NuMicro® Family

ARM926EJ-S™ Based 32-bit Microprocessor

NUMAKER-RTU-NUC980(Chili)
User Manual

Evaluation Board for NuMicro® NUC980 Series

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

www.nuvoton.com

Table of Contents

List of Figures

List of Tables

1 OVERVIEW

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

The NUMAKER-RTU-NUC980(Chili) board uses NUC980DR63YC microprocessor (MPU) which runs up to 300 MHz with built-in 64MB DDR2 memory, 16 KB I-cache, 16 KB D-cache and MMU, 16 KB embedded SRAM and 16.5 KB IBR (Internal Boot ROM) for system booting from USB and SPI Flash. All functions of the NUC980DR63YC are placed on the board, including peripheral interfaces such as SPI Flash memory, UART, 10/100 Mb Ethernet MAC controller, high speed USB (Device, Host), JTAG, RS485 and CAN transceiver controller. Users can use it to develop and verify applications to emulate the real behavior.

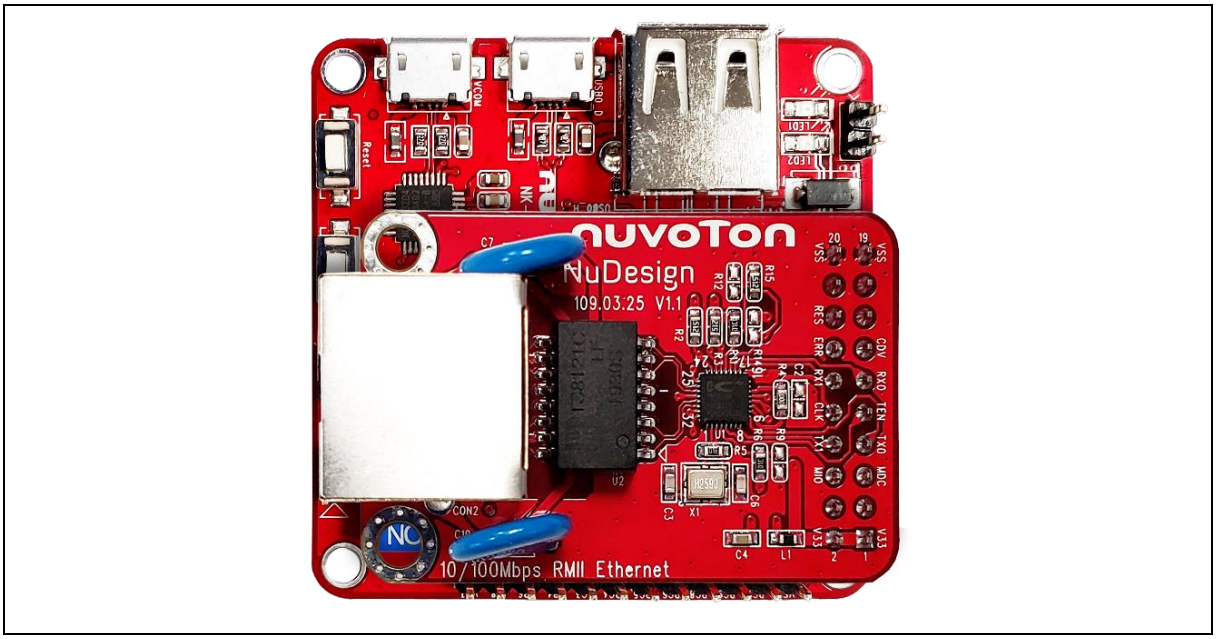


Figure 1-1 NUMAKER-RTU-NUC980(Chili) Development Board

2 FEATURES

- NUC980DR63YC: LQFP64 pin MCP package with DDR2 (64 MB), which can run up to 300 MHz operating speed
- SPI Flash: Normal mode system booting or data storage, use W25Q256JV SPI-NOR (256 M-Bit)
- UART0: Connected to Virtual COM port for system development, debug message output
- Peripheral interface connector, including UART, SPI, I^2C
- JTAG interface provided for software development
- RJ45 port (Ethernet0) connector
- UART8-RS485 header with transceiver controller interface
- CAN3 header with transceiver controller interface
- 2 sets of LED for status indication
- 1 set of user-configurable push button keys
- 1 set of system-reset push button keys
- USB port-0 that can be used as Device/HOST to support pen drives, keyboards, mouse and printers
- 3.3V I/O power, 1.8V Memory power and 1.2V core power

3 HARDWARE CONFIGURATION

3.1 Front View

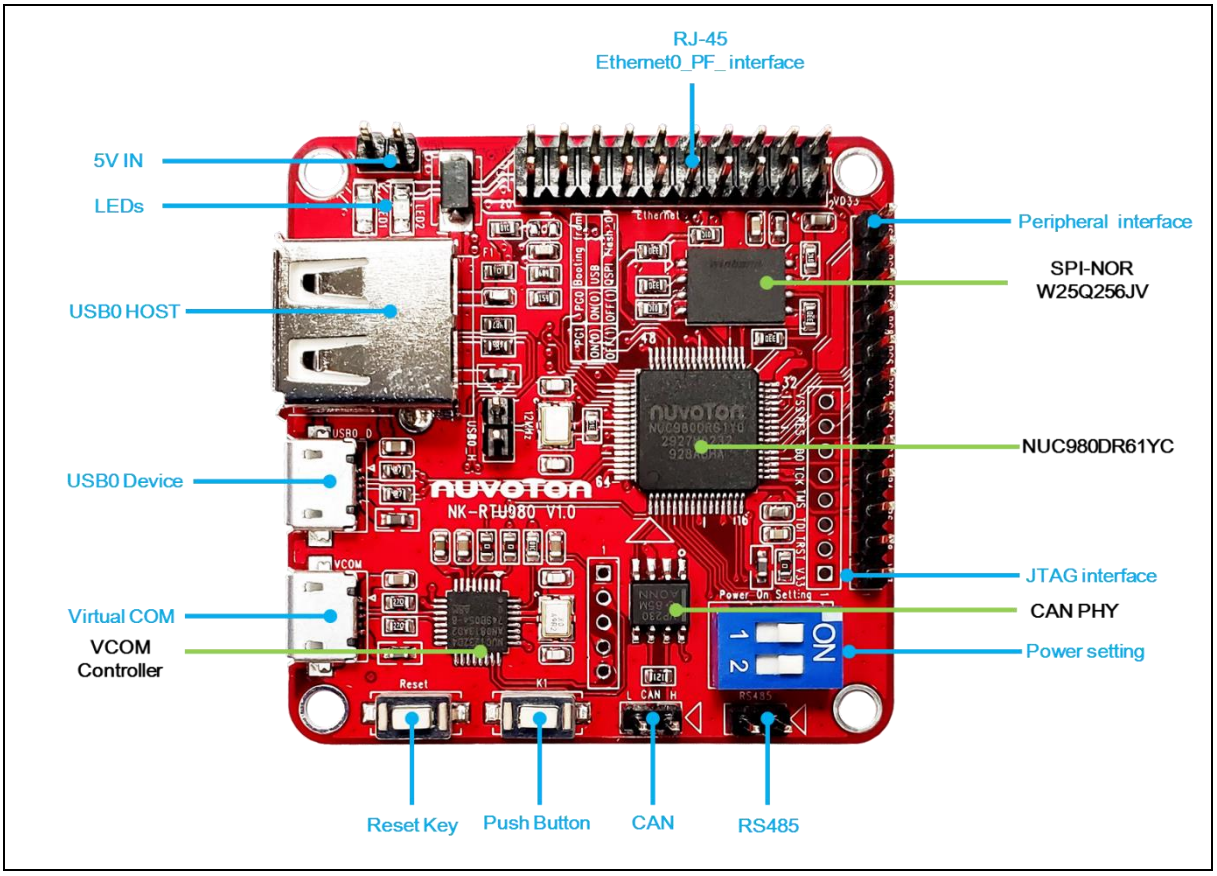


Figure 3-1 Front View of NUMAKER-RTU-NUC980(Chili)

Figure 3-1 shows the main components from the front view of NUMAKER-RTU-NUC980(Chili) Development Board

- +5V In (J1): Power 5V input

Power Model	CON2 USB Port (Micro-B)	CON4 USB Port (Micro-B)	J1
Model 1	Connect to PC	-	-
Model 2	-	Connect to PC	-
Model 3	-	-	VDD5V Input

- System Reset (SW3): System will be reset if the SW3 button is pressed

- Virtual COM (CON2, U8): NUC123ZD4AN0 microcontroller (U8), USB micro-B connector (CON2) to PC, for debug message output

- User indication LEDs (LED1, LED2):

LED	Color	GPIO pin of NUC980
LED1	Green	PC11
LED2	Green	PC3

- SPI NOR Flash (U5): Use Winbond W25Q256JV 256M Bit (U5) for system booting, supporting normal mode

- JTAG interface (J1/NC)

Connector	GPIO pin of NUC980	Function
J1.1	-	VDD33
J1.2	GPA6	nTRST
J1.3	GPA5	TDI
J1.4	GPA4	TMS
J1.5	GPA3	TCK
J1.6	GPA2	TDO
J1.7	-	nRESET
J1.8	-	VSS

- USB0 Device/HOST (CON3, JP4): USB0 Device/HOST Micro-B connector, By JP4 status or defined by the ID pin of the USB cable

- User Key SW (K1)

Key	GPIO pin of NUC980
K1	GPC15

- Ethernet port interface(CON1)

Connector	GPIO pin of NUN980	Function
CON1.1	-	VDD33
CON1.2	-	VDD33
CON1.3	-	NC
CON1.4	-	NC
CON1.5	GPF9	F_MDC
CON1.6	GPF8	F_MDIO

CON1.7	GPF7	F_TXD0
CON1.8	GPF6	F_TXD1
CON1.9	GPF5	F_TXEN
CON1.10	GPF4	F_REFCLK
CON1.11	GPF3	F_RXD0
CON1.12	GPF2	F_RXD1
CON1.13	GPF1	F_CRSDV
CON1.14	GPF0	F_RXERR
CON1.15	-	NC
CON1.16	-	nRESET
CON1.17	-	NC
CON1.18	-	NC
CON1.19	-	VSS
CON1.20	-	VSS

- Power on setting (SW1, R15, R16)

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

- CAN (JP2, U7): SN65HVD230 transceiver controller of CAN(U7), CAN header(JP2) connect to device for communication

- Peripheral user interface(J2), including I2C, SPI, UART

Connector	GPIO pin of NUN980	Function
J2.1	-	VDD33
J2.2	-	VDD18
J3.3	GPB6	I2C1_SDA
CON1.4	GPB4	I2C1_SCL
CON1.5	GPC3	GPIO
CON1.6	GPC4	SPI0_DO
CON1.7	GPC5	SPI0_SS0

CON1.8	GPC6	SPI0_CLK
CON1.9	GPC8	SPI0_DI
CON1.10	GPC9	UART4_TXD
CON1.11	GPC10	UART4_RXD
CON1.12	-	VSS

- SOC CPU: NUC980DR63YC (U4)

Rear View

Figure 3-2 shows the main components from the rear view of NUMAKER-RTU-NUC980(Chili) Development Board

- VCOM ICE interface: ICE Controller NUC123ZD4AN0 (U6), USB connector (CON3) to PC Host

3.2

Connector	Pin Name	Functions
CON3.1	VDD33	DC 3.3V
CON3.2	ICE_DAT	Serial Wired Debugger Data
CON3.3	ICE_CLK	Serial Wired Debugger Clock
CON3.4	RST#	VCOM Chip Reset, Active Low.
CON3.5	VSS	Power Ground

- RS485 (JP1, U6): SN65HVD11DR transceiver controller of RS485(U6), RS485 header(JP1) connect to device for communication

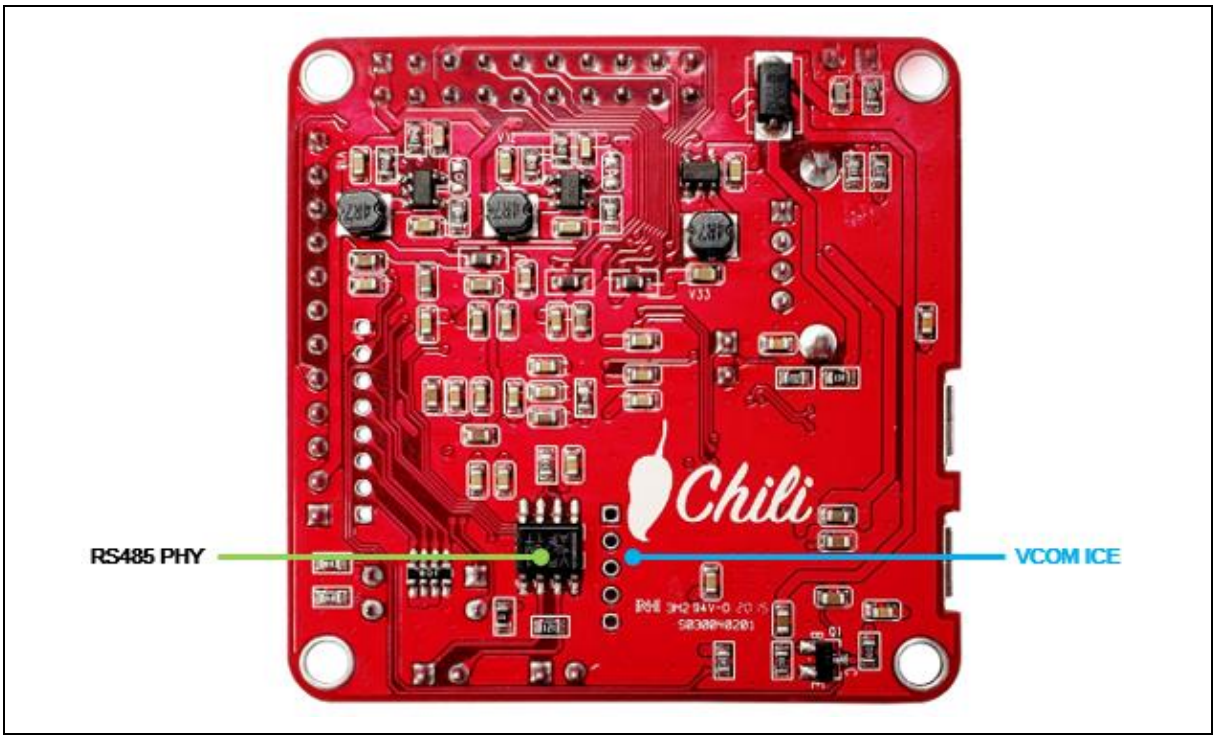


Figure 3-2 Rear View of NUMAKER-RTU-NUC980(Chili)

4 QUICK START

4.1 Nuvoton CDC 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.

Download and install the latest Nuvoton CDC driver:

- https://www.nuvoton.com/resource-download.jsp?tp_GUID=SW1020160914071736

The installation is presented Figure 4-1 and Figure 4-2.

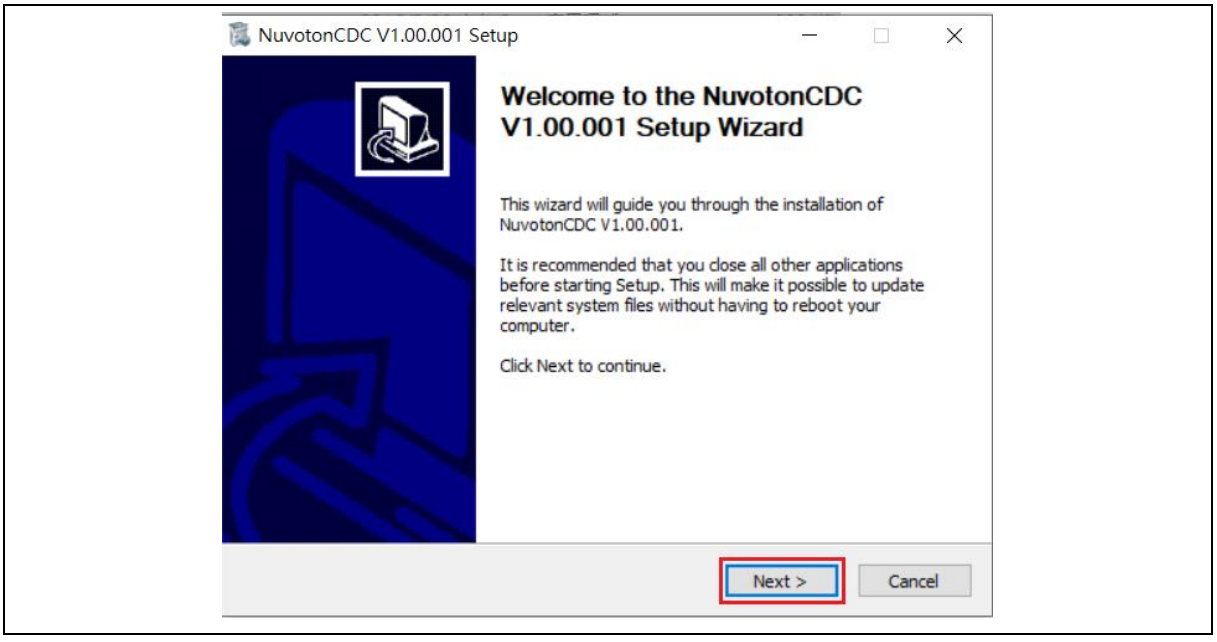
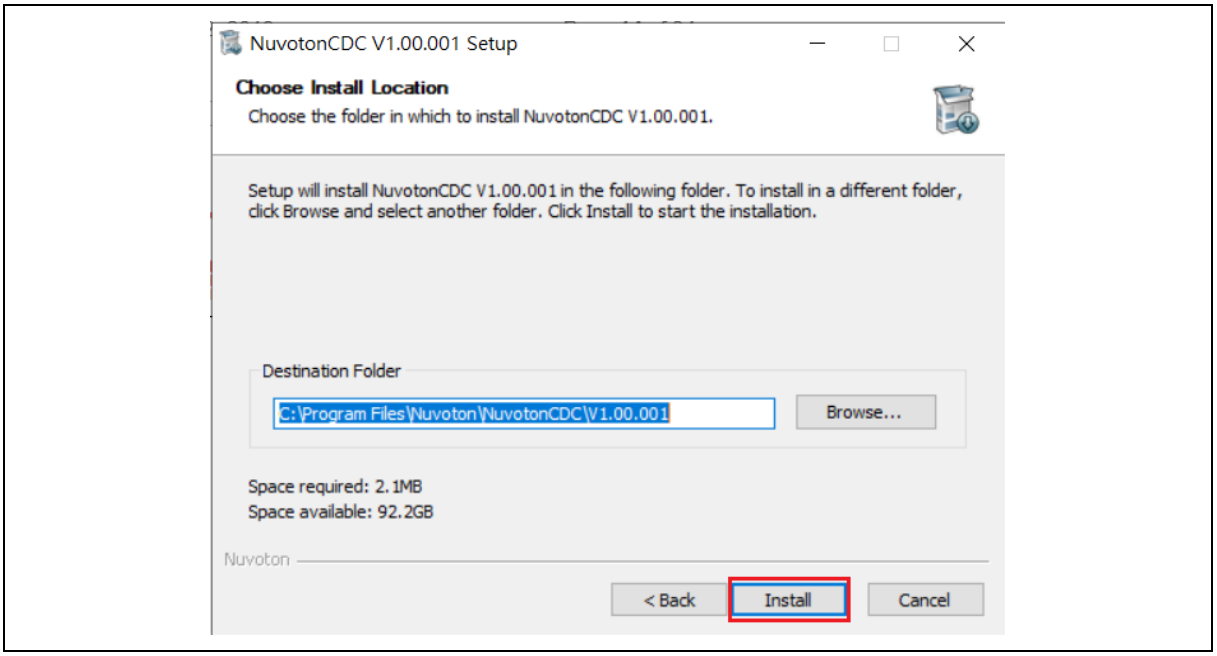


Figure 4-1 Nuvoton USB Driver Installation Setup



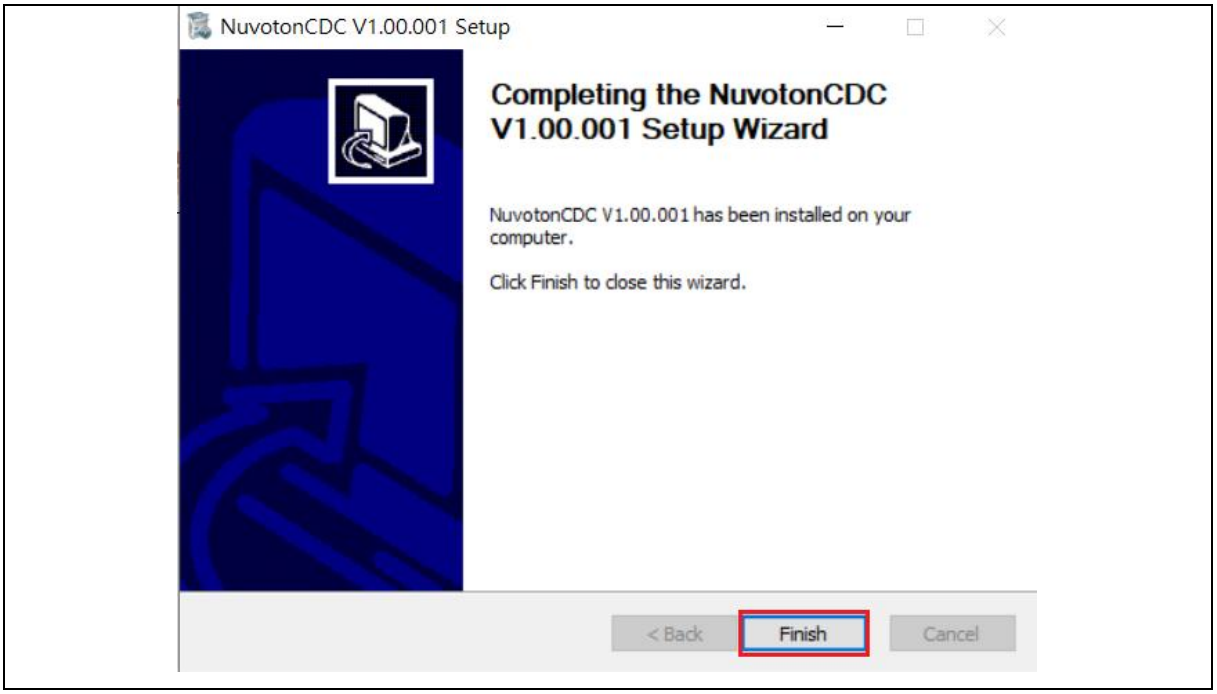


Figure 4-2 CDC Driver Installation

4.2 Nuvoton Virtual COM driver Installation

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

Download and install the latest Nuvoton Virtual COM driver:

https://github.com/OpenNuvoton/NUC980_NuWriter/tree/master/Driver

The installation is presented in Figure 4-3 and Figure 4-4.

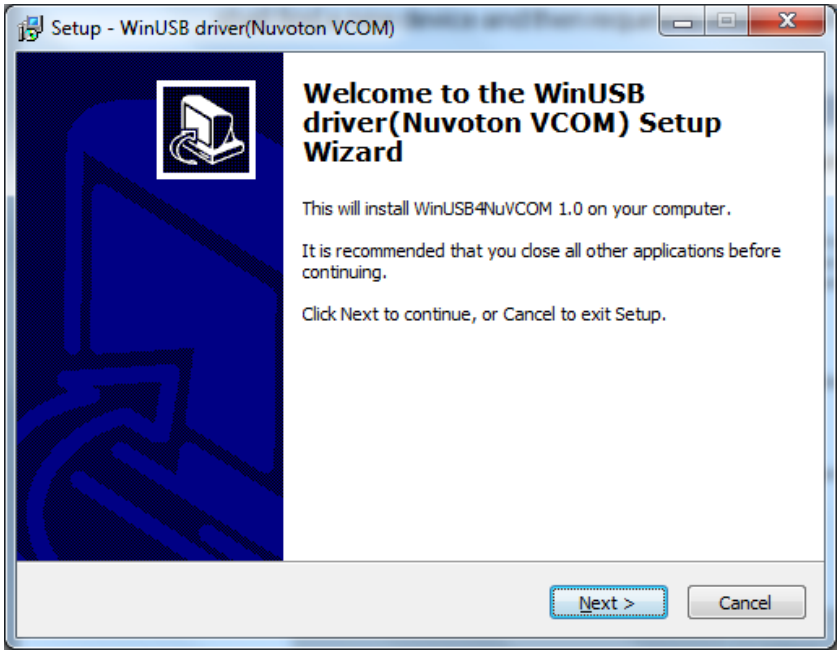
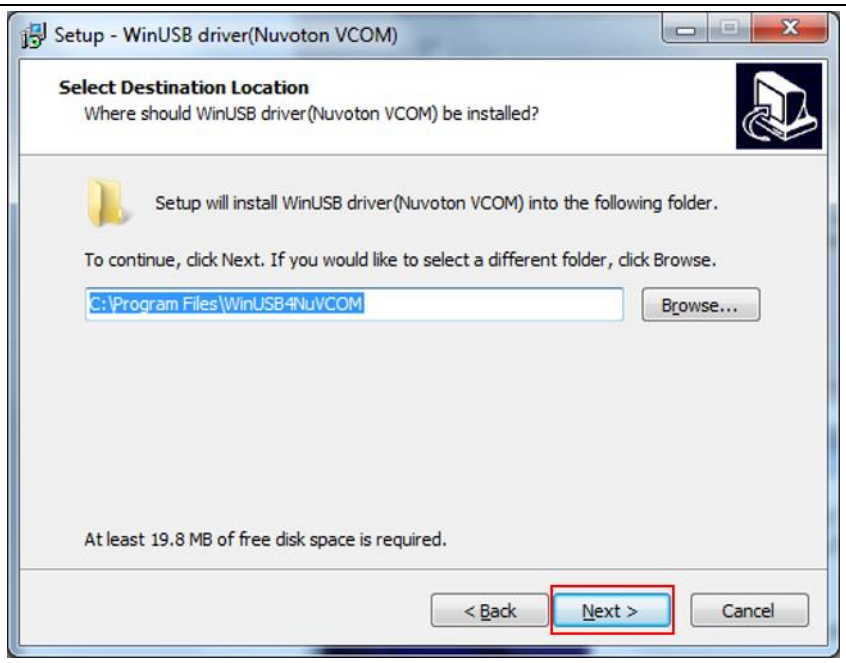
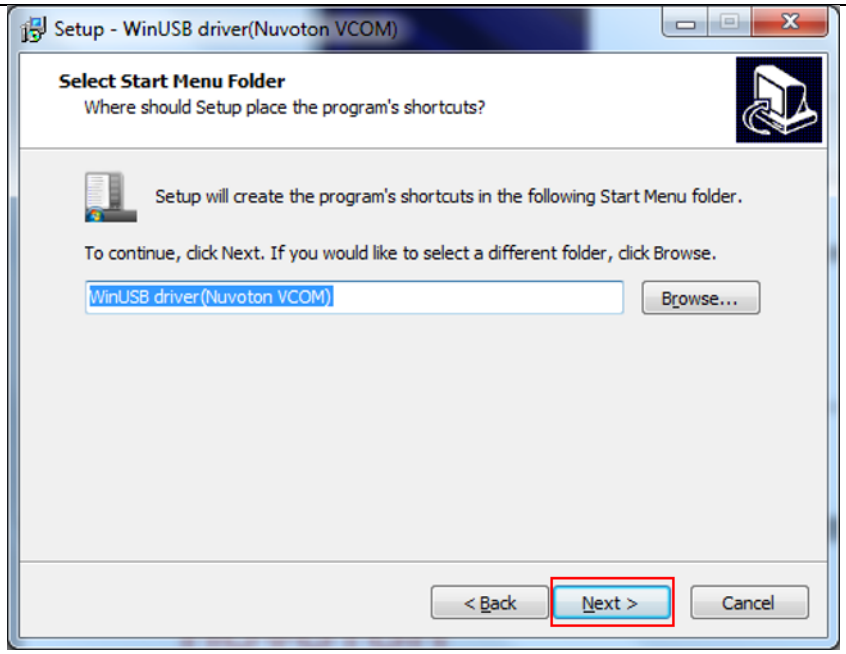
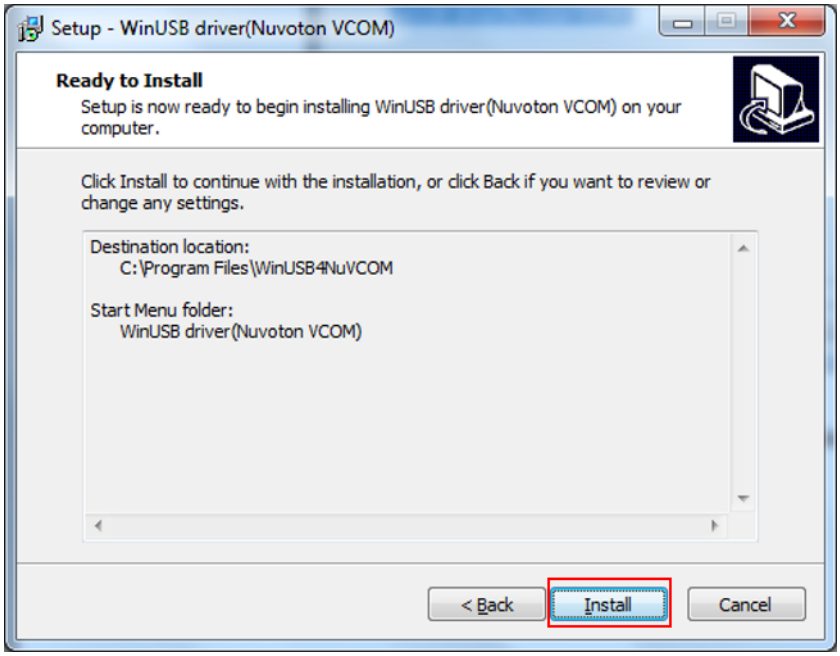


Figure 4-3 VCOM Driver Installation Setup

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




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


Setup - WinUSB driver(Nuvoton VCOM)
Ready to Install
Setup is now ready to begin installing WinUSB driver(Nuvoton VCOM) on your computer.
Click Install to continue with the installation, or click Back if you want to review or change any settings.
Destination location:
C:\Program Files\WinUSB4NuVCOM
Start Menu folder:
WinUSB driver(Nuvoton VCOM)
< Back
Install
Cancel

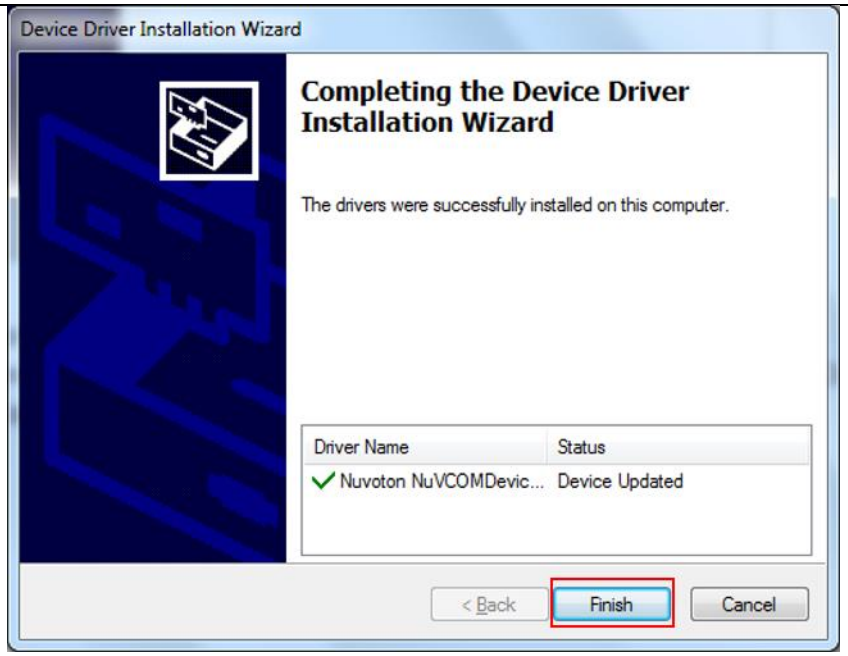


Figure 4-4 VCOM Driver Installation Setup

4.3 BSP Firmware Download

NUC980 Linux BSP provides cross compilation tools based on Linux. We have tested this BSP in different x86 Linux distributions, including Ubuntu, CentOS, and Debian…etc. Because there are so many distributions out there with different system configuration, sometimes it is necessary to change system setting or manually install some missing component in order to cross compile.

Linux development environment could either be native, or install in a virtual machine execute on top of other operating system.

BSP download locations:

Official website:

https://www.nuvoton.com/products/iot-solution/iot-platform/numaker-rtu-nuc980/

- VMware Linux Virtual machine image
  - An UBUNTU18.04 VMware Image with NUC980 toolchain and Buildroot
- VMware Linux Virtual machine image User Manual
  - Introduction of NUC980 Buildroot usage and how to compile firmware for NUC980
- Hardware
  - Schematics and Gerber files
- NUC980 Linux V4.4 BSP
  - Linux v4.4 BSP and relative tool documents
- NUC980 Linux V5.10 BSP
  - Buildroot for NUC980 Linux v.510

Github:

https://github.com/OpenNuvoton/MPU-Family

For more details about NUC980 Linux BSP, please refer to “NUC980 Linux 4.4 BSP User Manual EN” in the “BSP/Documents” directory.

Hardware Setup

The NuMaker-IIoT-NUC980 provides jumpers to select boot-up conditions. To select USB ISP mode, the statuses of SW1.1 and SW1.2 are ON. Other boot selects can refer to the following figure and table

4.4

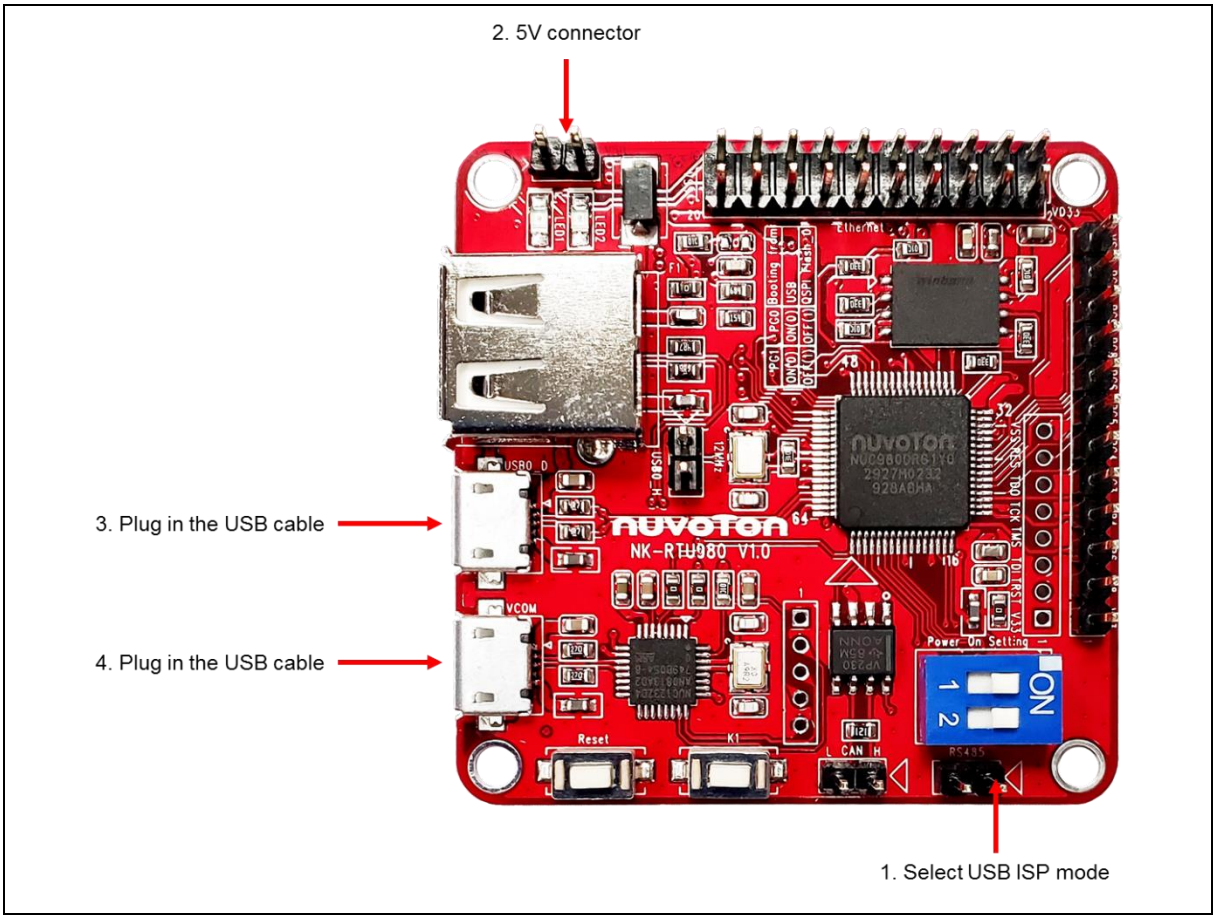


Figure 4-5 Hardware Setting

1. NUMAKER-RTU-NUC980(Chili) provides jumpers (SW1) to select boot-up conditions. The jumpers (SW1) ON to select USB ISP mode.

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

Table 4-1 Power On Setting

2. 5V input connector
3. Plug in the USB cable

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

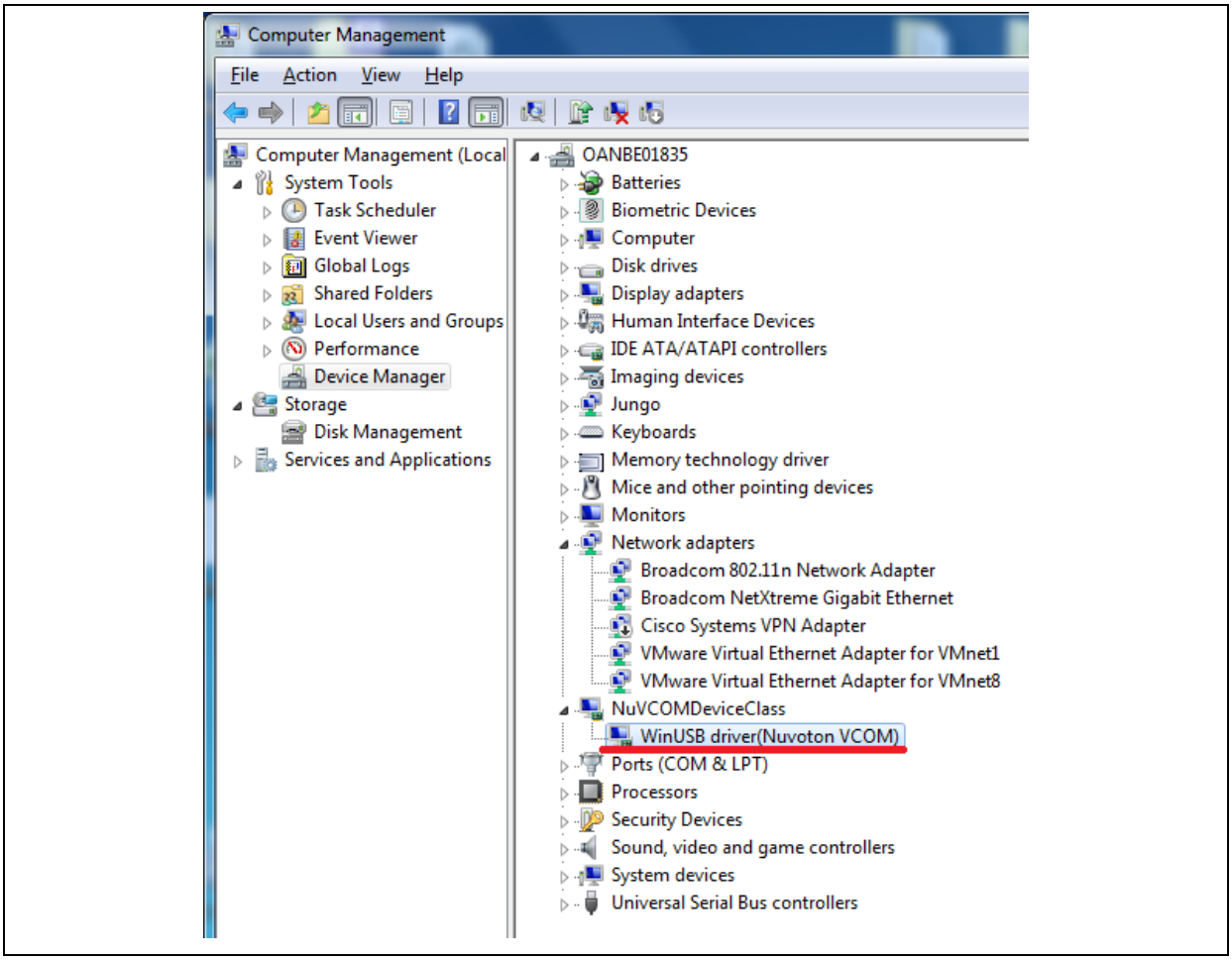


Figure 4-6 Device Manager(1)

4. Plug in the USB cable

   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.

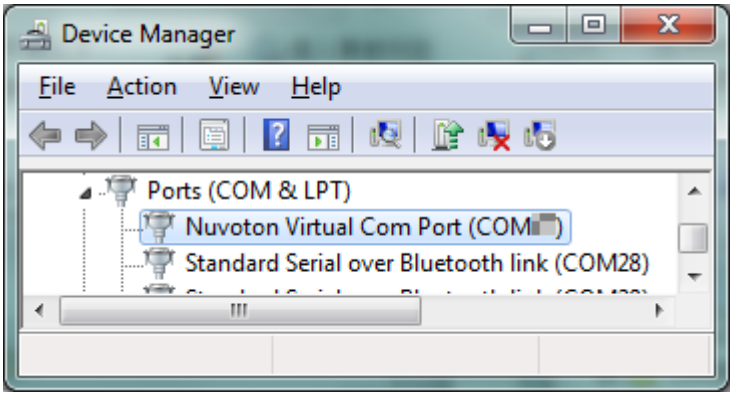


Figure 4-7 Device Manager(2)

NuWriter Tool

4.5.1 NuWriter Setup

4.5

1. Refer to chap.4.3 to install NuWriter tool
2. Connect USBD connector shown Figure 4-5 in to the PC USB port through a USB cable
3. Booting NUMAKER-RTU-NUC980(Chili) from USB ISP mode
4. Double click "**NuWriter.exe**" on PC. Select target chip as "NUC980 series" and select DDR parameter is "NUC980DR6xYC.ini". And then, press "**Continue**" button.

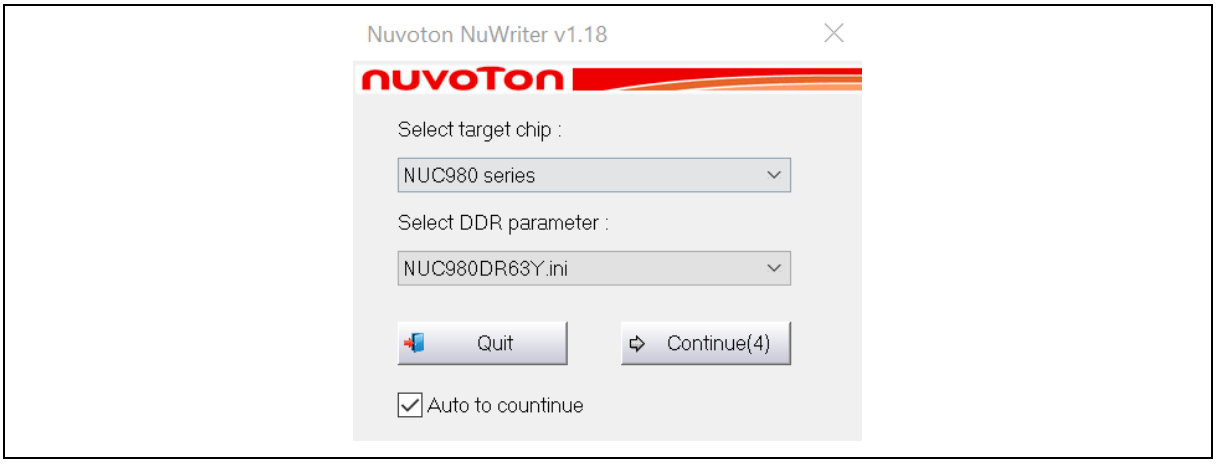


Figure 4-8 NuWriter Setting

NuWriter provides 7 types to be downloaded images including DDR/SRAM, SPI, NAND, eMMC/SD, SPI NAND, PACK and Mass Production. This chapter will guide you to download images to SPI NAND flash. If you want to choose others types to download images. For more details about NUC980 Linux BSP, please refer to **NUC980 NuWriter User Manual**" in the "BSP/Documents" directory.

4.5.2 SPI Mode

This mode can write a new image to SPI NOR flash and specify the type of the image. These types can be recognized by uboot or Linux. The Image type is set Loader, Data, Environment or Pack.

NuMaker-RTU-NUC980(Chili) default firmware consist of four images:

1. u-boot
2. uImage
3. dtb (Only required for Linux v5.10)
4. environment variables

Please refer to **VMware Linux Virtual machine image User Manual** to generate above firmware images.

The following the steps below to program u-boot.bin:

a. Select the "**SPI** " type.
b. Fill in the image information：
   - Image Name: u-boot.bin
   - Image Type: Loader
   - Image execute address: 0xe00000

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.

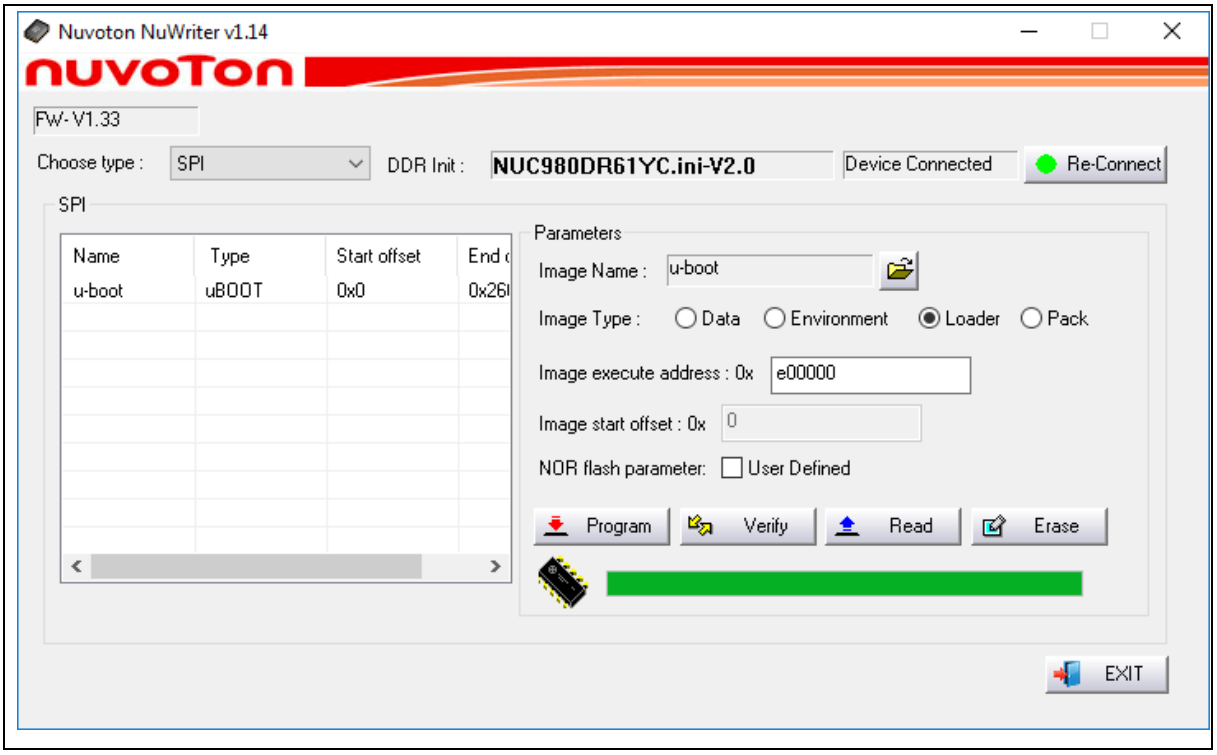


Figure 4-9 Program u-boot

dtb (Only required for Linux v5.10)

Linux 5.10 need to download the dtb into SPI NAND Flash at the specified address, depending on the value of image start offset (aligned on block size boundary, block size is based on SPI Flash specifications). If dtb start offset is equal to 0x1C0000, download the dtb into SPI NAND Flash at the address 0x1C0000.

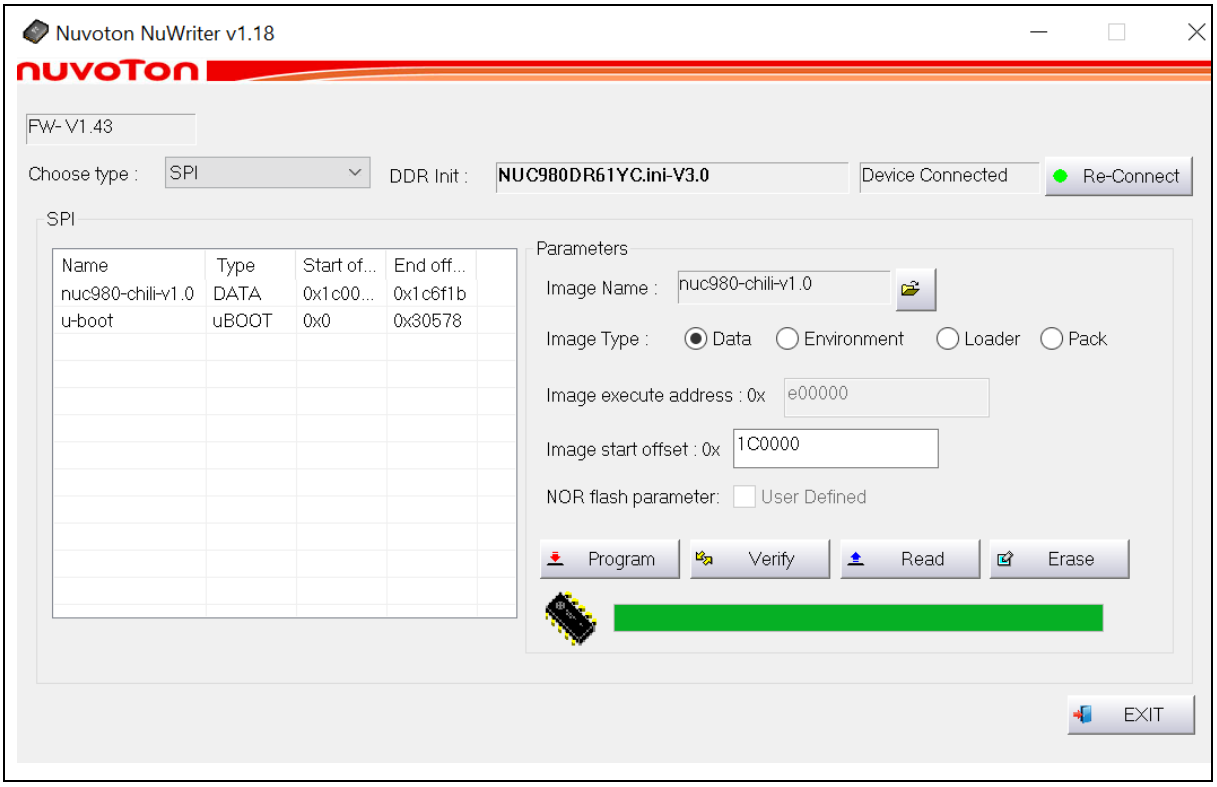


Figure 4-10 Program dtb

The following the steps to program kernel image:

a. Select the "**SPI**" type.
b. Fill in the image information：
   - Image Name: uimage
   - Image Type: Data
   - Image execute address: 0x200000
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.

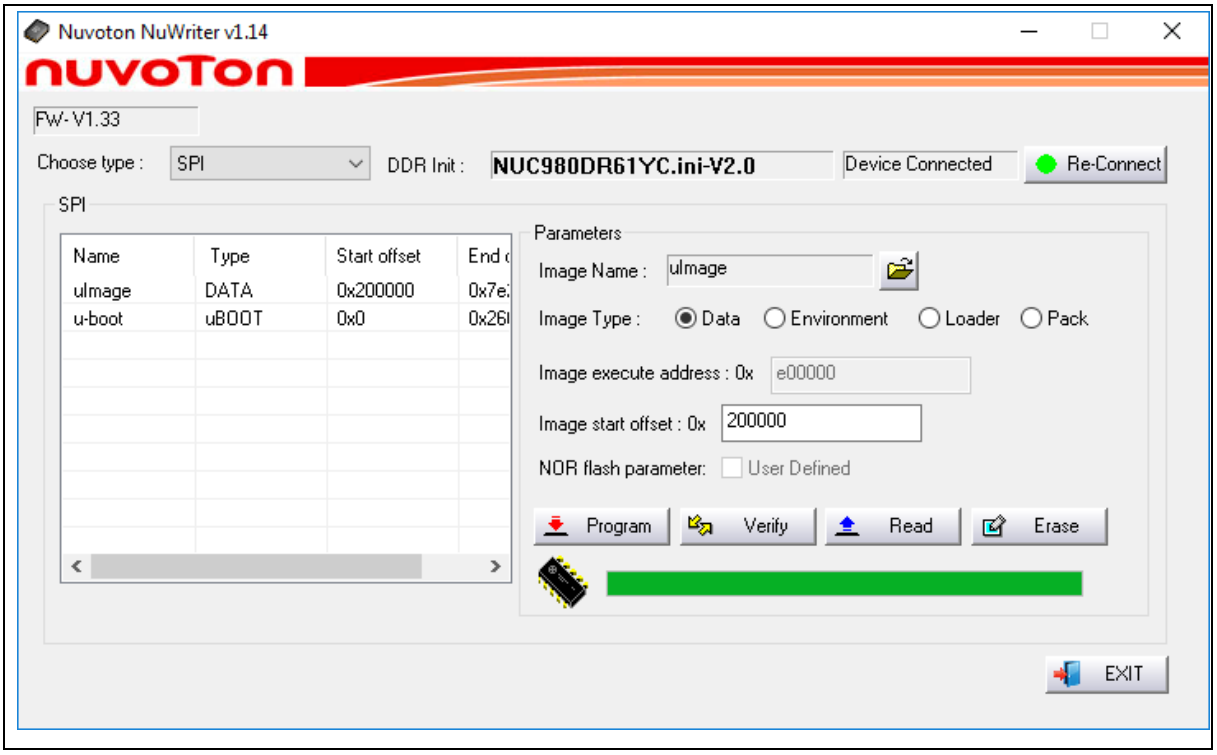


Figure 4-11 Program uImage

The following the steps below to program environment:

a. Select the "**SPI**" type.
b. Fill in the image information：
   - Image Name: env.txt
   - Image Type: environment
   - Image start offset address: 0x80000
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.

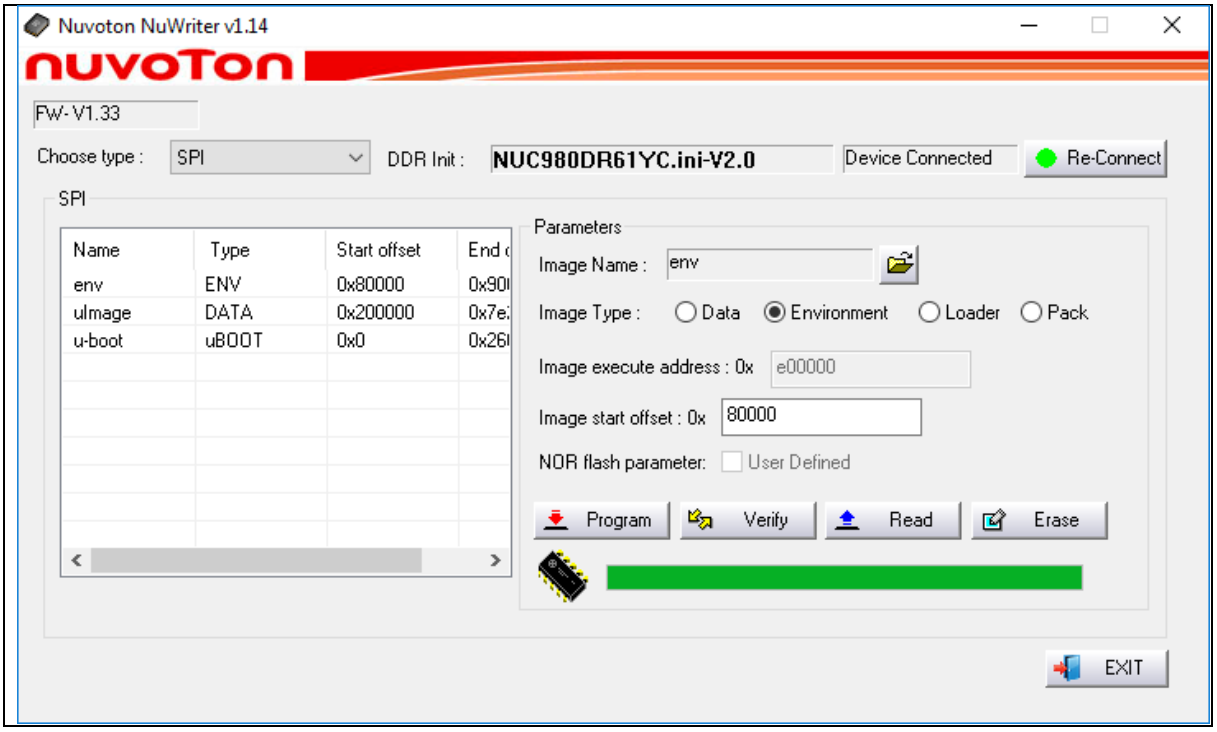


Figure 4-12 Program environment

You could create a TXT file extension and add contents. NuWriter will transform env.txt to an environment image and download the image to SPI NOR Flash.

For Linux v4.4

Here is an example for NuMaker-RTU-NUC980 with Linux v4.4 environment variables:

```
baudrate=115200
bootdelay=1
stderr=serial
stdin=serial
stdout=serial
setspi=sf probe 0 30000000
loadkernel=sf read 0x7fc0 0x200000 0x800000
bootcmd=run setspi;run loadkernel;bootm 0x7fc0
```

For Linux v5.10

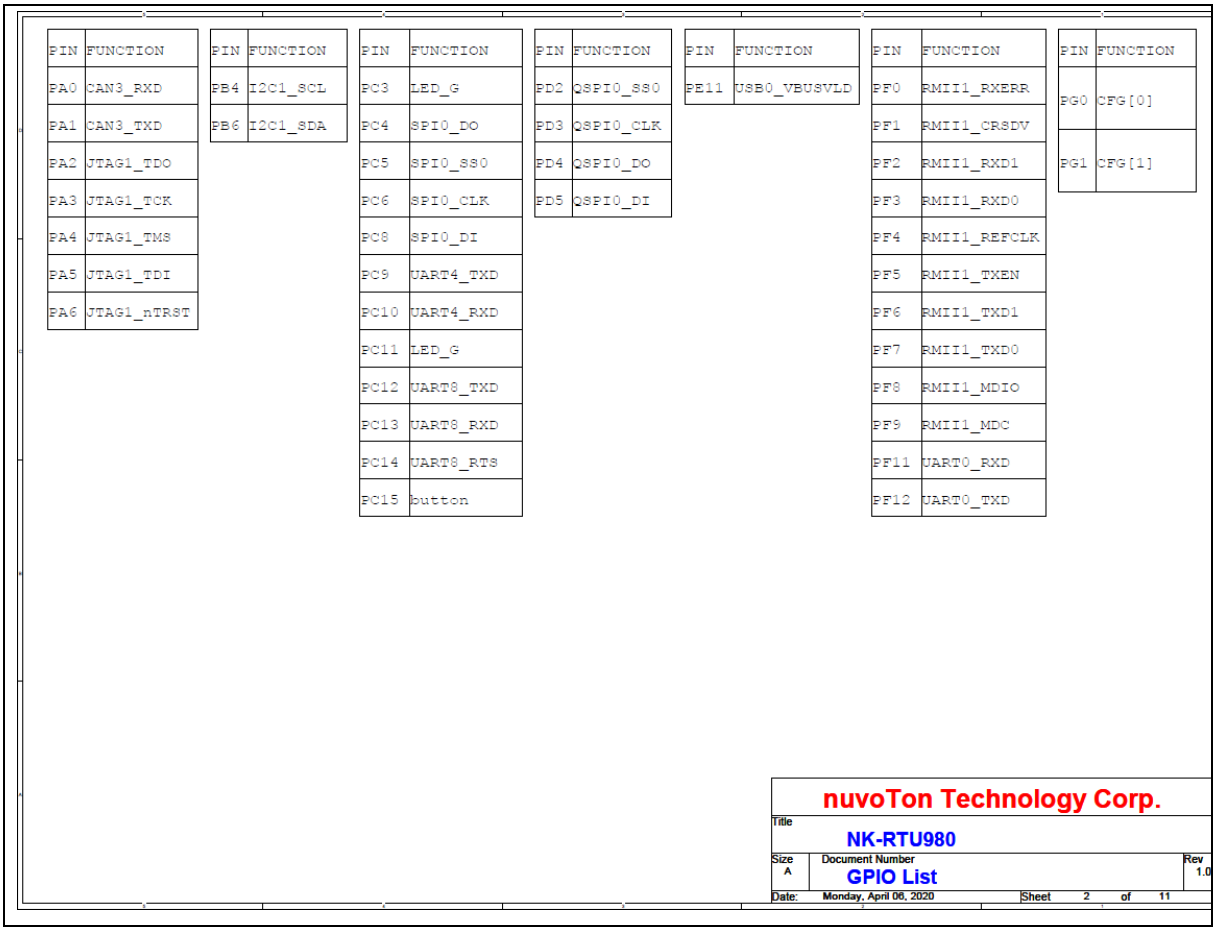
Here is an example for NuMaker-RTU-NUC980 with Linux v5.10 environment variables:

```
baudrate=115200
bootdelay=1
stderr=serial
```

```
stdin=serial
stdout=serial
setspi=sf probe 0 30000000
loadkernel=sf read 0x07fc0 0x200000 0xA00000
loaddtb=sf read 0x1400000 0x1C0000 0x20000
bootcmd=run setspi;run loadkernel;run loaddtb;bootm 0x07fc0 - 0x1400000
```

5 SCHEMATIC

5.1 GPIO List Schematic



PIN	FUNCTION
PA0	CAN3_RXD
PA1	CAN3_TXD
PA2	JTAG1_TDO
PA3	JTAG1_TCK
PA4	JTAG1_TMS
PA5	JTAG1_TDI
PA6	JTAG1_nTRST

PIN	FUNCTION
PB4	I2C1_SCL
PB6	I2C1_SDA

PIN	FUNCTION
PC3	LED_G
PC4	SPI0_DO
PC5	SPI0_SS0
PC6	SPI0_CLK
PC8	SPI0_DI
PC9	UART4_TXD
PC10	UART4_RXD
PC11	LED_G
PC12	UART8_TXD
PC13	UART8_RXD
PC14	UART8_RTS
PC15	button

PIN	FUNCTION
PD2	QSPI0_SS0
PD3	QSPI0_CLK
PD4	QSPI0_DO
PD5	QSPI0_DI

PIN	FUNCTION
PE11	USB0_VBUSVLD

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
PF11	UART0_RXD
PF12	UART0_TXD

PIN	FUNCTION
PG0	CFG[0]
PG1	CFG[1]

Figure 5-1 GPIO List Schematic

5.2 Power Schematic

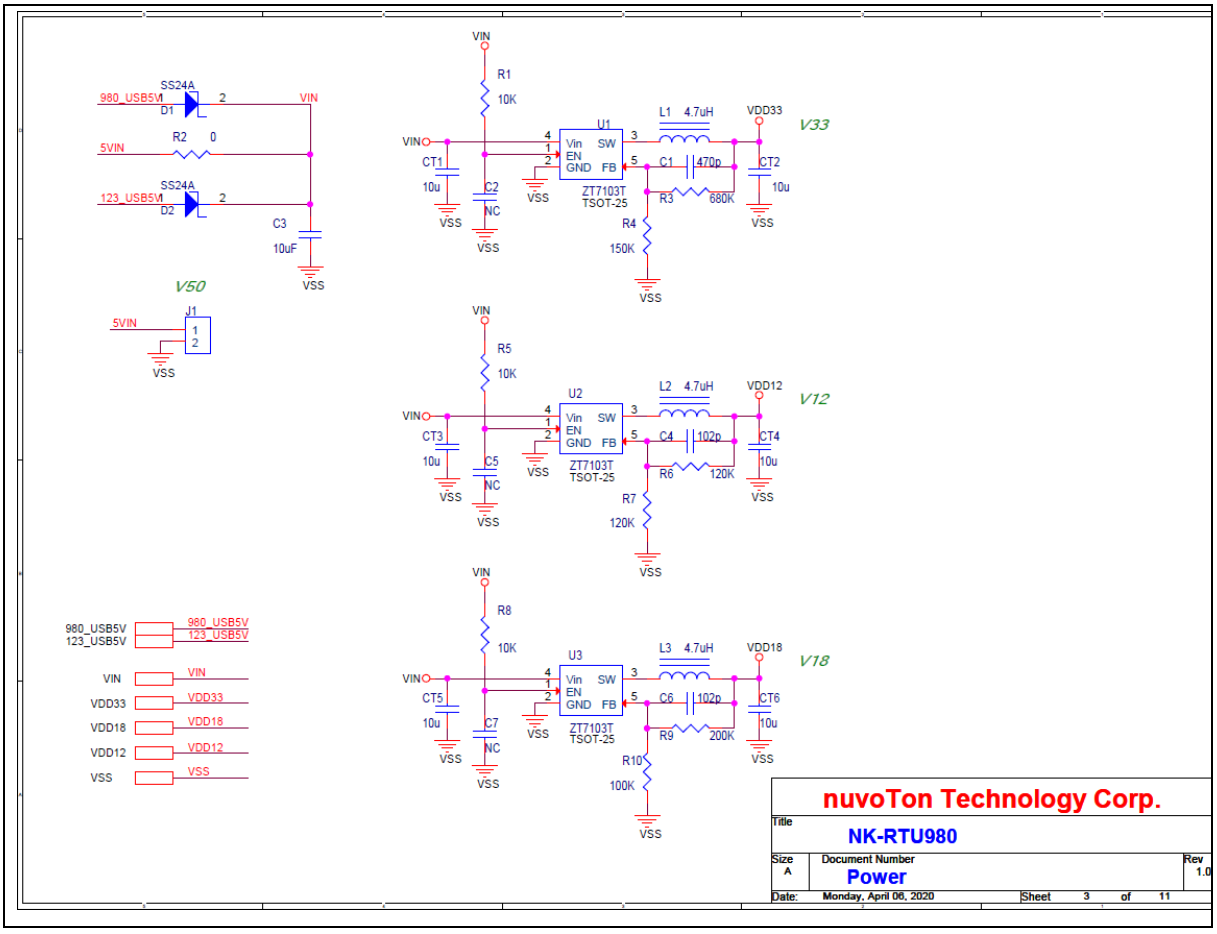


Figure 5-2 Power Schematic

NUC980DR Schematic

5.3

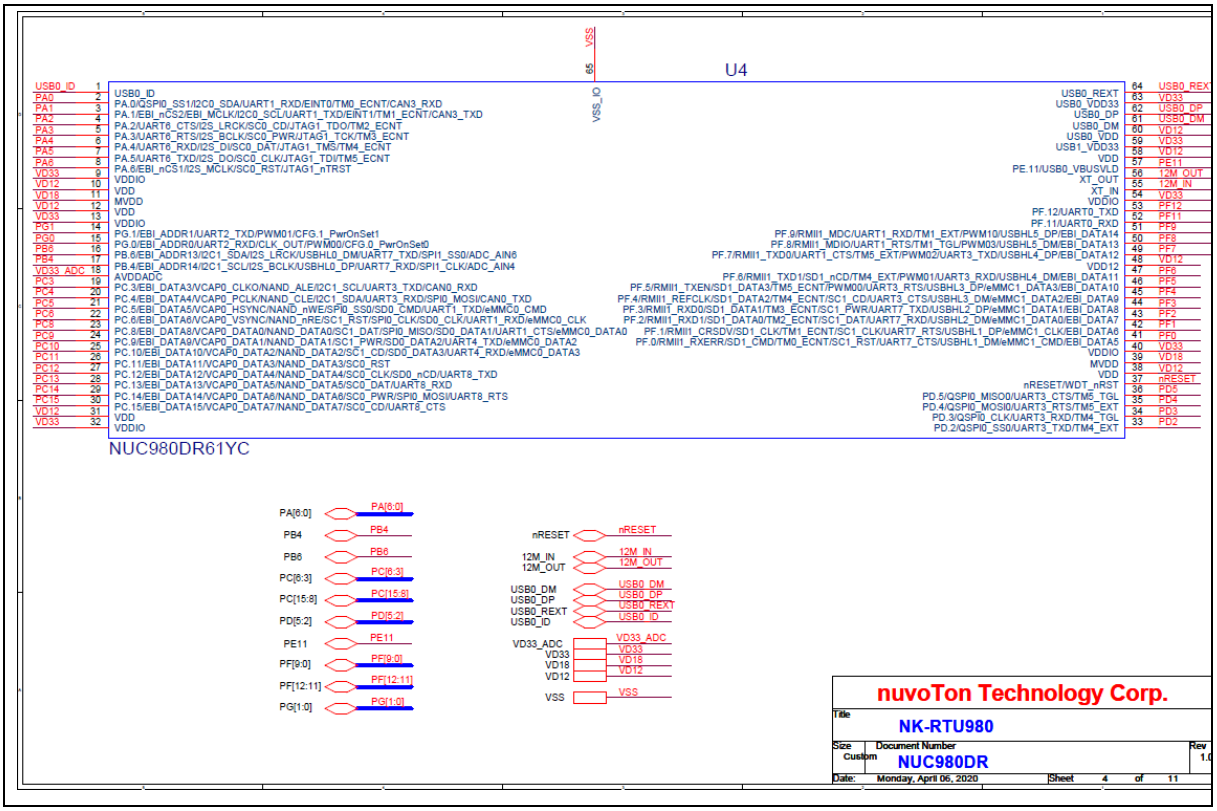


Figure 5-3 NUC980DR Schematic

5.4 Power Filter Schematic

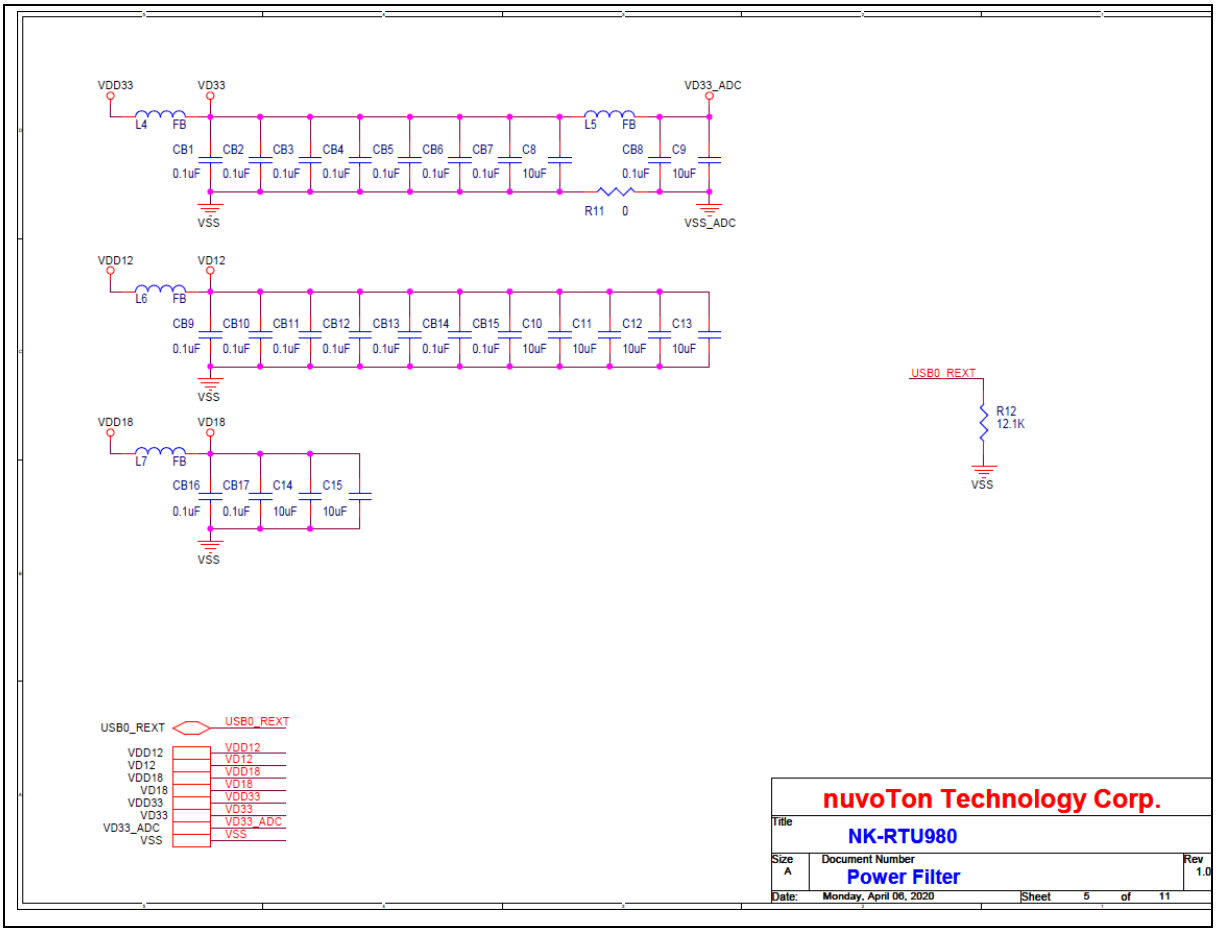


Figure 5-4 Power Filter Schematic

5.5 Configure Schematic

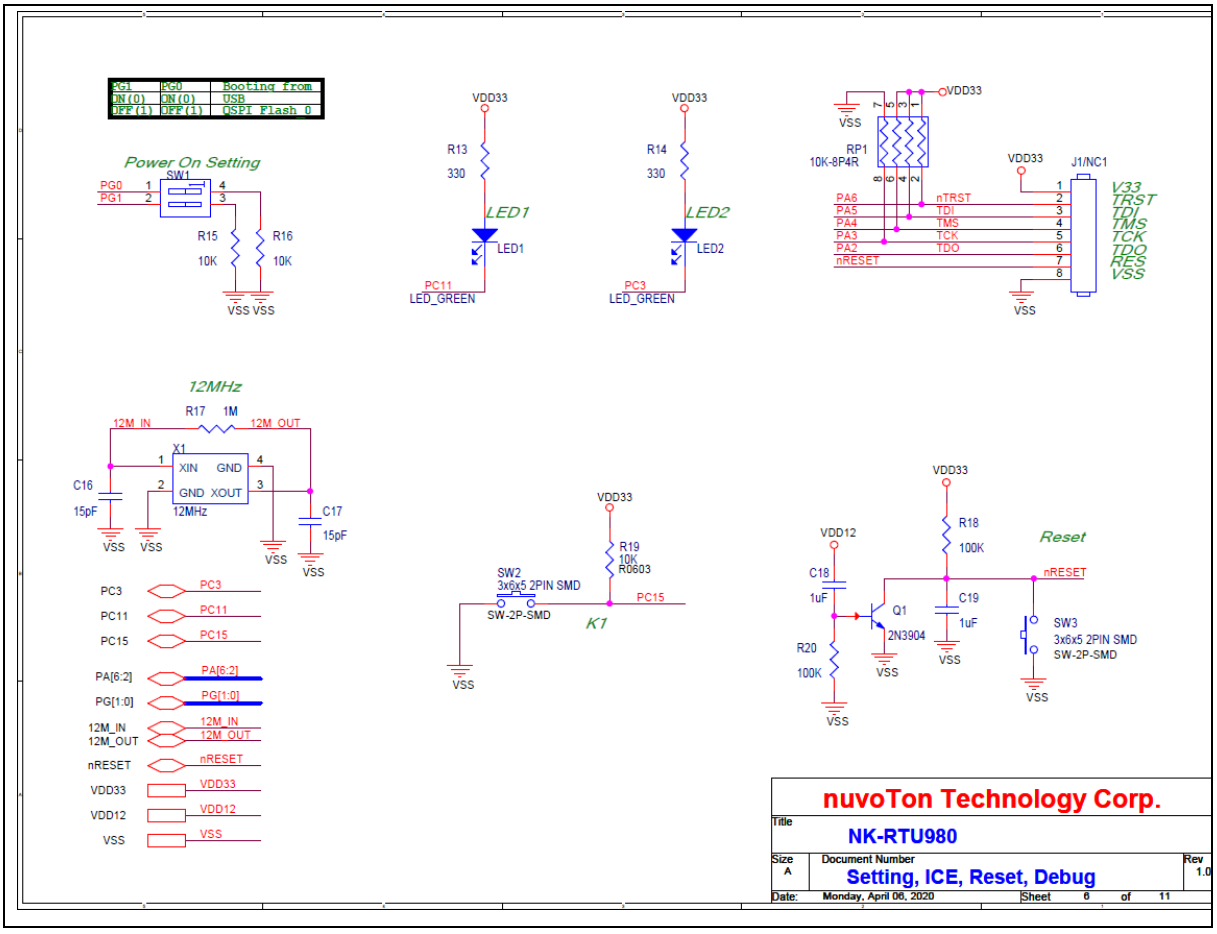


Figure 5-5 Configure Schematic

5.6 NUC123ZD4AN0 Schematic

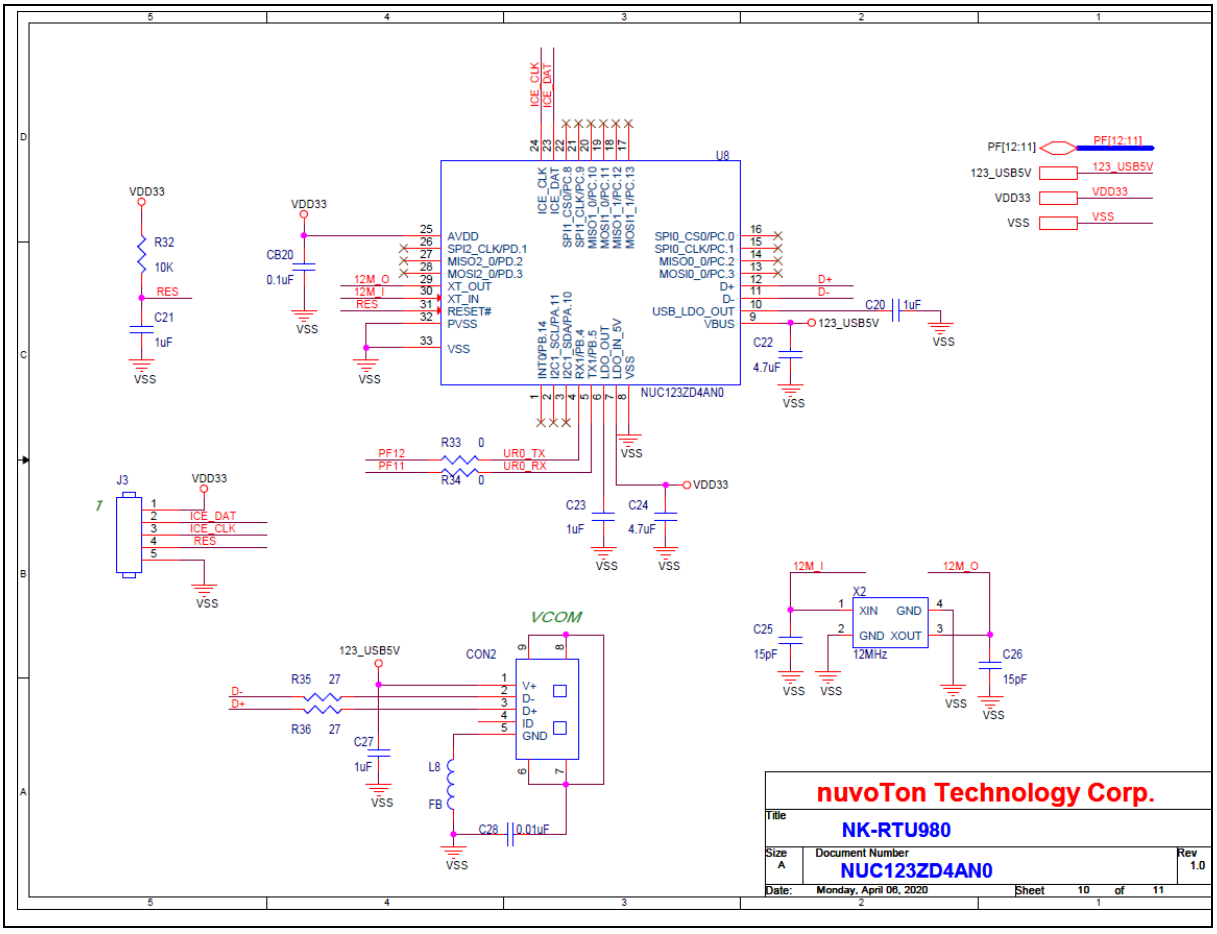


Figure 5-6 NUC123ZD4AN0 Schematic

5.7 Memory Schematic

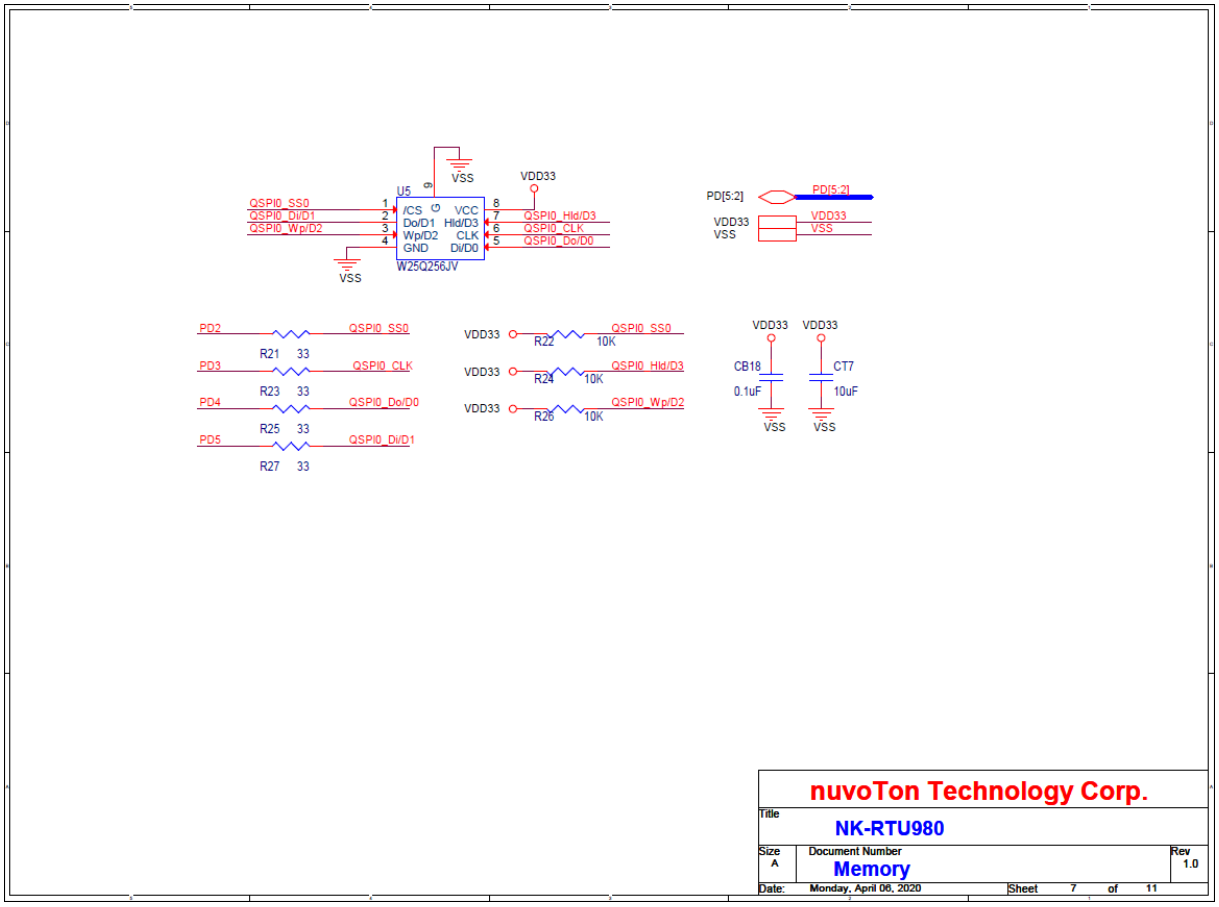


Figure 5-7 Memory Schematic

5.8 RMII_PF connector Schematic

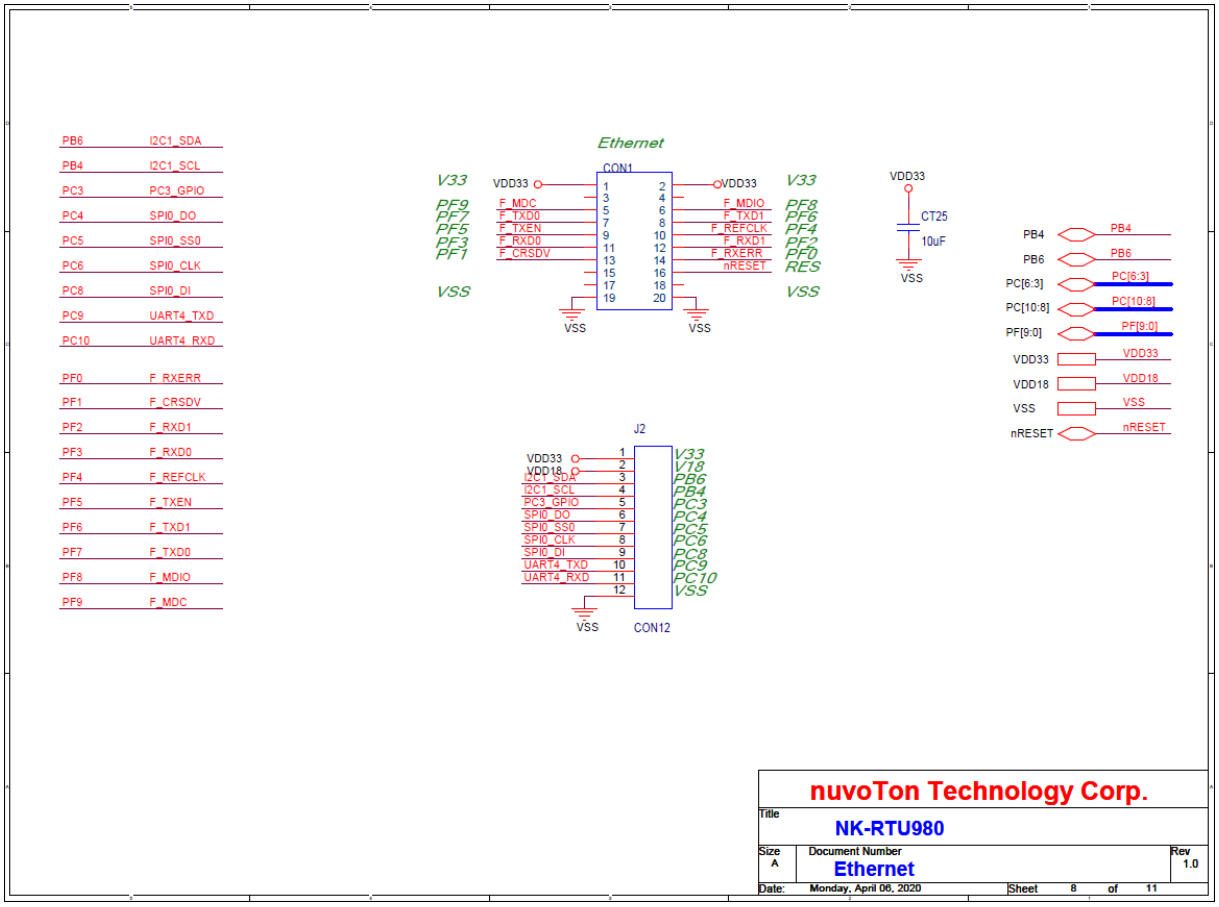


Figure 5-8 RMII_PF connector Schematic

5.9 RS485 and CAN Schematic

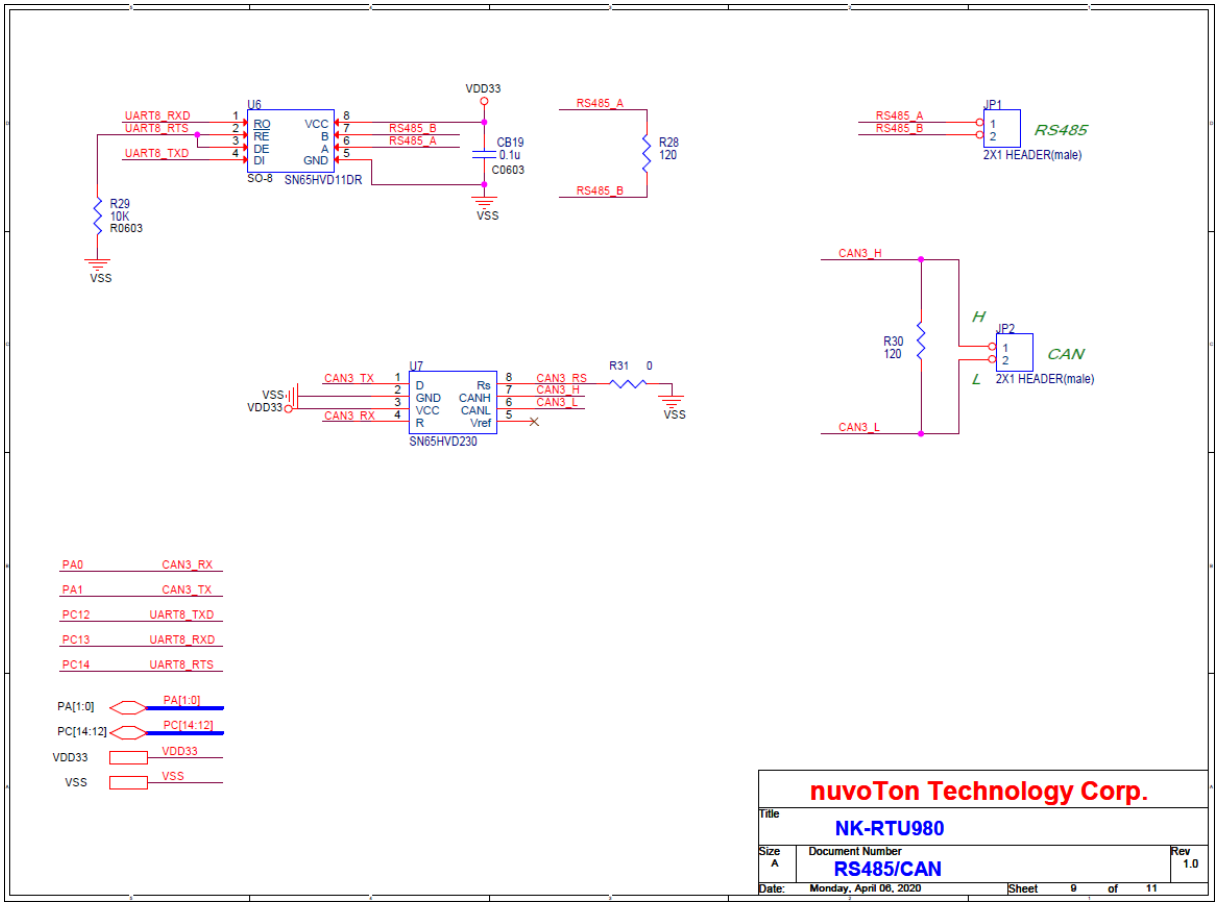


Figure 5-9 RS485 and CAN Schematic

5.10 USB Schematic

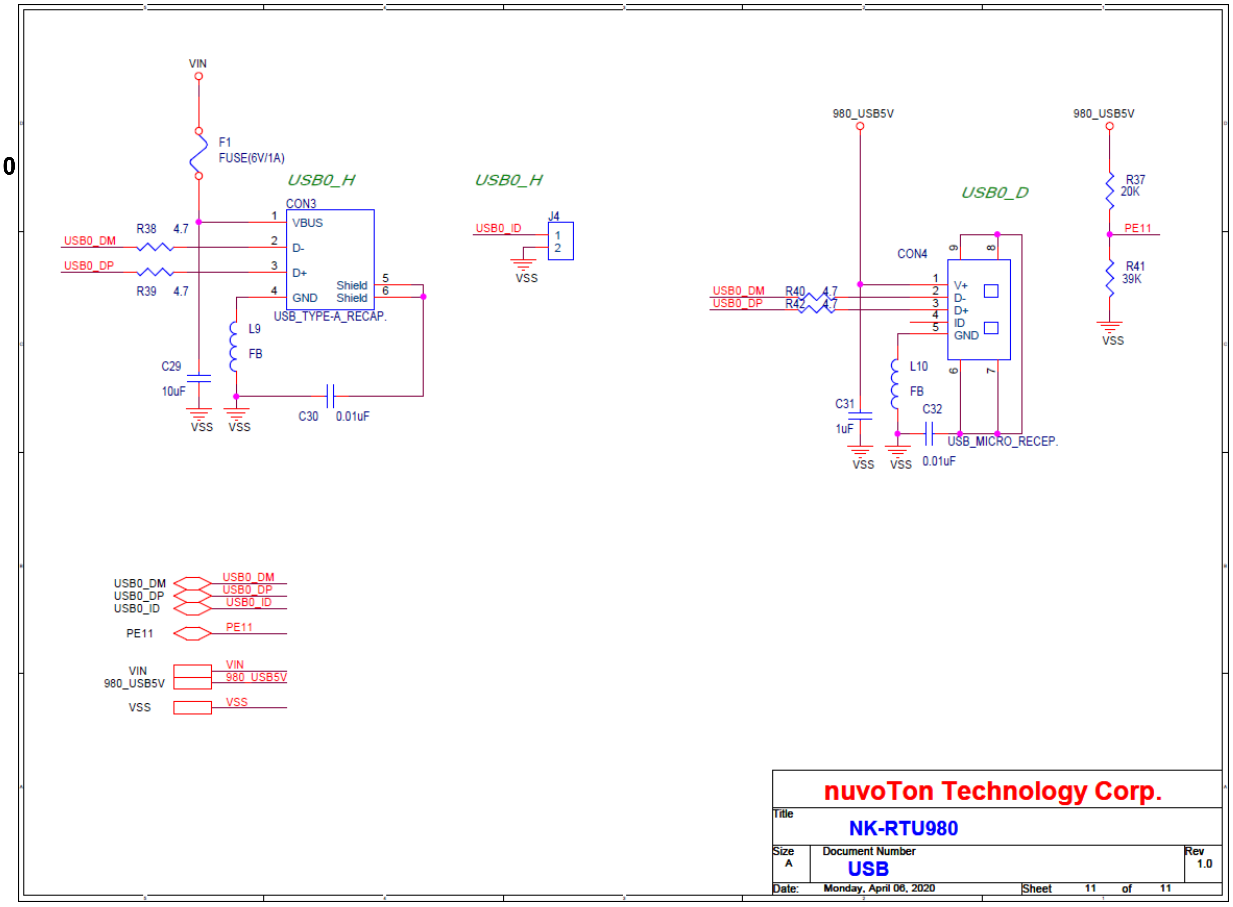


Figure 5-10 USB Schematic

5.11 PCB Placement

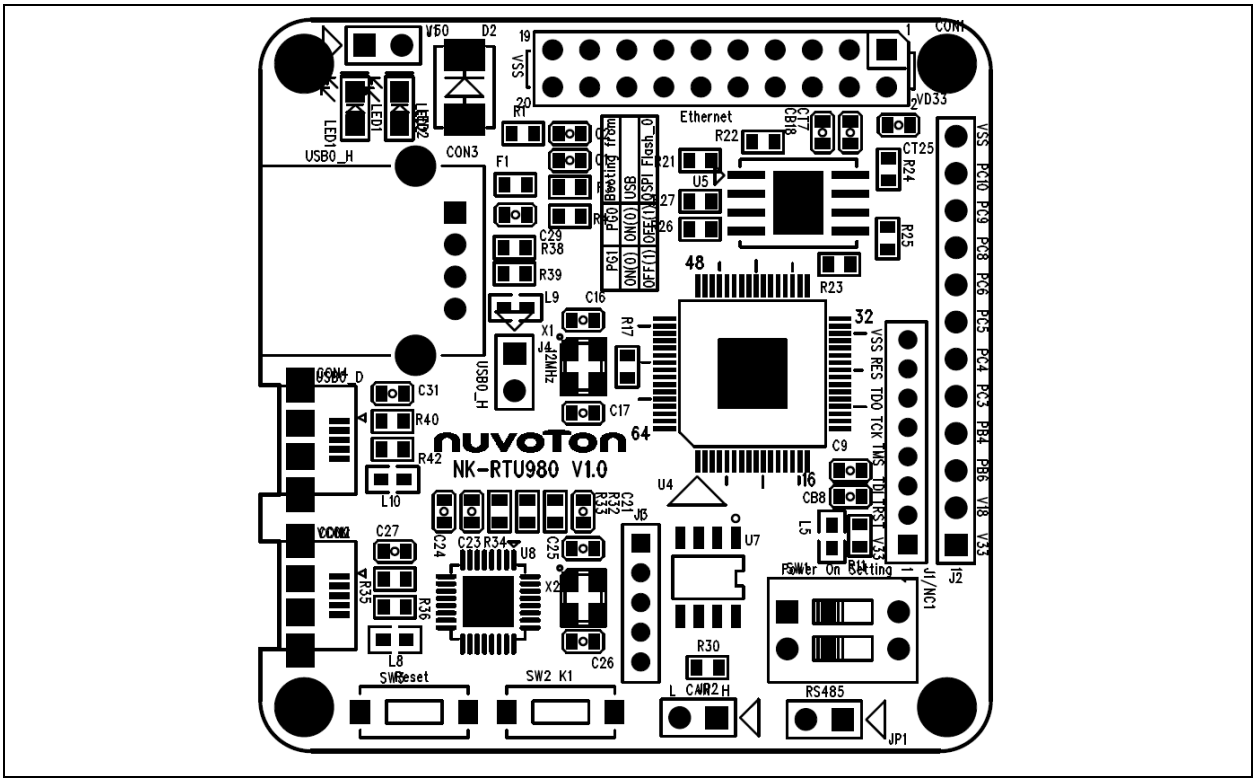


Figure 5-11 Front PCB Placement

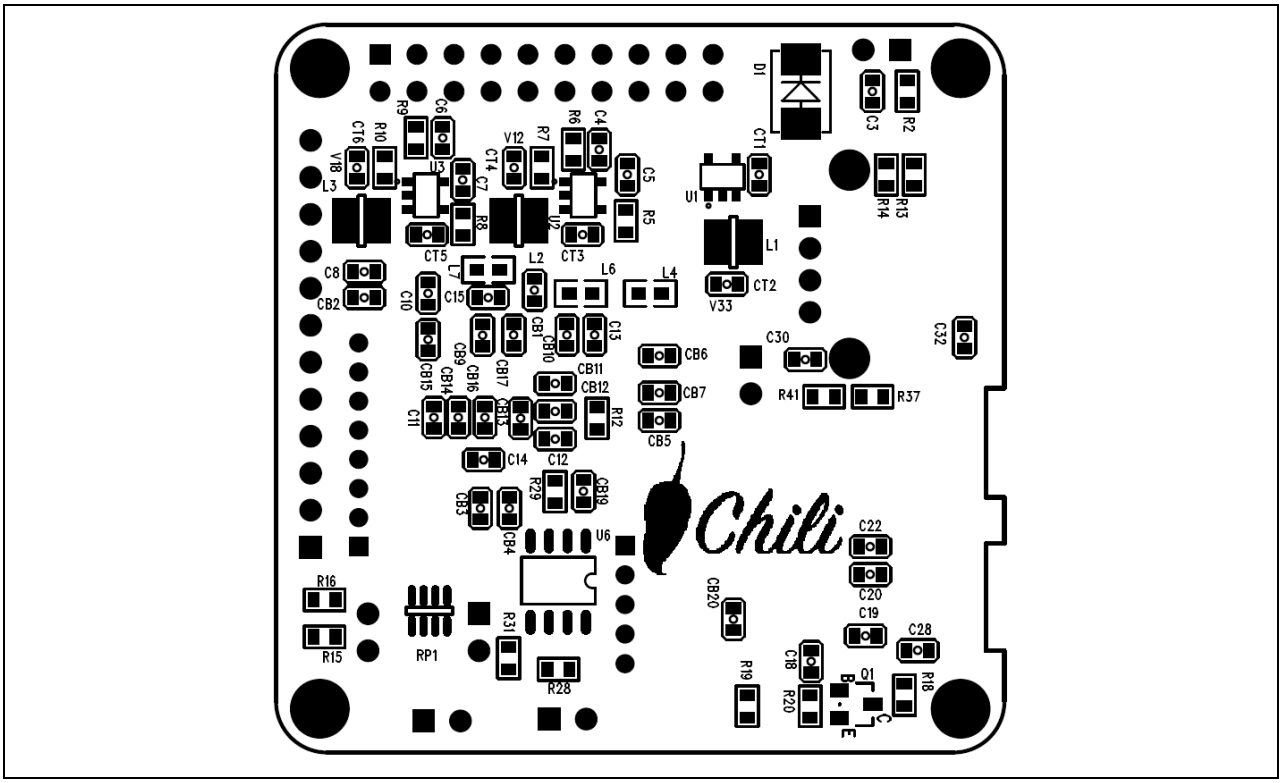


Figure 5-12 Back PCB Placement

6 REVISION HISTORY

Date	Revision	Description
2020.05.22	1.00	Initial version
2023.07.18	1.10	Resources path and Chip ID updated.
2023.11.29	1.20	Add descriptions for Linux v5.10

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.