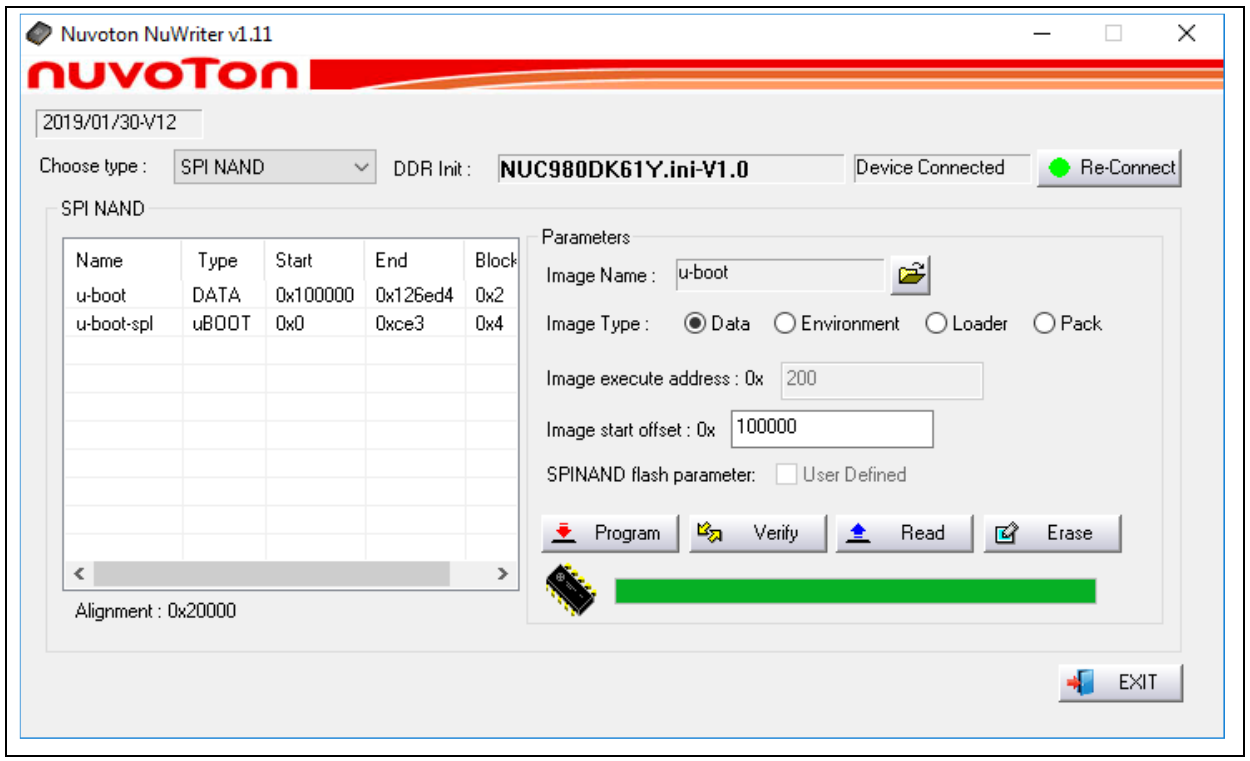


Figure 4-7 Program u-boot

6. According to Figure 4-8, following the steps below to program kernel image:
   a. Select the "**SPI NAND**" type.
   b. Fill in the image information：
      - Image Name: 980uimage.bin
      - Image Type: Data
      - Image start offset address: 0x200000
   c. Click "**Program**".
   d. Waiting for the progress bar to be finished.
   e. After "**Program**" the image, click the "**Verify**" button to read back the image data to make sure the burning status.

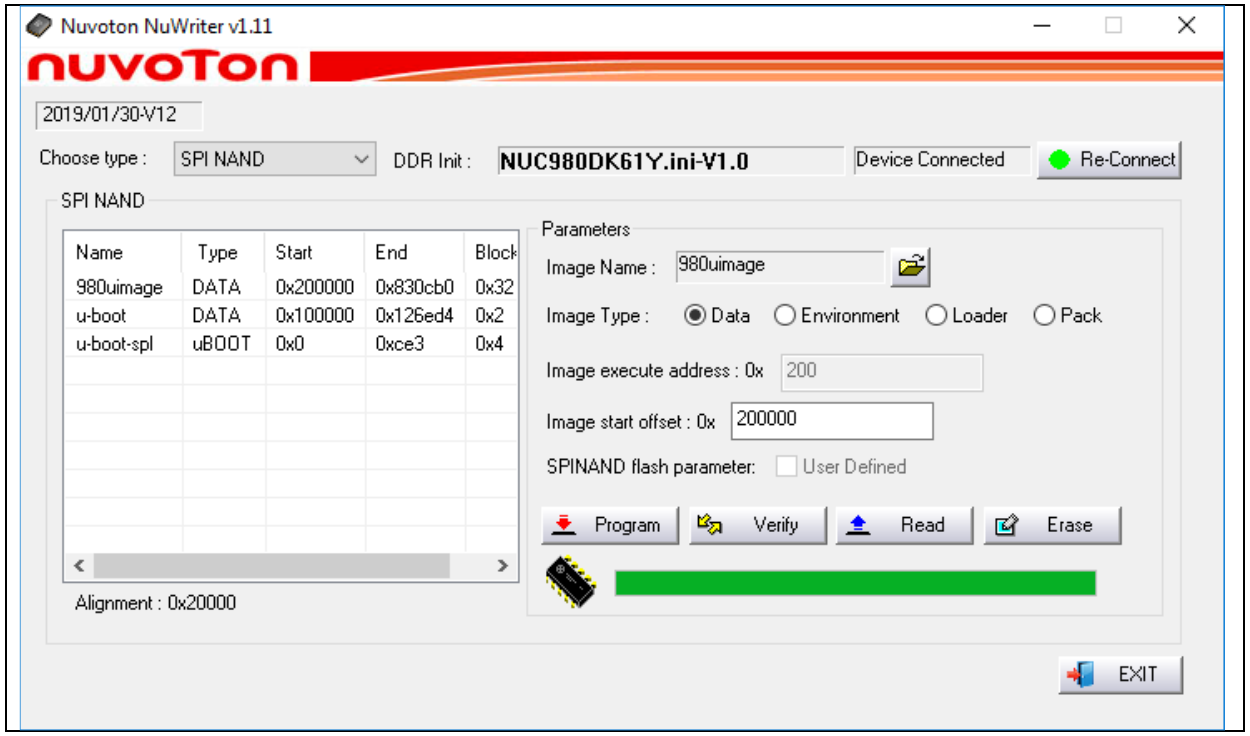


Figure 4-8 Program Kernel Image

For more details about NuWriter tool, please refer to "**NUC980 NuWriter User Manual**" in the "Documents" directory.

For more details about kernel image and uboot, please refer to "**NUC980_970 Linux environment on VMware User Manual**" from Nuvoton website.
URL: https://www.nuvoton.com/products/iot-solution/iot-platform/numaker-server-nuc980/?group=Document&tab=2

4.5 Booting Linux Kernel

This chapter describes how to boot up Linux kernel.

A. Set SW1(Power On Setting) to Boot from QSPI 0 Flash(refer to Table 4-1 and Figure 4-3).

B. Press Reset button on Development Board. From console can find system enter to U-Boot. User can use following commands to launch Linux kernel after enter U-Boot shell.

   1. Type "`sf probe 0 18000000`" to set SPI speed (optinoal)
   2. 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: Lock ops not supported for ef flash
SF: Detected W25N01GV with page size 2 KiB, erase size 128 KiB, total 128
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: Lock ops not supported for ef flash
SF: Detected W25N01GV with page size 2 KiB, erase size 128 KiB, total 128
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 ...
```

A. After boot 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 bitsof
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.
```

```
~ #
```

For the detail Linux kernel compile and setting, please refer to "**NUC980 Linux BSP User Manual**" in the "Documents" directory.

4.6 Executing Sample Code

Please make sure the UART console connect to PC first and follow the steps.

A. Set SW1(Power On Setting) to Boot from QSPI 0 Flash(refer to Table 4-1 and Figure 4-3).

B. Connect UART console port.

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

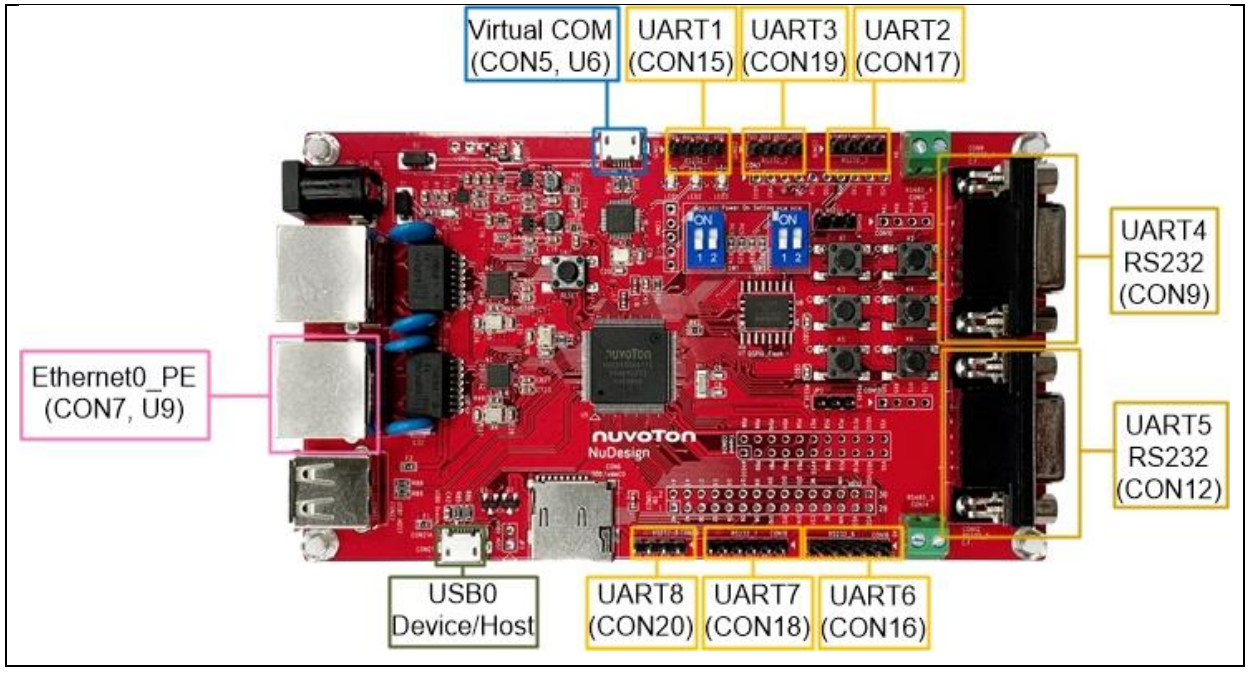


Figure 4-9 NuMaker NUC980 Serial Server Board Setup

Then, use Terminal tool, such as PuTTY, Tera term, etc to open the serial COM port. The COM port configuration is baudrate 115200bps, 8-bit data length and no-parity.

User also need to ensure that the PC Ethernet port is connected to the Development Board Ethernet0 (or Ethernet1) port. PC Internet settings can refer to Figure 4-10.

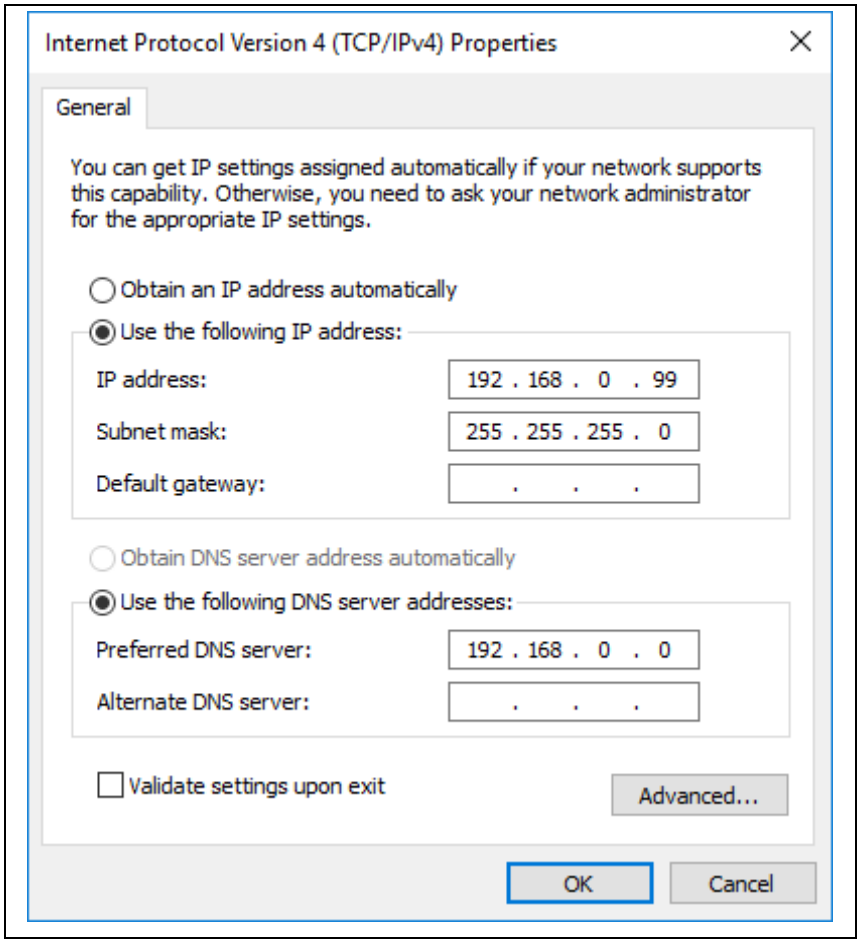


Figure 4-10 Serial COM Port Setting

Power ON or push the Reset key and boot system.

```
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”.

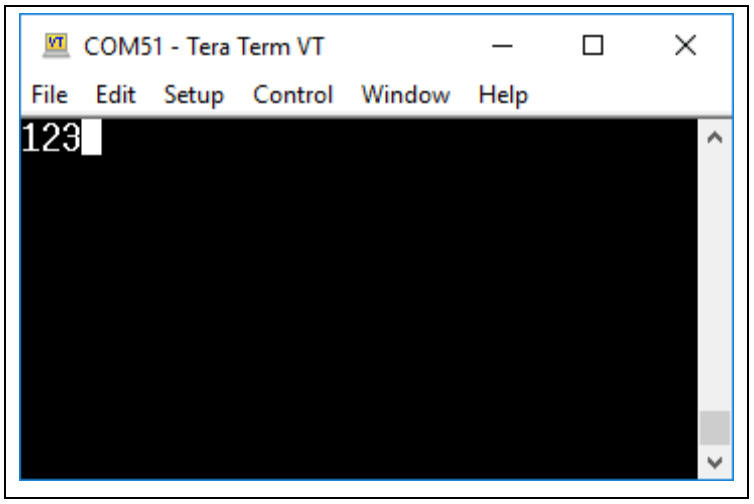


Figure 4-11 Serial COM Port

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

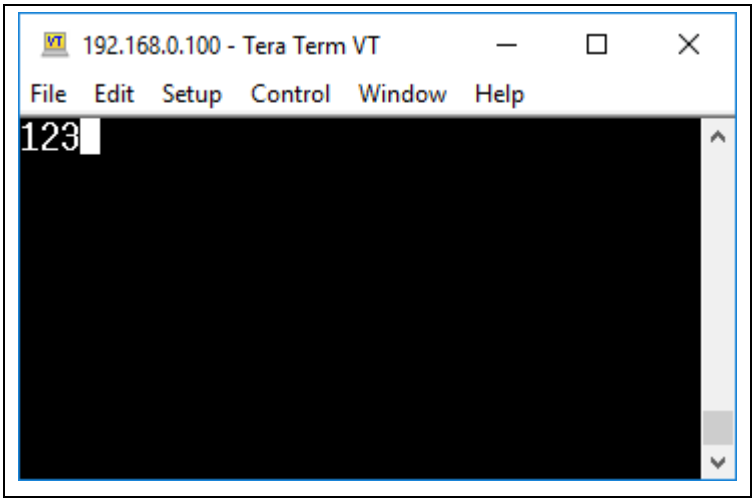


Figure 4-12 TCP/IP Connection Window

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 the UART settings. Including port, baudrate, data length, parity, stop bit, flow control, disable/enable RS485.
3. Press Submit

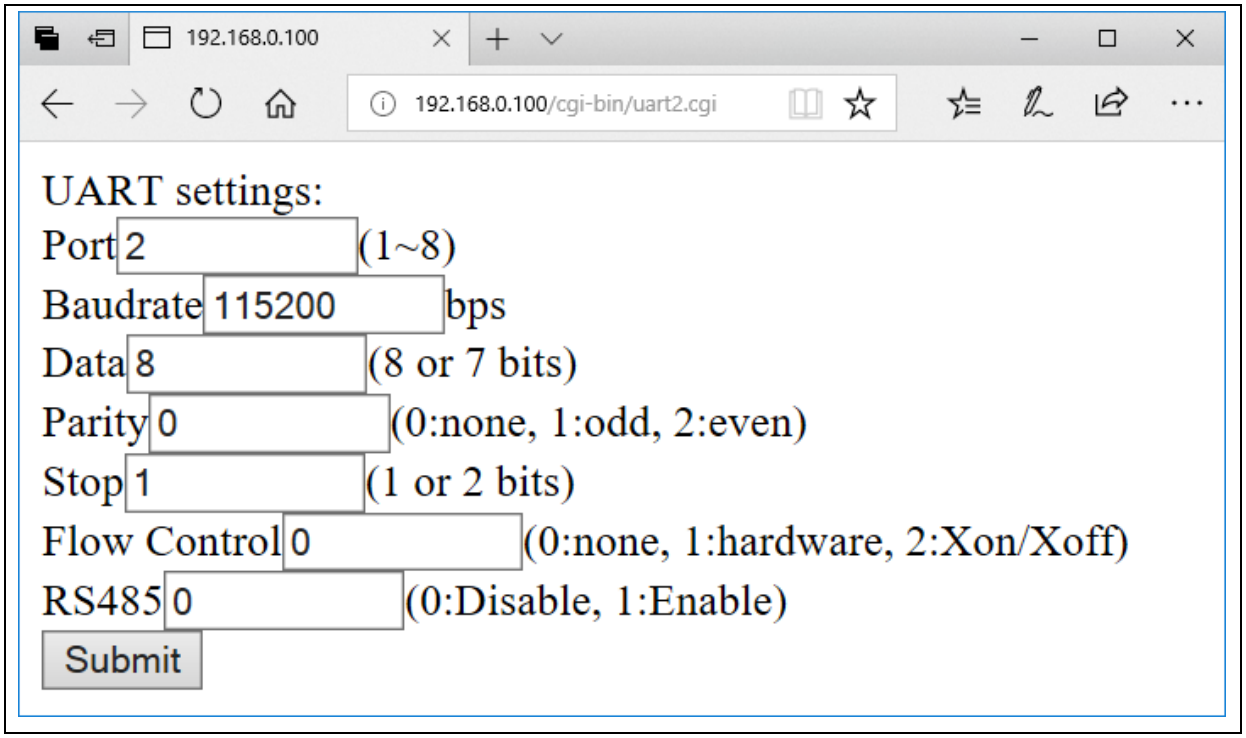
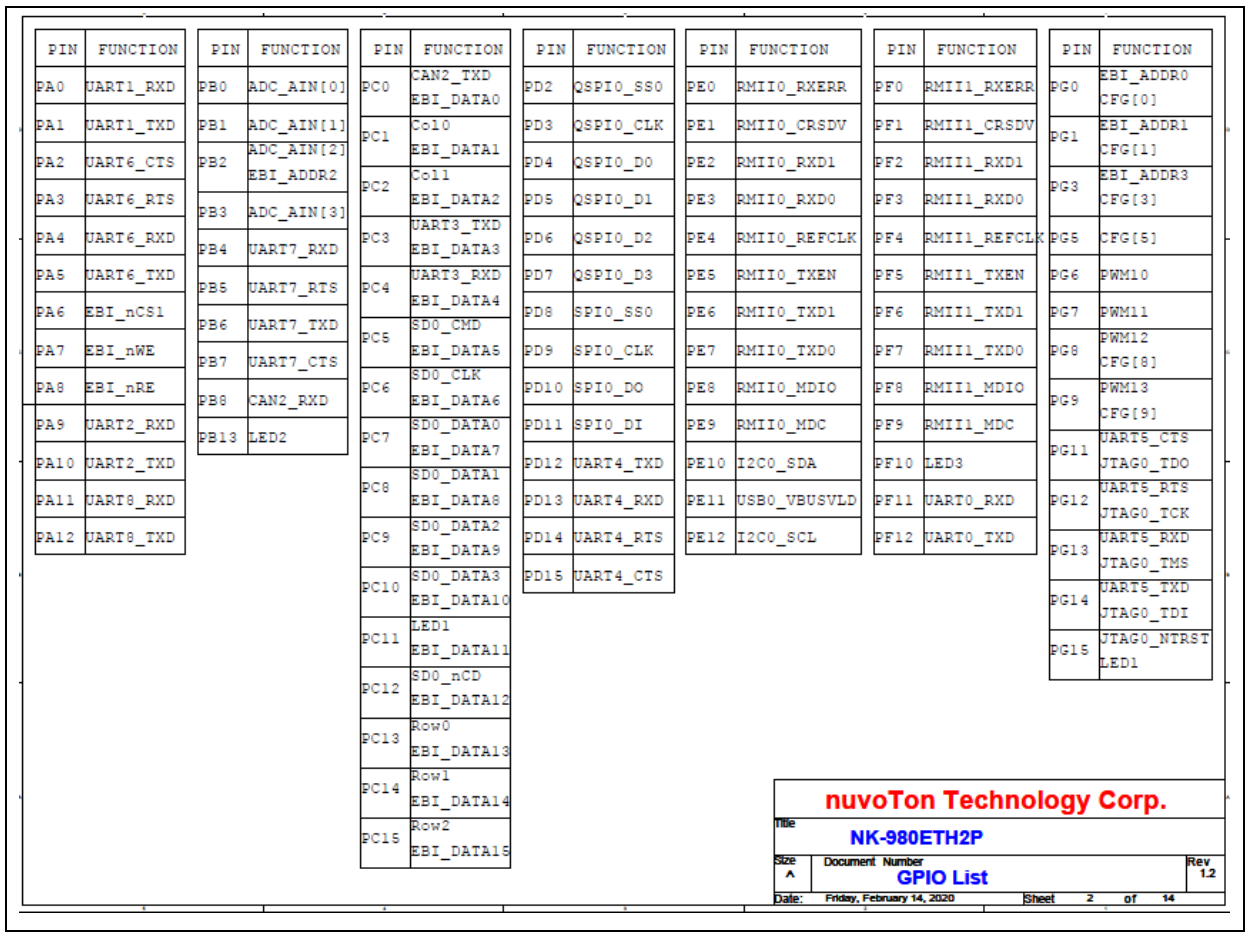


Figure 4-13 UART Setting Web Page

5 BLOCK DIAGRAM SCHEMATIC

5.1 GPIO List Schematic



PIN	FUNCTION
PA0	UART1_RXD
PA1	UART1_TXD
PA2	UART6_CTS
PA3	UART6_RTS
PA4	UART6_RXD
PA5	UART6_TXD
PA6	EBI_nCS1
PA7	EBI_nWE
PA8	EBI_nRE
PA9	UART2_RXD
PA10	UART2_TXD
PA11	UART8_RXD
PA12	UART8_TXD

PIN	FUNCTION
PB0	ADC_AIN[0]
PB1	ADC_AIN[1]
PB2	ADC_AIN[2] EBI_ADDR2
PB3	ADC_AIN[3]
PB4	UART7_RXD
PB5	UART7_RTS
PB6	UART7_TXD
PB7	UART7_CTS
PB8	CAN2_RXD
PB13	LED2

PIN	FUNCTION
PC0	CAN2_TXD EBI_DATA0
PC1	Col0 EBI_DATA1
PC2	Col1 EBI_DATA2
PC3	UART3_TXD EBI_DATA3
PC4	UART3_RXD EBI_DATA4
PC5	SD0_CMD EBI_DATA5
PC6	SD0_CLK EBI_DATA6
PC7	SD0_DATA0 EBI_DATA7
PC8	SD0_DATA1 EBI_DATA8
PC9	SD0_DATA2 EBI_DATA9
PC10	SD0_DATA3 EBI_DATA10
PC11	LED1 EBI_DATA11
PC12	SD0_nCD EBI_DATA12
PC13	Row0 EBI_DATA13
PC14	Row1 EBI_DATA14
PC15	Row2 EBI_DATA15

PIN	FUNCTION
PD2	QSPI0_SS0
PD3	QSPI0_CLK
PD4	QSPI0_D0
PD5	QSPI0_D1
PD6	QSPI0_D2
PD7	QSPI0_D3
PD8	SPI0_SS0
PD9	SPI0_CLK
PD10	SPI0_DO
PD11	SPI0_DI
PD12	UART4_TXD
PD13	UART4_RXD
PD14	UART4_RTS
PD15	UART4_CTS

PIN	FUNCTION
PE0	RMII0_RXERR
PE1	RMII0_CRSDV
PE2	RMII0_RXD1
PE3	RMII0_RXD0
PE4	RMII0_REFCLK
PE5	RMII0_TXEN
PE6	RMII0_TXD1
PE7	RMII0_TXD0
PE8	RMII0_MDIO
PE9	RMII0_MDC
PE10	I2C0_SDA
PE11	USB0_VBUSVLD
PE12	I2C0_SCL

PIN	FUNCTION
PF0	RMII1_RXERR
PF1	RMII1_CRSDV
PF2	RMII1_RXD1
PF3	RMII1_RXD0
PF4	RMII1_REFCLK
PF5	RMII1_TXEN
PF6	RMII1_TXD1
PF7	RMII1_TXD0
PF8	RMII1_MDIO
PF9	RMII1_MDC
PF10	LED3
PF11	UART0_RXD
PF12	UART0_TXD

PIN	FUNCTION
PG0	EBI_ADDR0 CFG[0]
PG1	EBI_ADDR1 CFG[1]
PG3	EBI_ADDR3 CFG[3]
PG5	CFG[5]
PG6	PWM10
PG7	PWM11
PG8	PWM12 CFG[8]
PG9	PWM13 CFG[9]
PG11	UART5_CTS JTAG0_TDO
PG12	UART5_RTS JTAG0_TCK
PG13	UART5_RXD JTAG0_TMS
PG14	UART5_TXD JTAG0_TDI
PG15	JTAG0_NTRST LED1

Figure 5-1 GPIO List Schematic

5.2 Power Schematic

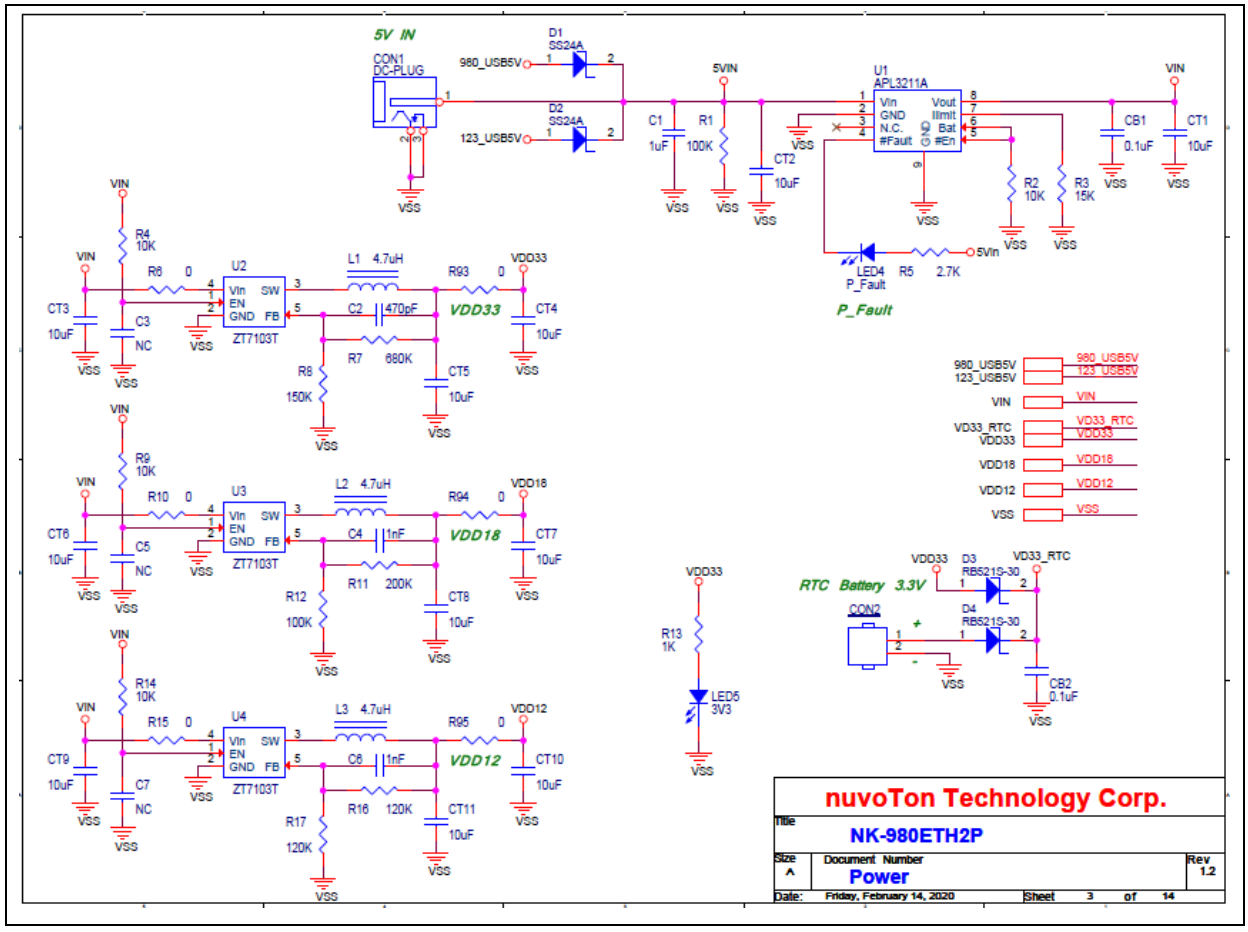


Figure 5-2 Power Schematic

5.3 NUC980DK Schematic

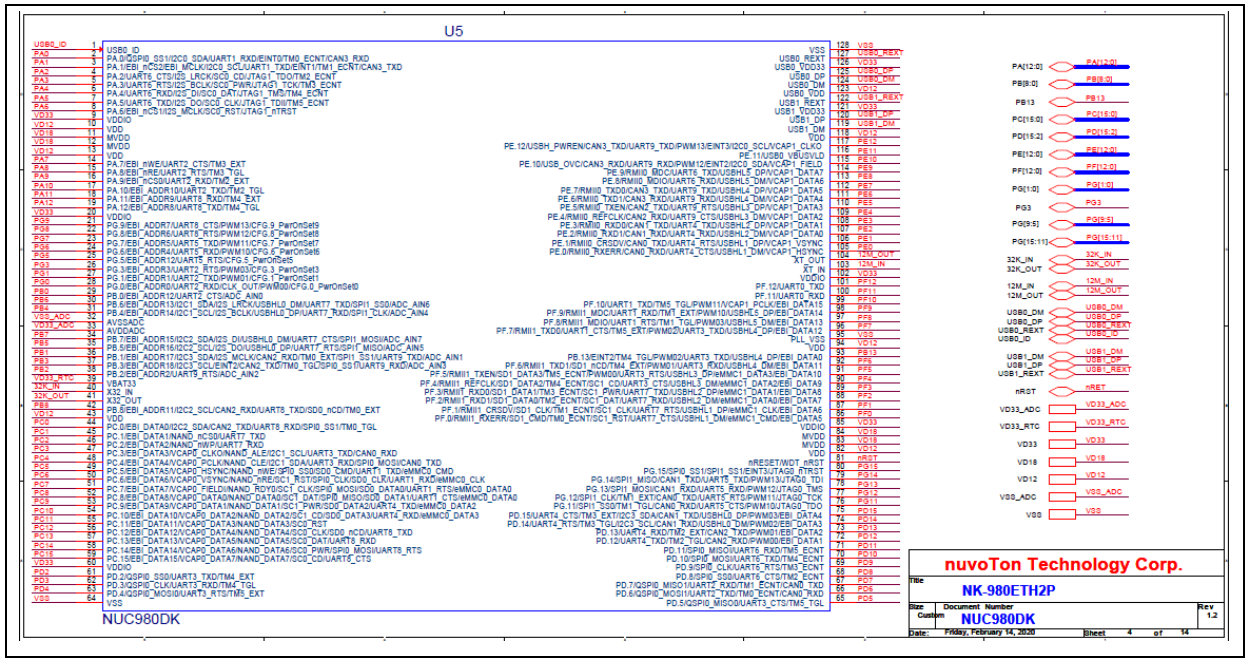


Figure 5-3 NUC980DK Schematic

5.4 Power Filter Schematic

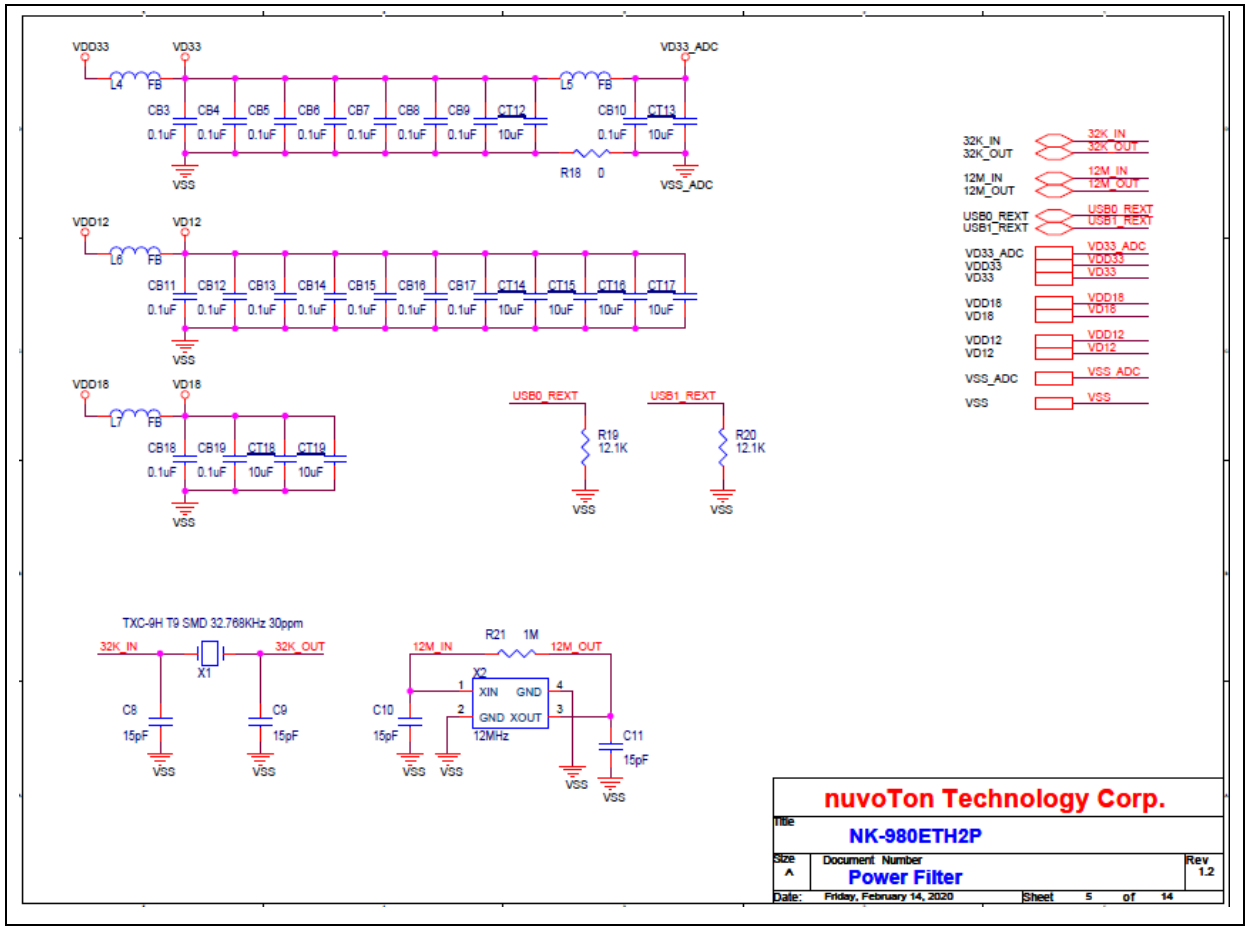


Figure 5-4 Power Filter Schematic

5.5 Configure Schematic

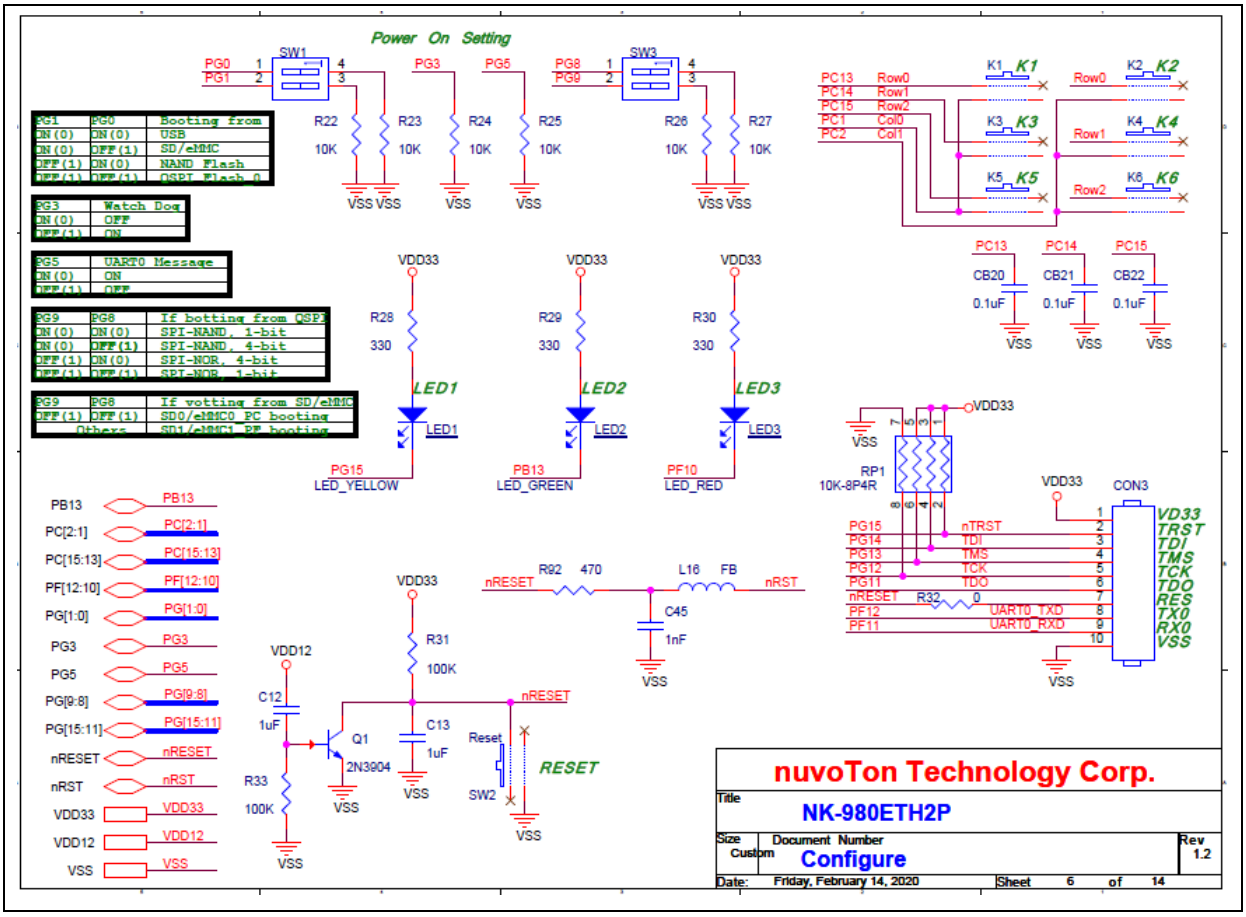


Figure 5-5 Configure Schematic

5.6 NUC123ZD4AN0 Schematic

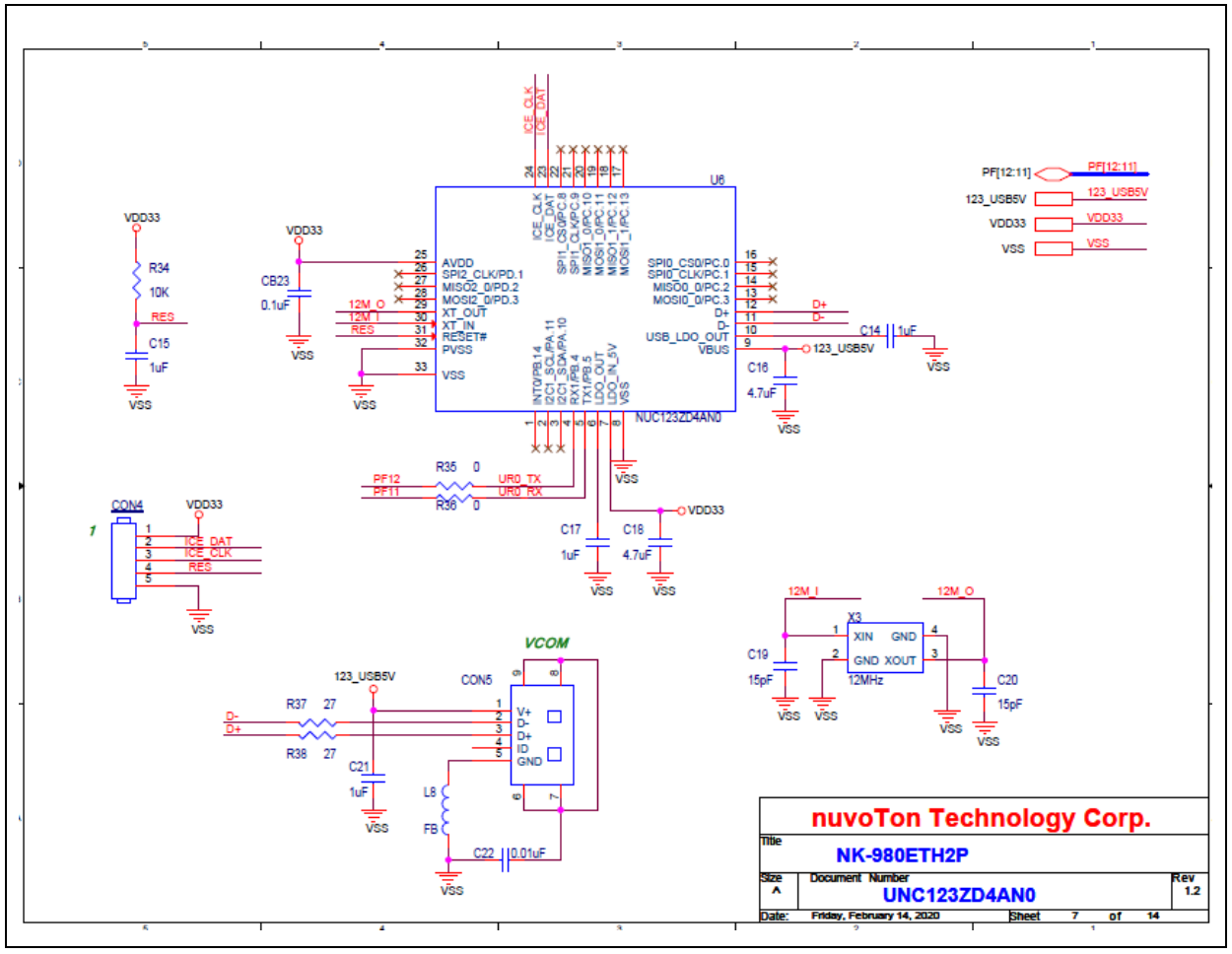


Figure 5-6 NUC123ZD4AN0 Schematic

5.7 Memory Schematic

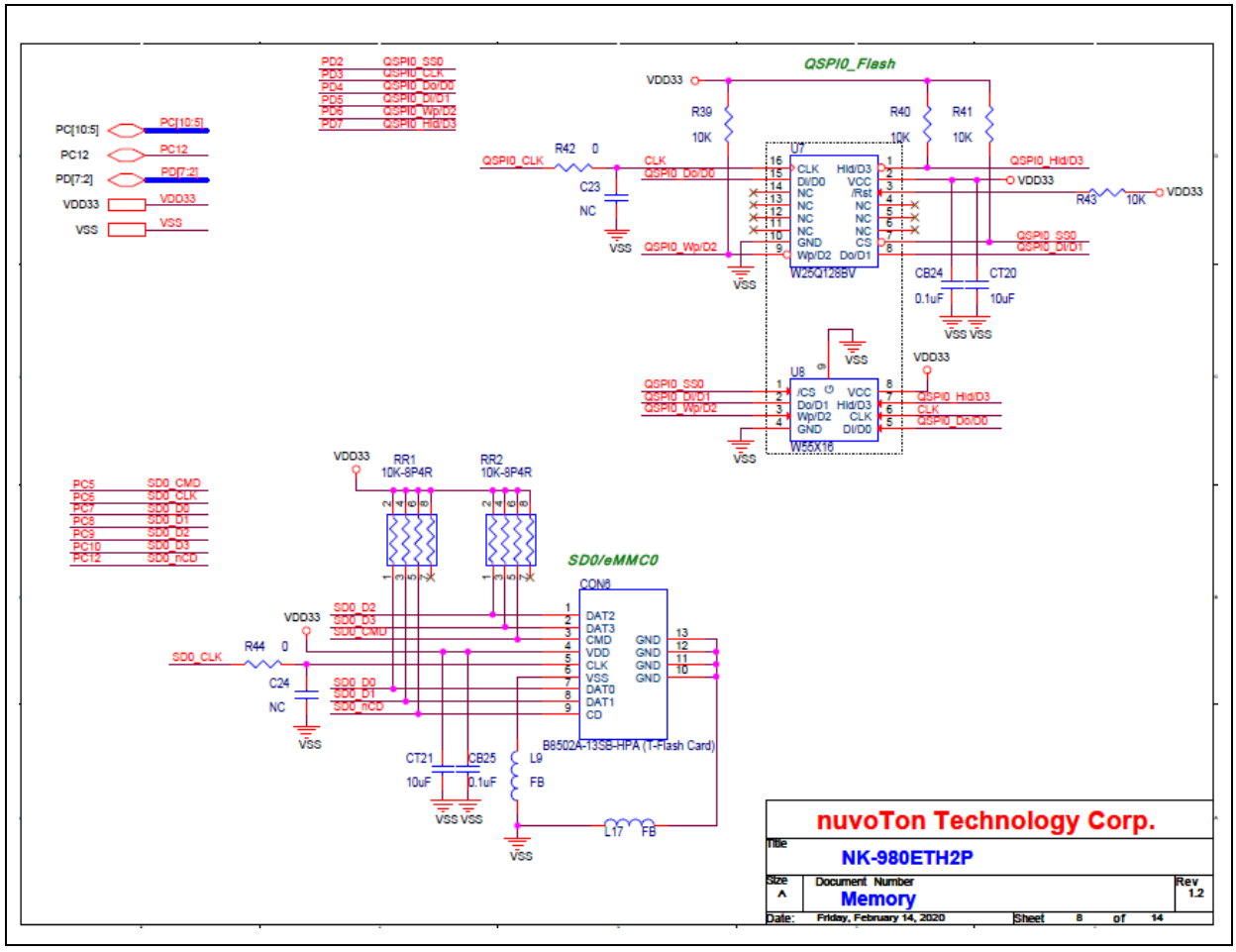


Figure 5-7 Memory Schematic

5.8 RMII_PE Schematic

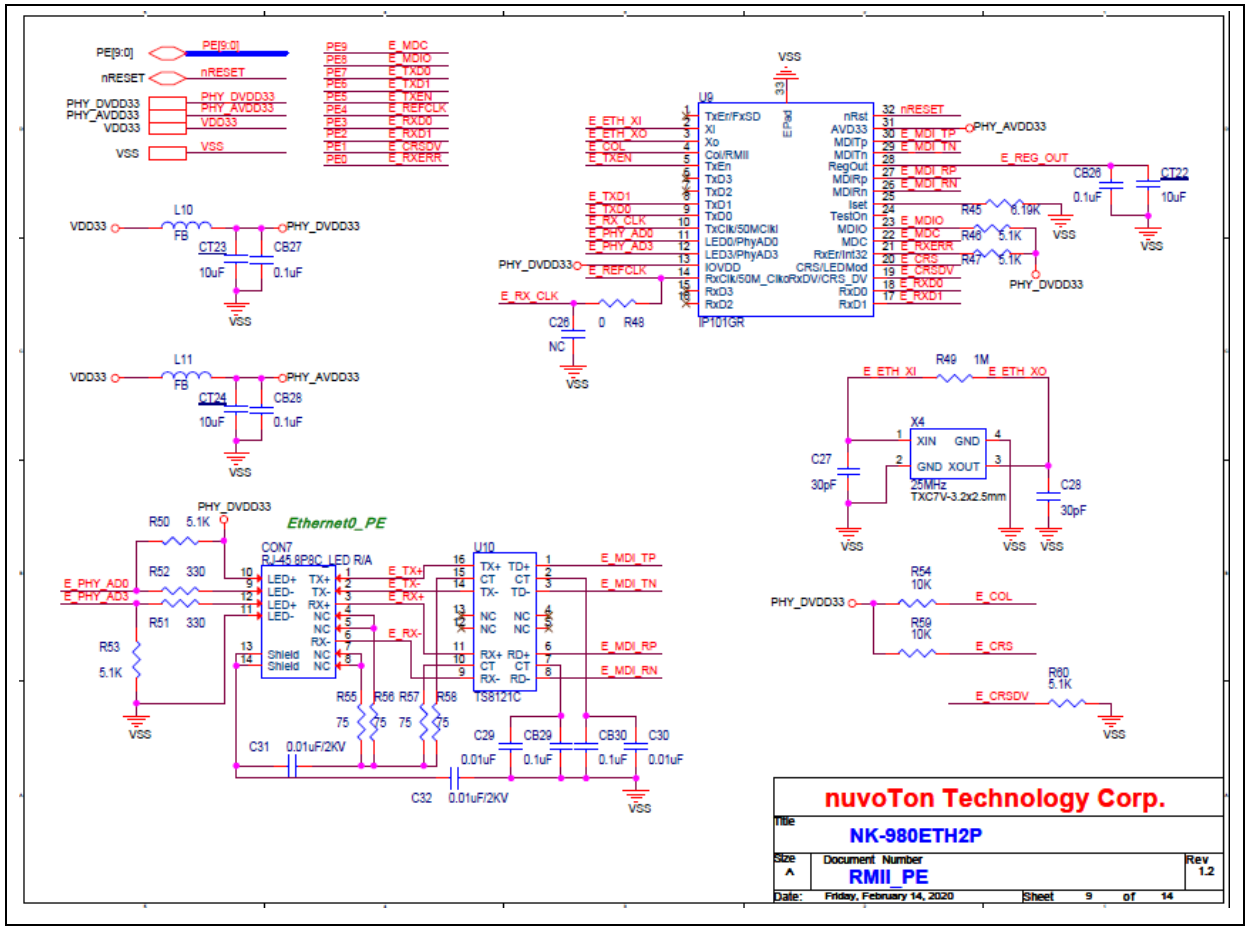


Figure 5-8 RMII_PE Schematic

5.9 RMII_PF Schematic

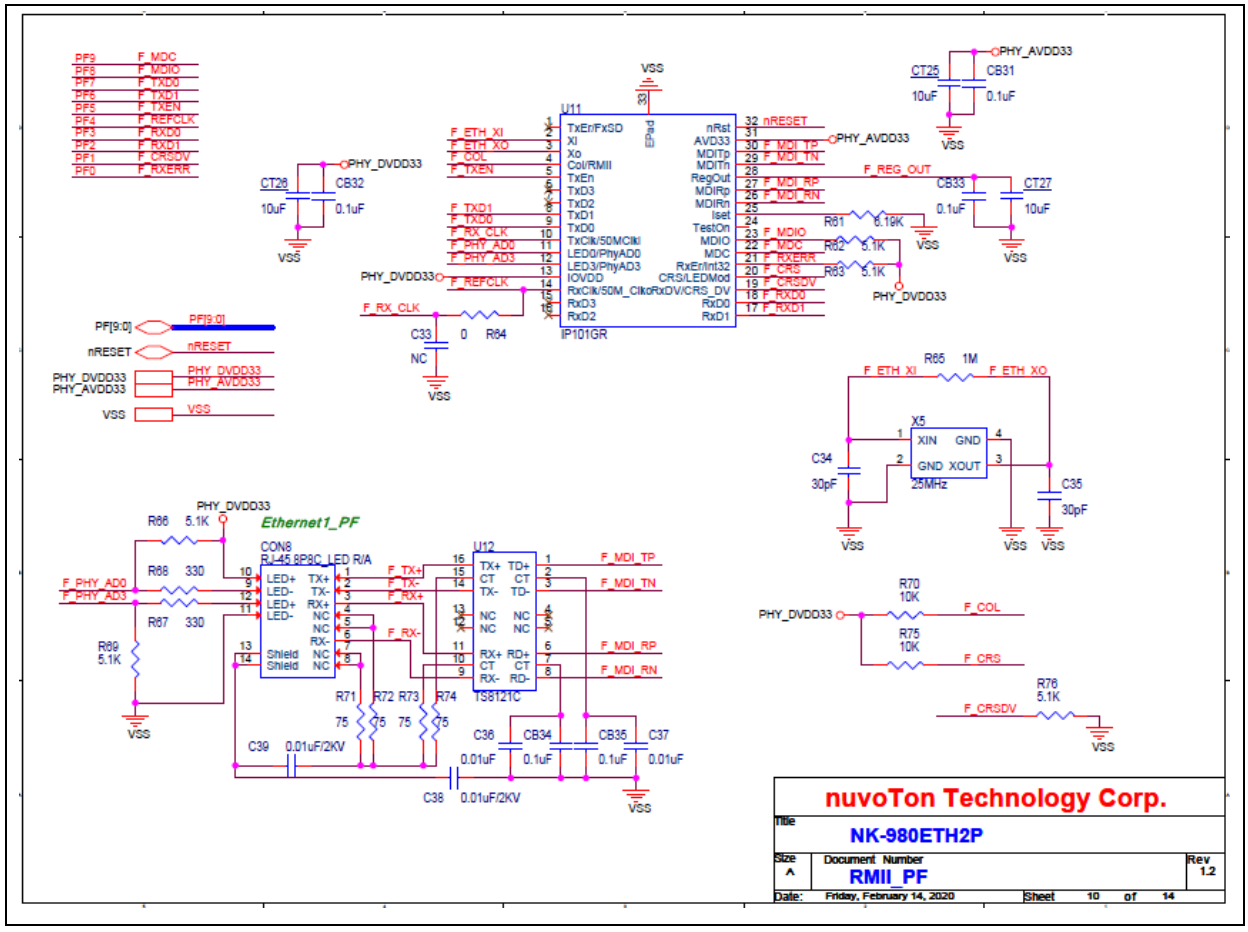


Figure 5-9 RMII_PF Schematic

5.10 UART_A Schematic

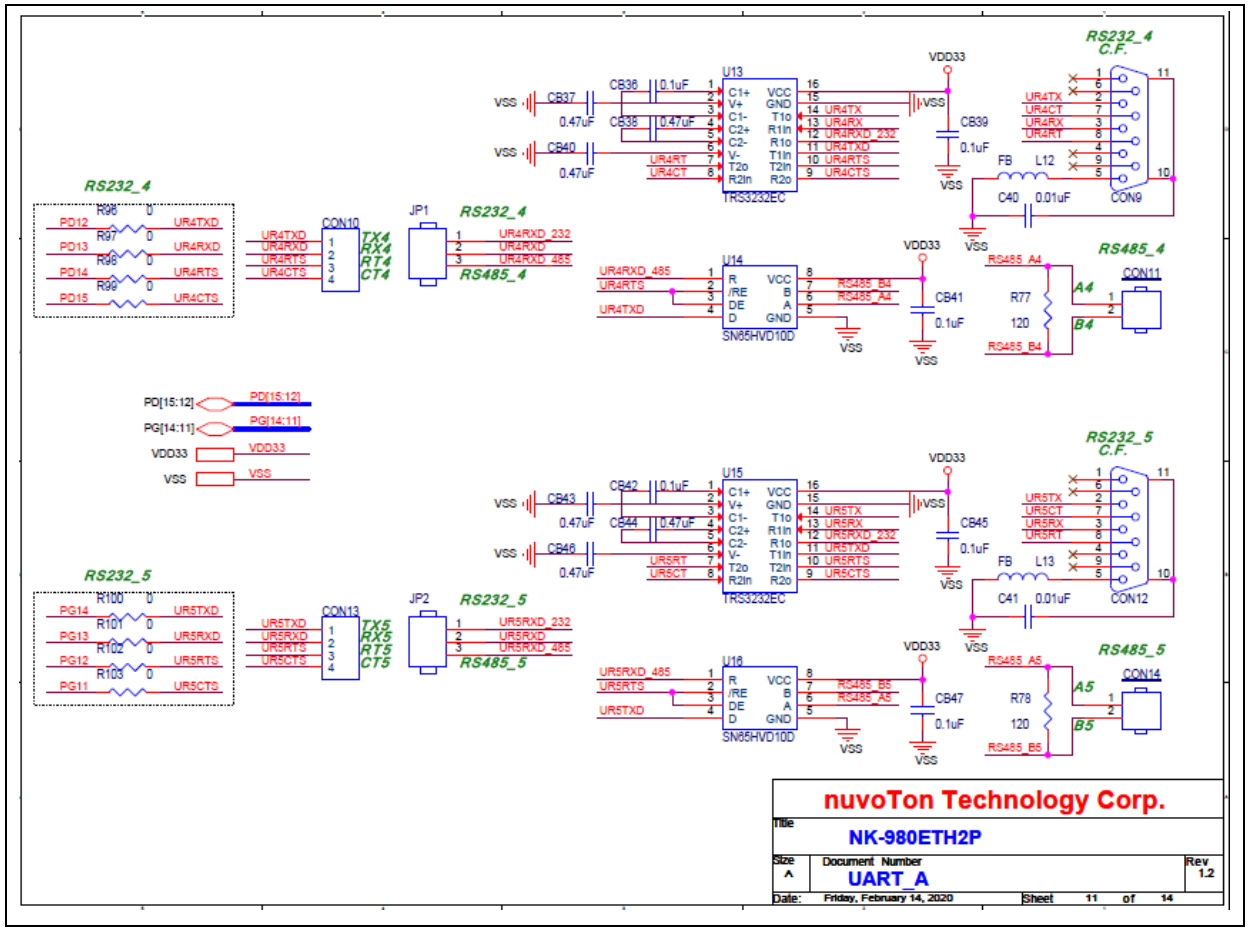


Figure 5-10 UART_A Schematic

5.11 UART_B Schematic

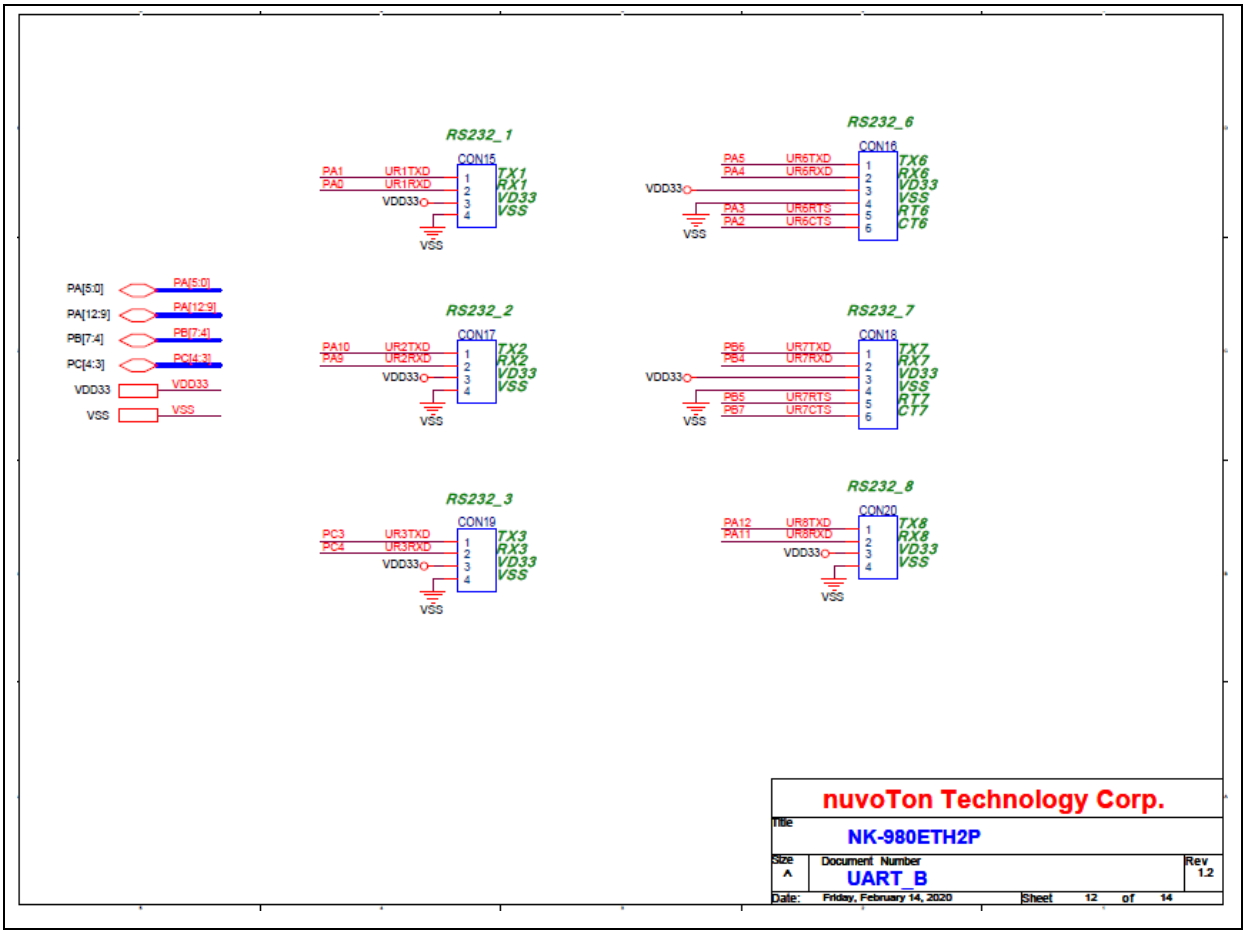


Figure 5-11 UART_B Schematic

5.12 USB Schematic

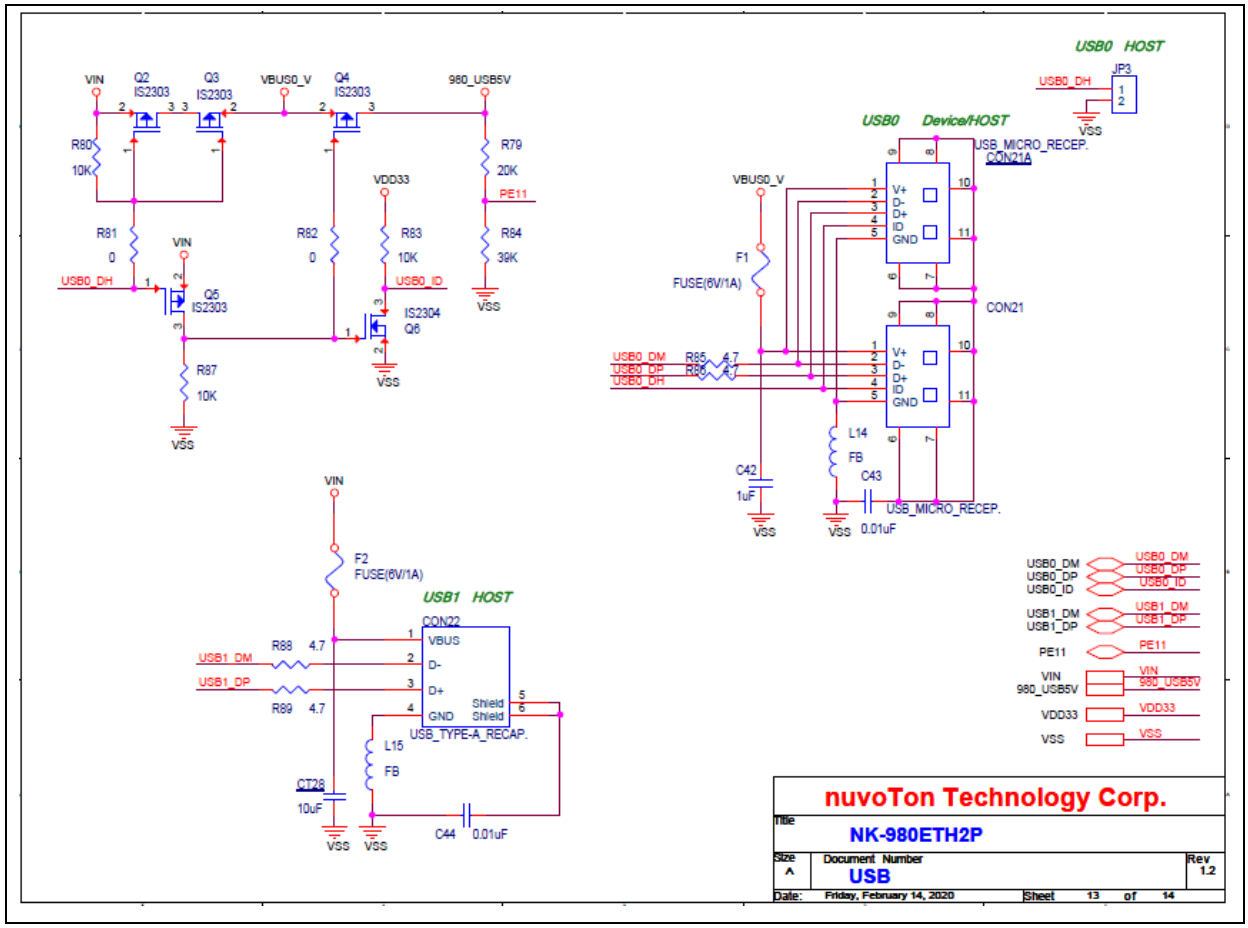


Figure 5-12 USB Schematic

5.13 Expand Schematic

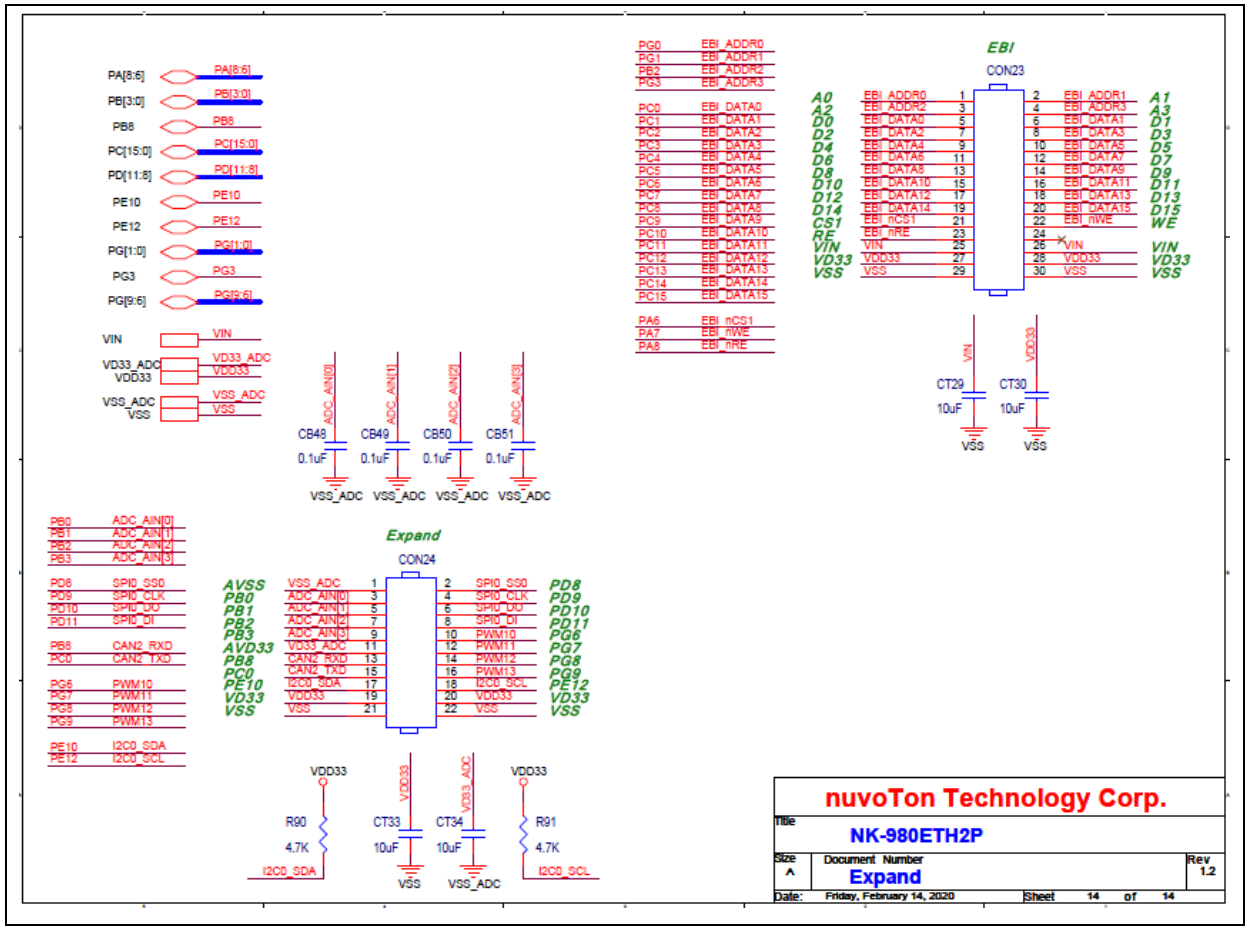


Figure 5-13 Expand Schematic

5.14 PCB Placement

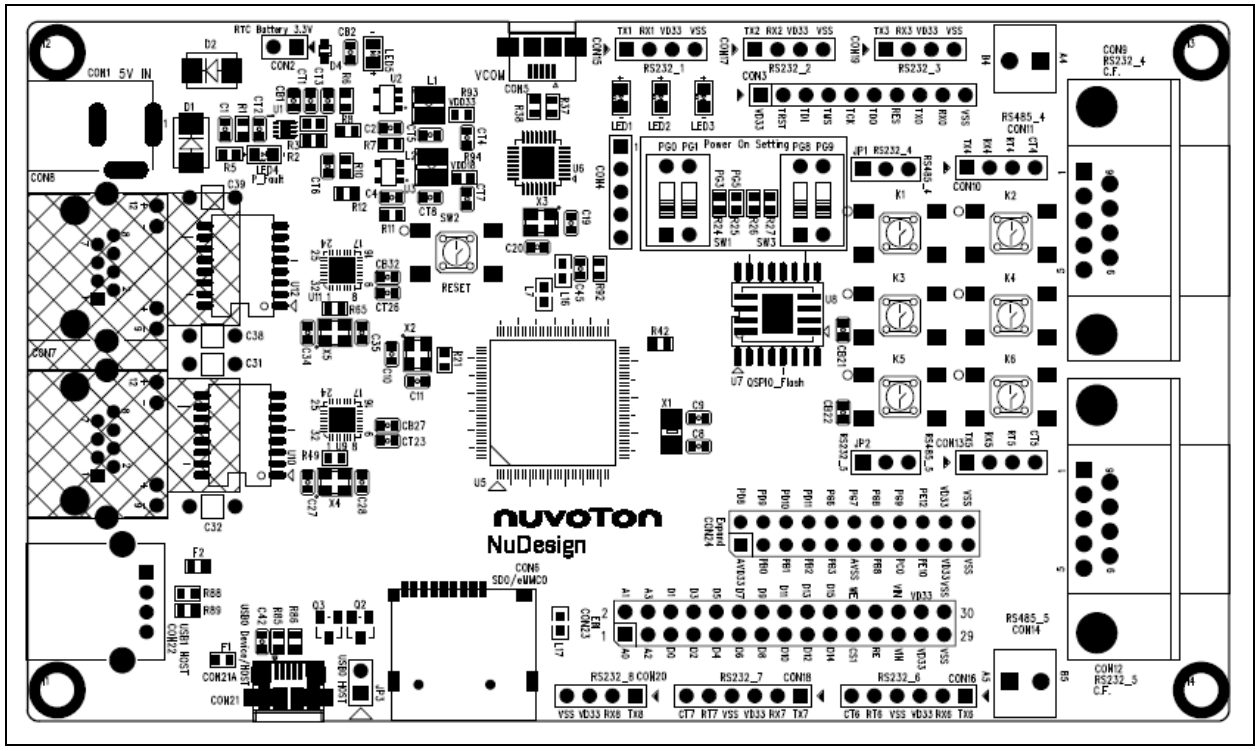


Figure 5-14 Front PCB Placement

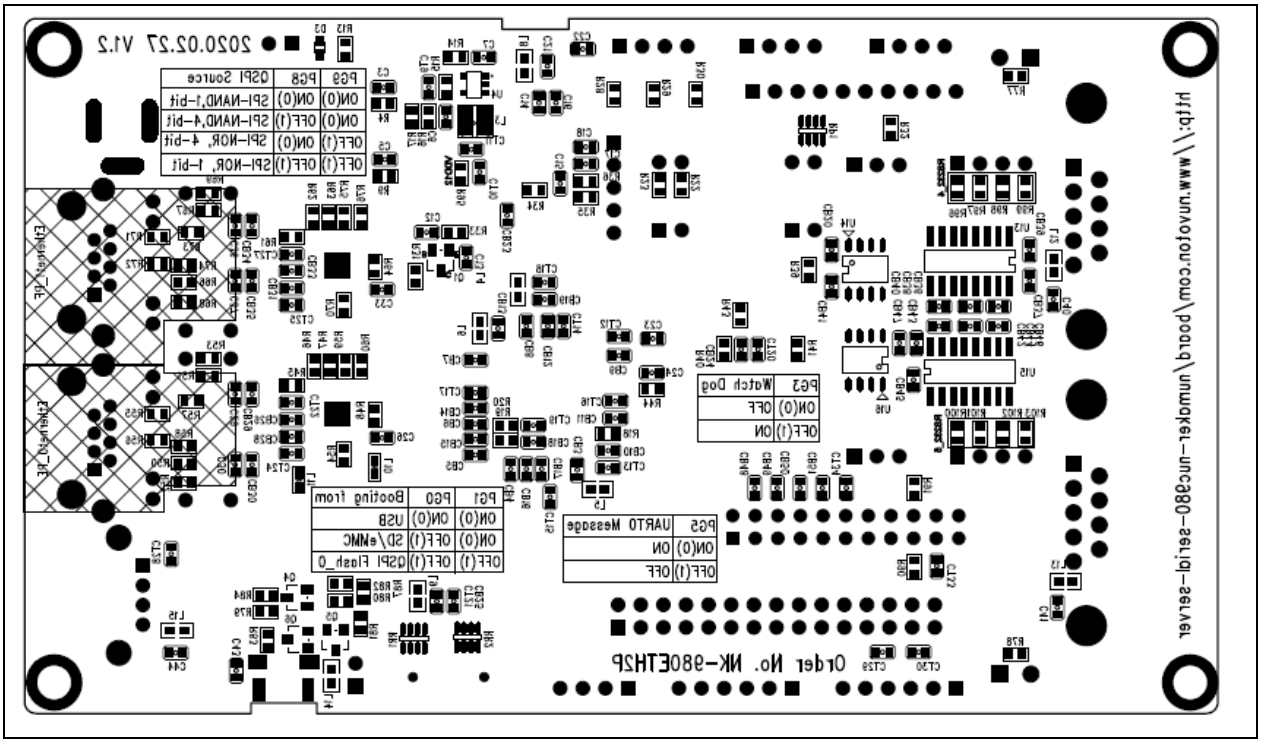


Figure 5-15 Back PCB Placement

6 REVISION HISTORY

Date	Revision	Description
2020.05.22	1.00	1. Initial version
2020.07.22	1.20	2. For board versionV1.2

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.