

**ARM<sup>®</sup> 926EJ-S**  
**32-bit Microprocessor**

# NuMaker Tomato 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](http://www.nuvoton.com)

**Table of Contents**

1 Overview ..... 3

2 Hardware Features ..... 4

    2.1 NuMaker Tomato Board Features ..... 4

    2.2 NUC976DK62Y Features..... 5

3 Board Extension Interfaces ..... 7

4 Getting Started with NuMaker Tomato..... 10

    4.1 Download Software Package and USB CDC Driver ..... 10

    4.2 Install Nuvoton USB CDC Driver ..... 10

    4.3 Copy Software Package to MicroSD Card..... 11

    4.4 Connect and Set Up Hardware ..... 11

    4.5 Open USB CDC Serial COM Port..... 12

5 Example of Executing Sample Code..... 14

6 Appendix..... 15

    6.1 Schematics ..... 15

    6.2 Python API ..... 22

        6.2.1 I<sup>2</sup>C (pysmbus) ..... 22

        6.2.2 SPI (spidev)..... 22

        6.2.3 GPIO (gpio) ..... 22

        6.2.4 PWM (pwm)..... 23

        6.2.5 ADC (adc)..... 23

        6.2.6 UART(pyserial) ..... 23

7 REVISION HISTORY ..... 24

# 1 OVERVIEW

NuMaker Tomato is an open development board. The NuMaker Tomato development platform enables you to design prototypes for Internet of Things (IoT) devices, Gateway / Router or audio stream application and Ethernet industrial control, using hardware and an API that are similar to those offered for Arduino boards.

The platform is based on Nuvoton 300MHz ARM9 SoC which is highly integrated System-on-Chip (SOC) design, The SoC provides features such as 10/100Mbps Ethernet, high-speed USB 2.0 host/device ports, and built-in 64M Byte DDRII memory .

The NuMaker Tomato uses the NUC976DK62Y as the target microprocessor. Figure 1-1 shows the NuMaker Tomato board.

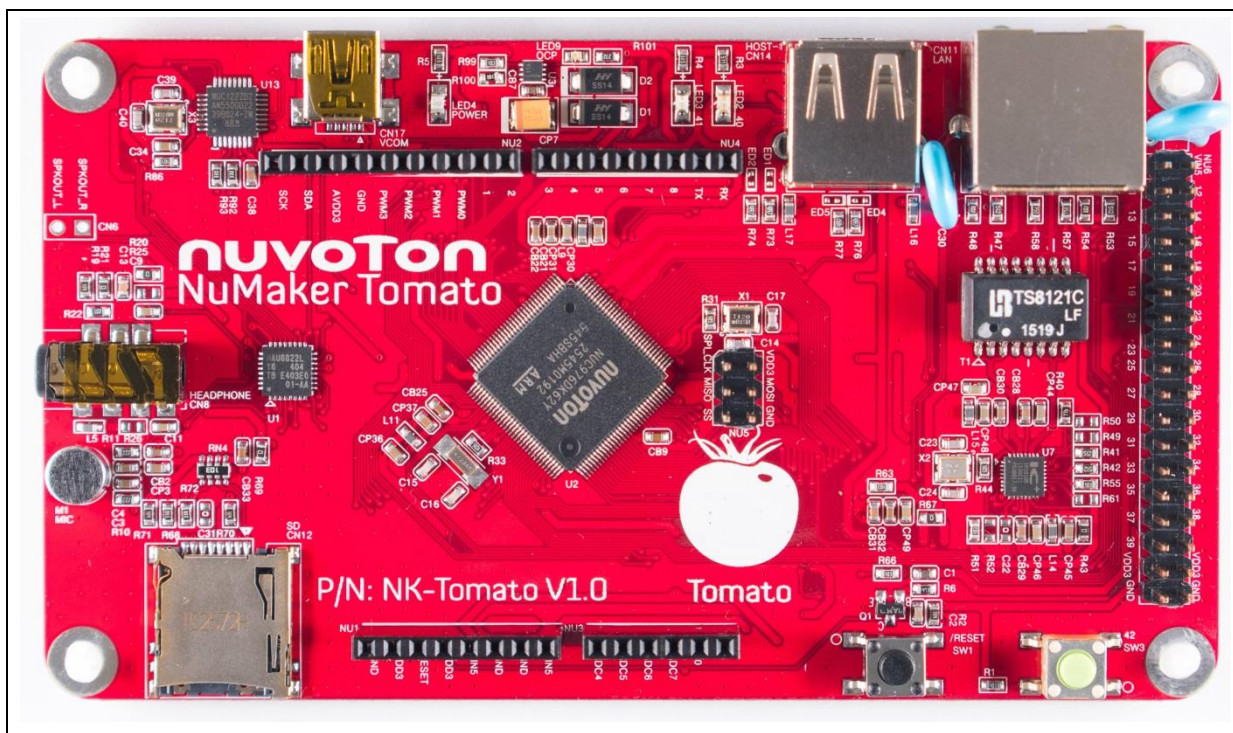


Figure 1-1 NuMaker Tomato Board

## 2 HARDWARE FEATURES

### 2.1 NuMaker Tomato Board Features

- MPU: NUC976DK62Y with ARM® ARM926EJ-S™ processor core running up to 300 MHz
- Built-in 64M Byte DDRII memory
- 8 MB SPI Flash
- USB CDC (Communications Device Class) port
- 10/100Mbps Ethernet
- Stereo Audio Headphone out
- Mono Microphone input
- 1.5W Mono Speaker output
- MicroSD card slot up to 32G Byte
- USB 2.0 High Speed Host
- USB 2.0 High Speed Device

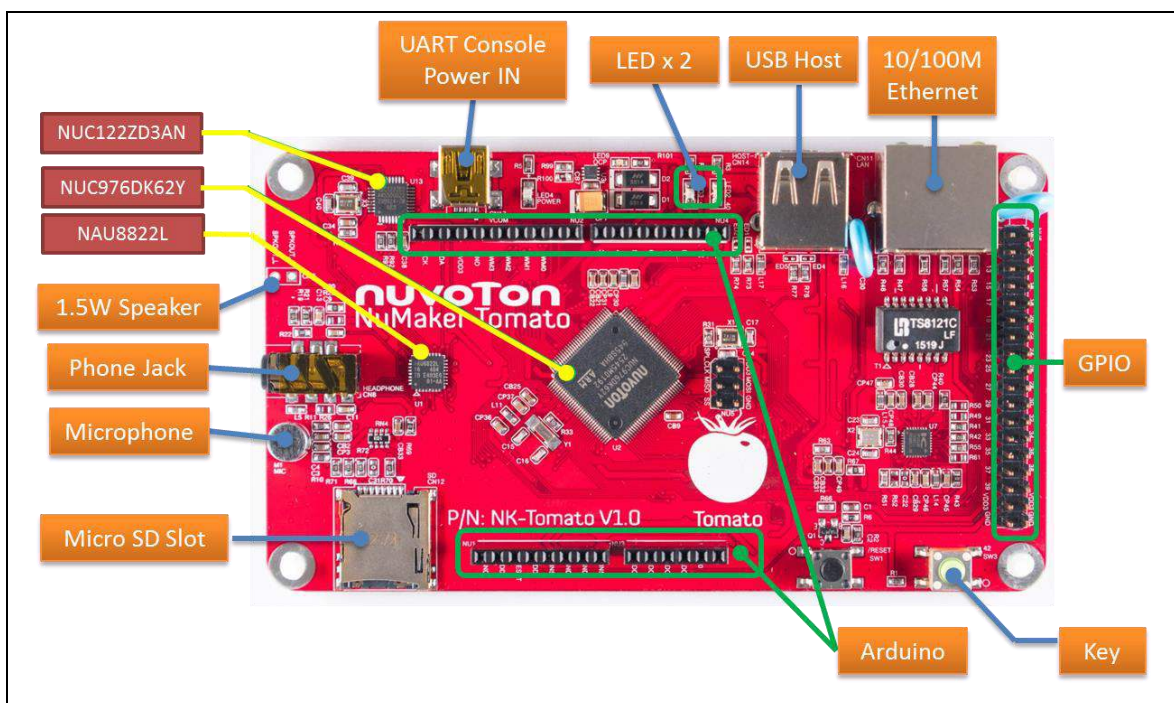


Figure 2-1 NuMaker Tomato Board Features

## 2.2 NUC976DK62Y Features

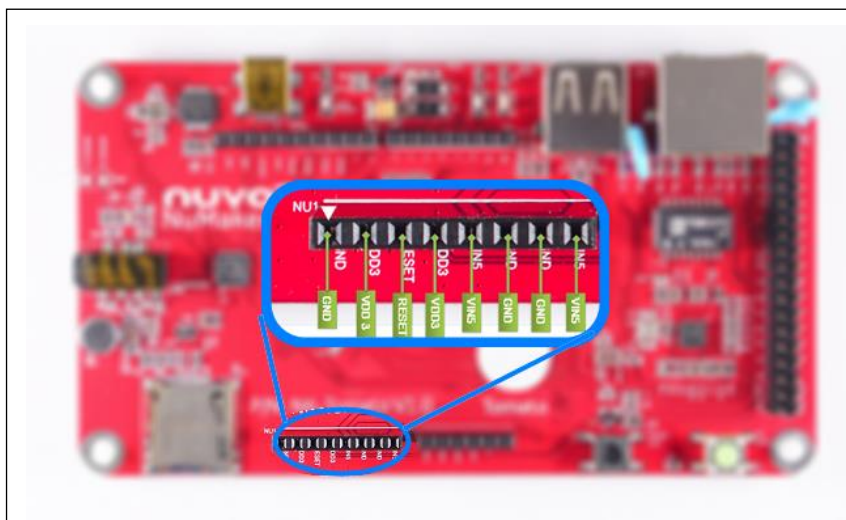
- Core
  - ARM® ARM926EJ-S™ processor core running up to 300 MHz
  - Supports 16 KB instruction cache and 16 KB data cache
- Embedded SRAM and ROM
  - Embedded 64MB DDRII
  - Supports up to four booting modes:  
USB / eMMC / NAND Flash / SPI Flash
- Ethernet MAC Controller
  - Supports IEEE Std. 802.3 CSMA/CD protocol
  - Supports packet time stamping for IEEE Std. 1588 protocol
  - Supports 10 and 100 Mbps operations
  - Supports RMIII interface to Ethernet physical layer PHY
  - Supports Ethernet physical layer PHY management through MDC and MDIO interface
- USB 2.0 Controller
  - Supports USB Revision 2.0 specification
  - Supports one set of USB 2.0 High-Speed (HS) Device/Host with embedded transceiver
  - Supports one set of USB 2.0 High-Speed (HS) Host with embedded transceiver
  - Supports 2048 Bytes internal SRAM for USB host function and 4096 Bytes internal SRAM for USB device function
- Flash Memory Interface
  - Supports NAND Flash interface
  - Supports 8-bit data bus width
  - Supports SLC and MLC type NAND Flash device
  - Supports 512 B, 2 KB, 4 KB and 8 KB page size NAND Flash device
  - Supports ECC4, ECC8, ECC12, ECC15 and ECC24 BCH algorithm for ECC code generation, error detection and error correction
  - Supports eMMC Flash interface
- I<sup>2</sup>S Controller
  - I<sup>2</sup>S interface
    - ◆ Supports both mono and stereo
    - ◆ Supports both record and playback
    - ◆ Supports 8-bit, 16-bit, 20-bit and 24-bit data precision
    - ◆ Supports master and slave mode
  - PCM interface
    - ◆ Supports 2 slots mode to connect 2 devices
    - ◆ Supports 8-bit, 16-bit, 20-bit and 24-bit data precision
    - ◆ Supports master mode
- JPEG Codec
  - Supports Baseline Sequential mode JPEG codec function compliant with ISO/IEC 10918-1 international JPEG standard
- Crypto Engine
  - PRNG
  - Supports 64-bit, 128-bit, 192-bit and 256-bit key generation
  - DES / 3DES / AES / SHA / HMAC
  - Supports both encryption and decryption
- Periphery

- UART/ SPI / I<sup>2</sup>C / 4-channel PWM / 12-bit ADC / CAN

### 3 BOARD EXTENSION INTERFACES

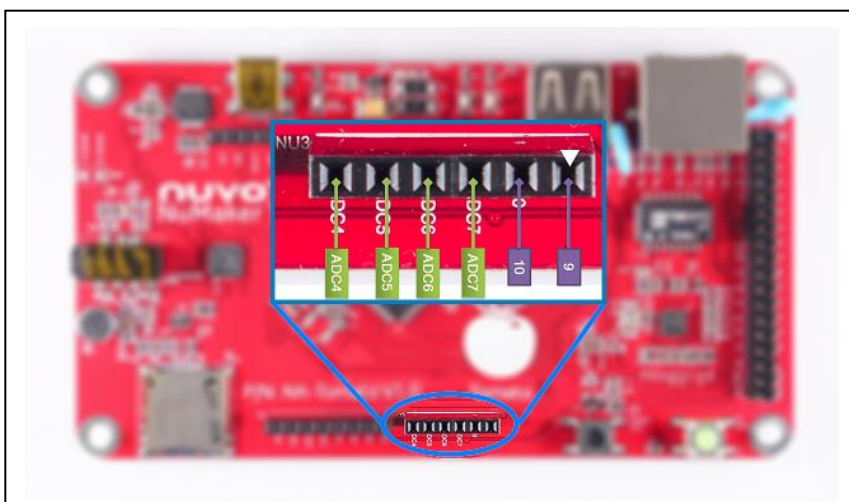
- NU1

NU1	Name
Pin 1	GND
Pin 2	VDD3
Pin 3	RESET
Pin 4	VDD3
Pin 5	VIN5
Pin 6	GND
Pin 7	GND
Pin 8	VIN5



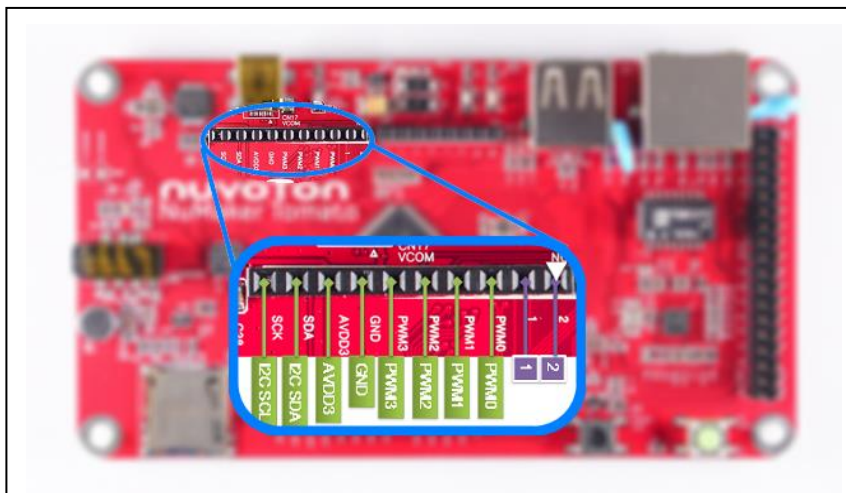
- NU3

NU3	Name	MPU Pin
Pin 1	9 (SW3)	GPF10
Pin 2	10 (LED2)	GPD7
Pin 3	ADC7	AIN7
Pin 4	ADC6	AIN6
Pin 5	ADC5	AIN5
Pin 6	ADC4	AIN4



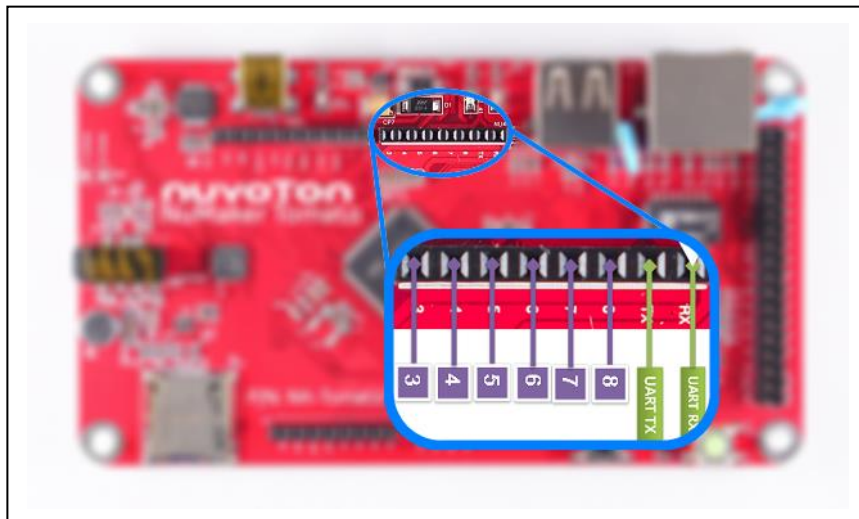
- NU2

NU2	Name	MPU Pin
Pin 1	2	GPA10
Pin 2	1	GPA11
Pin 3	PWM0	GPA12
Pin 4	PWM1	GPA13
Pin 5	PWM2	GPA14
Pin 6	PWM3	GPA15
Pin 7	GND	GND
Pin 8	AVDD3	AVDD3 V
Pin 9	SDA	I2C SDA
Pin 10	SCK	I2C SCL



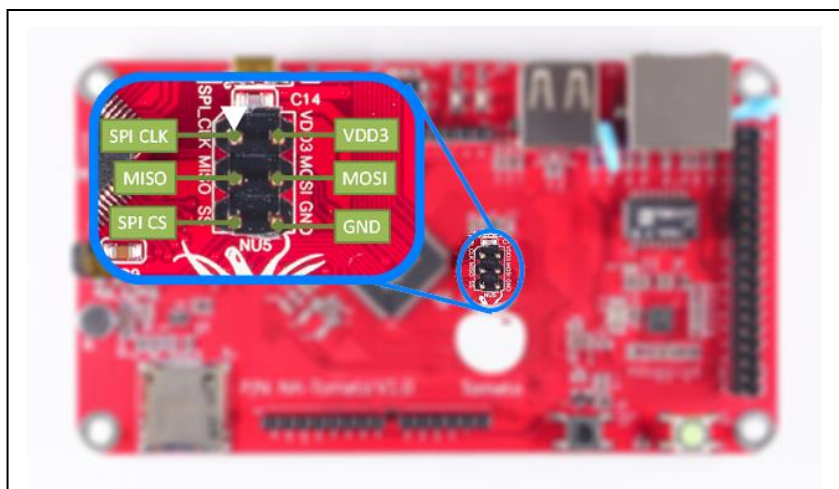
● NU4

NU2	Name	MCU Pin
Pin 1	RX	UART RX
Pin 2	TX	UART TX
Pin 3	8	GPI4
Pin 4	7	GPI3
Pin 5	6	GPB2
Pin 6	5	GPB3
Pin 7	4	GPA8
Pin 8	3	GPA9



● NU5

NU5	Name	MCU Pin
Pin 1	SPICLK	SPI CLK
Pin 2	VDD3	3.3V
Pin 3	MISO	SPI MISO
Pin 4	MOSI	SPI MOSI
Pin 5	SS	SPI CS
Pin 6	GND	GND

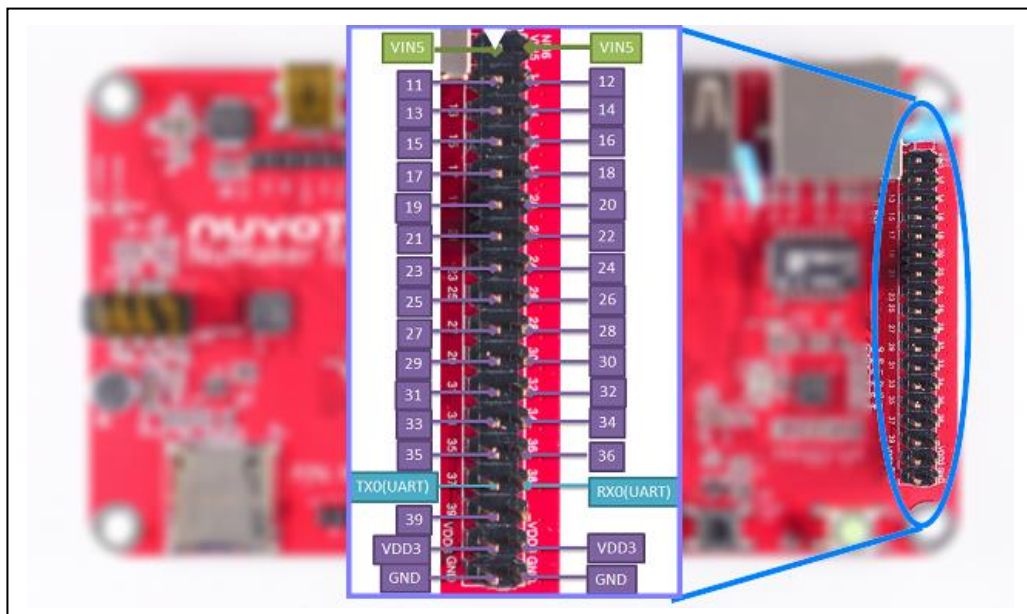


● NU6

Name	MCU Pin	NU6	NU6	MCU Pin	Name
VIN5	VIN5	Pin 1	Pin 2	VIN5	VIN5
11	GPA0	Pin 3	Pin 4	GPA1	12
13	GPA2	Pin 5	Pin 6	GPA3	14
15	GPA4	Pin 7	Pin 8	GPA5	16
17	GPA6	Pin 9	Pin 10	GPA7	18
19	GPG6	Pin 11	Pin 12	GPG7	20
21	GPG8	Pin 13	Pin 14	GPG9	22
23	GPI7	Pin 15	Pin 16	GPI8	24
25	GPI9	Pin 17	Pin 18	GPI10	26



27	GPI11	Pin 19	Pin 20	GPI12	28
29	GPI13	Pin 21	Pin 22	GPI14	30
31	GPI15	Pin 23	Pin 24	GPJ0	32
33	GPJ1	Pin 25	Pin 26	GPJ2	34
35	GPJ3	Pin 27	Pin 28	GPJ4	36
TX0(UART)	GPE0	Pin 29	Pin 30	GPE1	RX0(UART)
39(LED3)	GPH1	Pin 31	Pin 32	NC	NC
VDD3	VDD3	Pin 33	Pin 34	VDD3	VDD3
GND	GND	Pin 35	Pin 36	GND	GND



## 4 GETTING STARTED WITH NUMAKER TOMATO

### 4.1 Download Software Package and USB CDC Driver

Please download the software package from Nuvoton’s official webpage:

[http://www.nuvoton.com/opencms/resource-download.jsp?tp\\_GUID=SW1020160914071736](http://www.nuvoton.com/opencms/resource-download.jsp?tp_GUID=SW1020160914071736)

The software package includes the Linux 3.10 kernel, precompiled Python bin, USB Wi-Fi driver, Periphery sample code and application demo code.

Please download Nuvoton USB CDC driver from Nuvoton’s official webpage:

[http://www.nuvoton.com/opencms/resource-download.jsp?tp\\_GUID=SW0620160913015934](http://www.nuvoton.com/opencms/resource-download.jsp?tp_GUID=SW0620160913015934)

### 4.2 Install Nuvoton USB CDC Driver

Please execute the “NuvotonCDC\_V1.00.001\_Setup.exe” to install the driver.

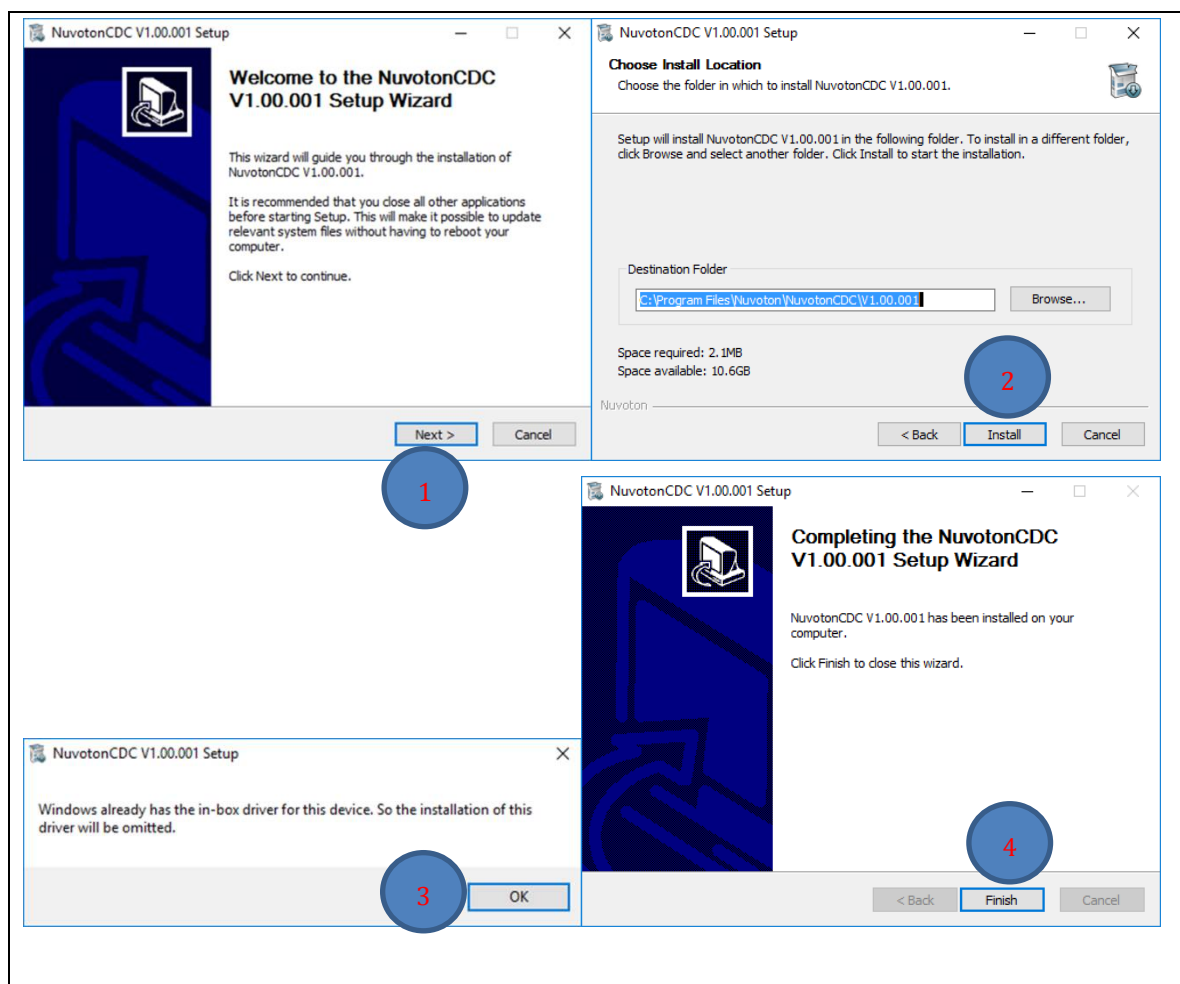


Figure 4-1 Software Package List

### 4.3 Copy Software Package to MicroSD Card

Unzip the software package file to SD card. You will see folders and kernel file.

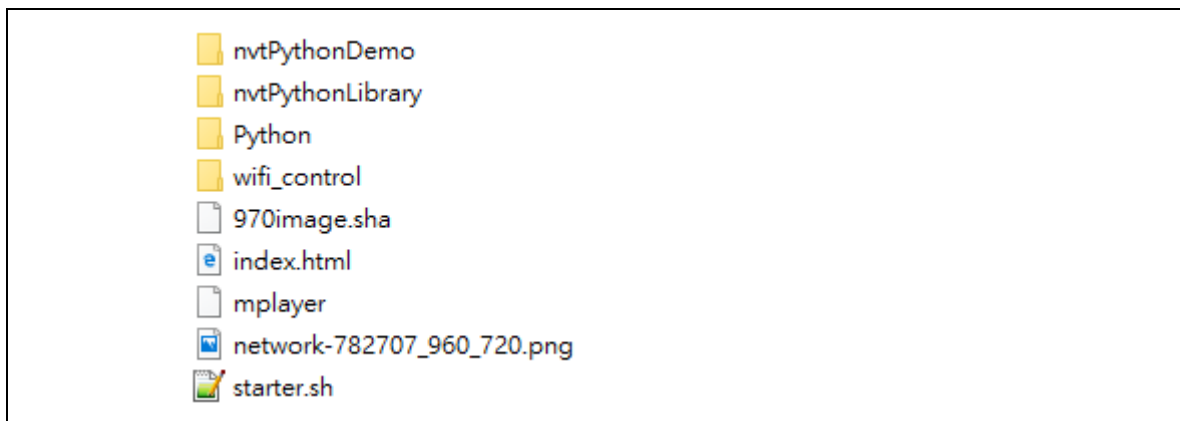


Figure 4-2 Software Package List

### 4.4 Connect and Set Up Hardware

The hardware setup is shown as Figure 4-2. Connect CN17 (MiniUSB) to a USB port on PC.

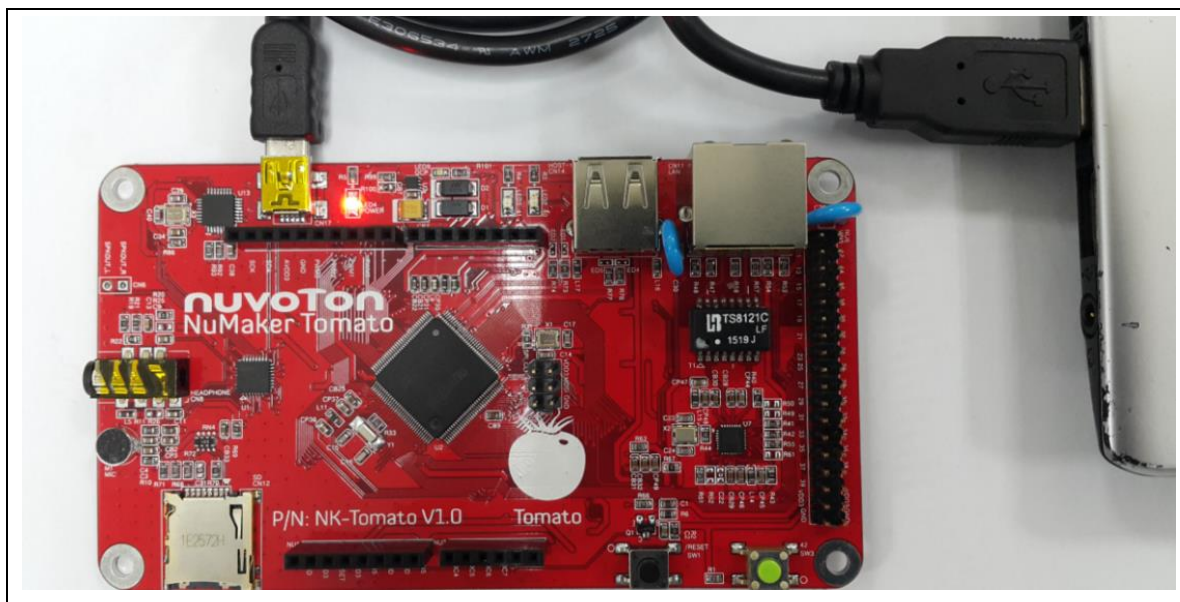


Figure 4-3 NuMaker Tomato Hardware Setup

### 4.5 Open USB CDC Serial COM Port

- Check the COM port number from device manager.

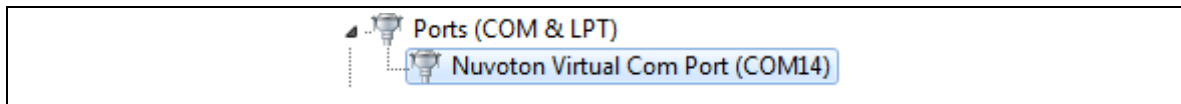


Figure 4-4 Check COM Port Number from Windows Device Manager

1. Use HyperTerminal, Putty or TeraTerm to open the serial COM port, and set the baud rate to 115200.
2. Push the RESET key.
3. Wait the Linux boot up for about 5 sec. If it is finished, you will see the " ~# " message on HyperTerminal.

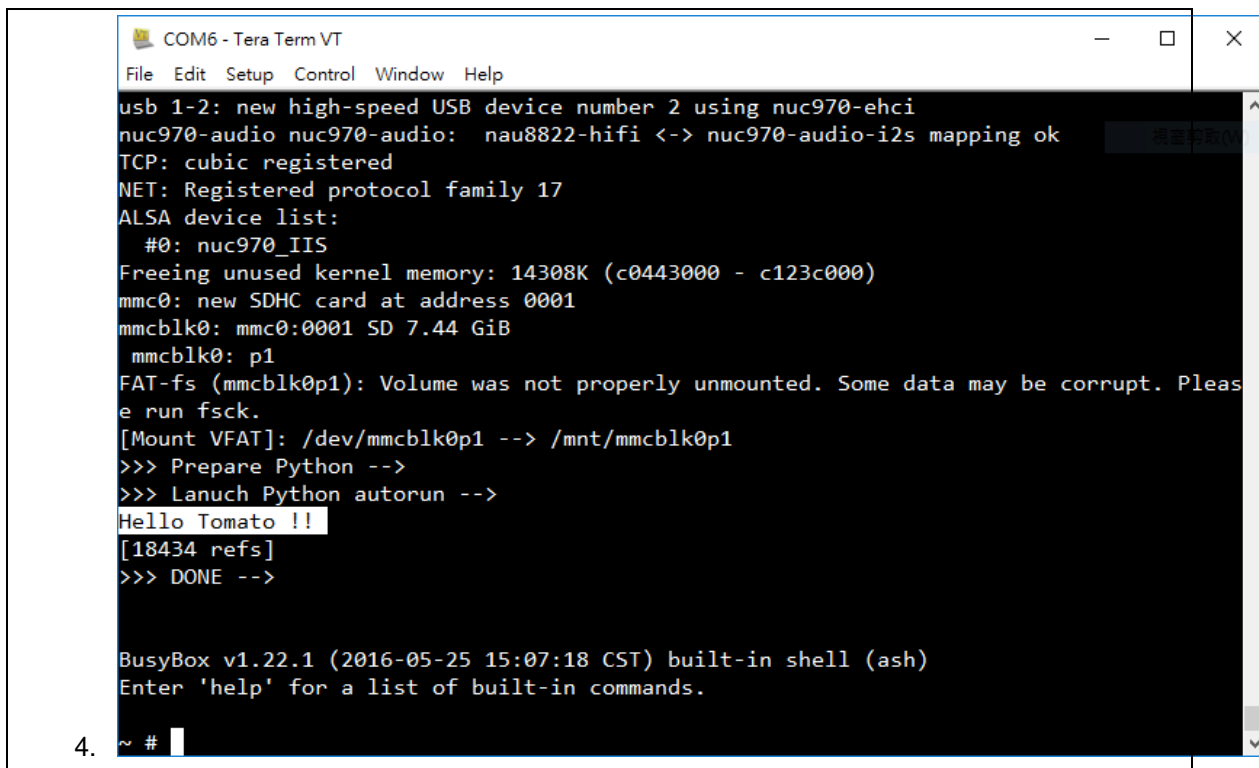


Figure 4-5 NuMaker Uni and Nu-Link Mini Connection



## 5 EXAMPLE OF EXECUTING SAMPLE CODE

User can modify starter.sh to change auto executing Python sample code

Example: Start a HTTP server.

`/mnt/mmcblk0p1/Python/bin/python -m SimpleHTTPServer`

```

10
11 trap control_c SIGNINT
12
13
14 # echo ">>> Setup WiFi -->"
15 # cd /mnt/mmcblk0p1/wifi_control
16 # ./wifi.sh
17
18 # echo ">>> Get time -->"
19 # busybox ntpd -p watch.stdtime.gov.tw -qNnd
20
21 echo ">>> Prepare Python -->"
22 mkdir /usr/lib
23 mount -t ramfs none /usr/lib
24
25 cd /usr/lib
26 cp /mnt/mmcblk0p1/Python/lib/python2.7/lib-dynload/* .
27 cp /mnt/mmcblk0p1/Python/lib/python2.7/*.pyc .
28 cp /mnt/mmcblk0p1/Python/lib/python2.7/encodings/*.pyc .
29
30 cp /mnt/mmcblk0p1/Python/bin/python2.7 .
31 ln -s python2.7 python
32
33 echo ">>> Lanuch Python autorun -->"
34 /mnt/mmcblk0p1/Python/bin/python /mnt/mmcblk0p1/autorun.py
35

```

Figure 5-1 starter.sh

6 APPENDIX

6.1 Schematics

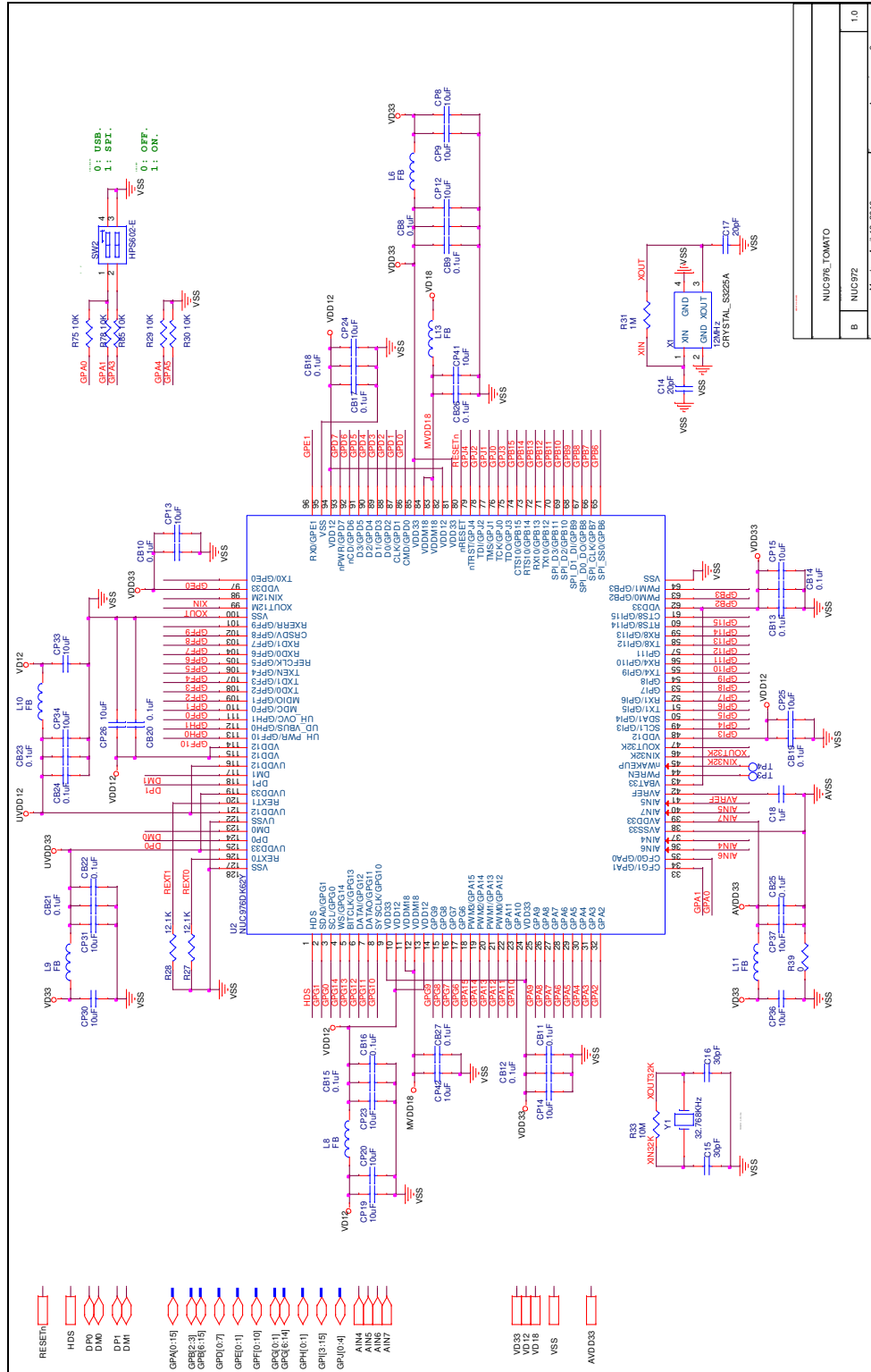


Figure 5-1 MPU Schematic

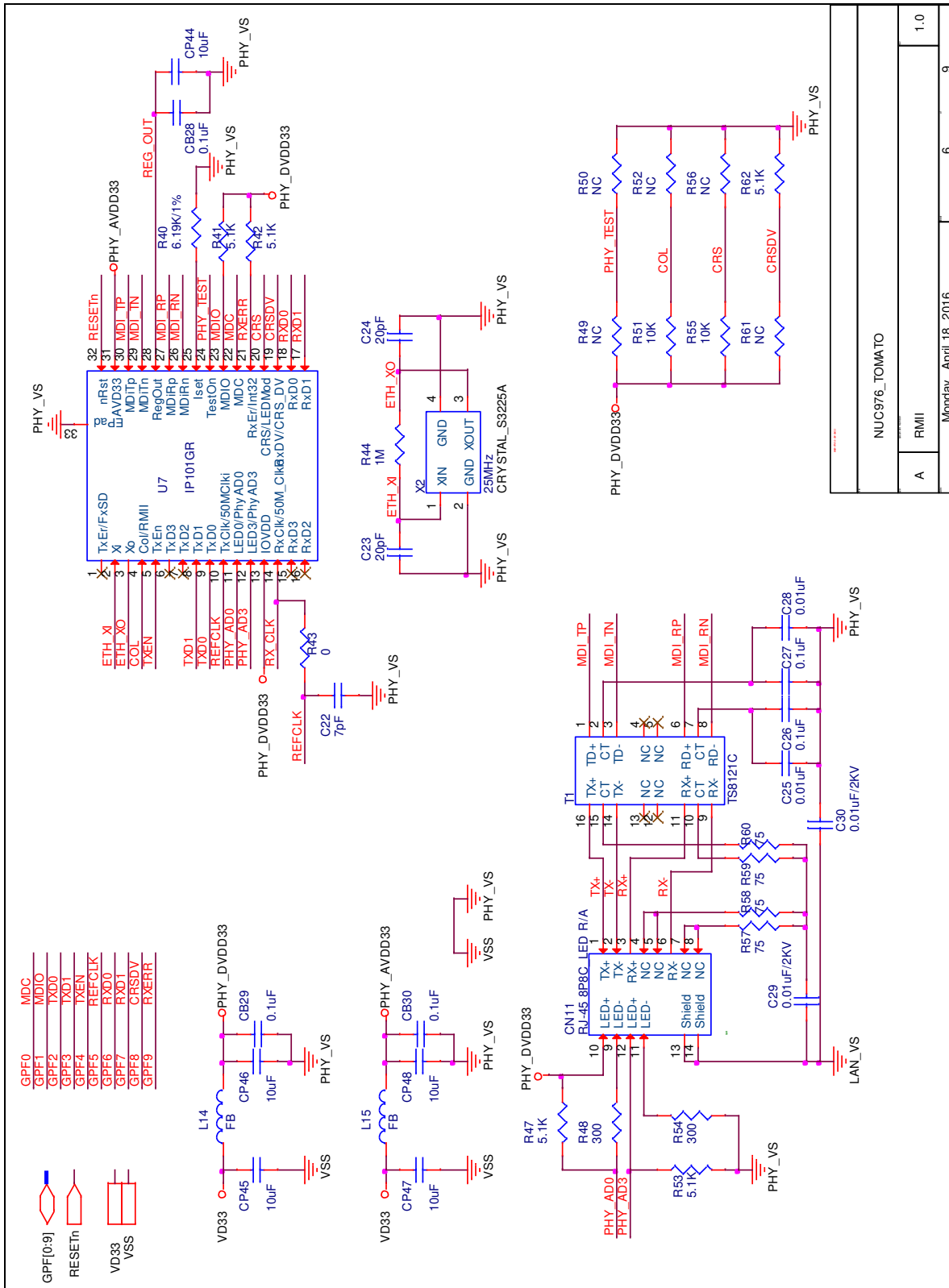
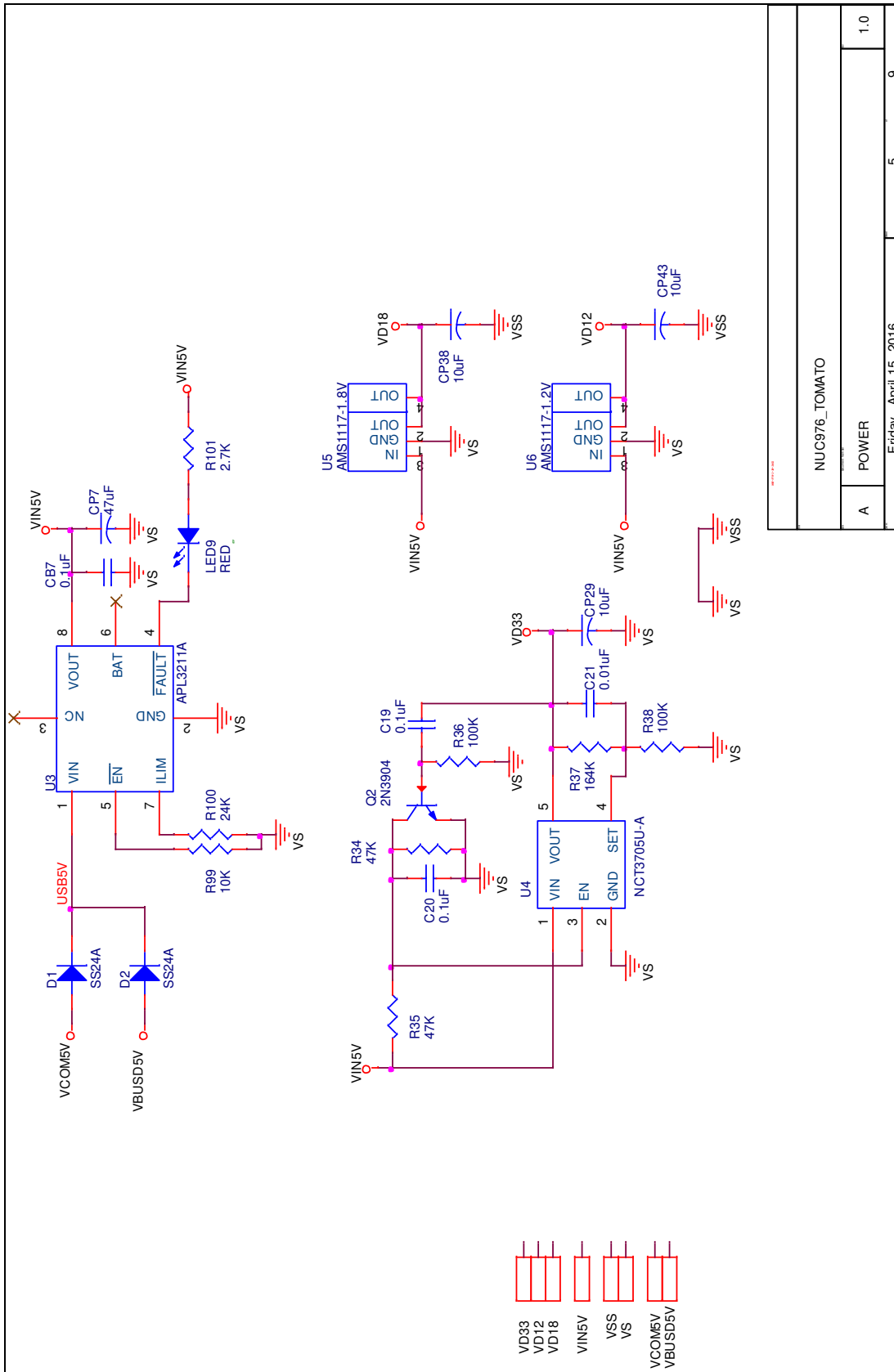


Figure 5-2 Ethernet Schematic

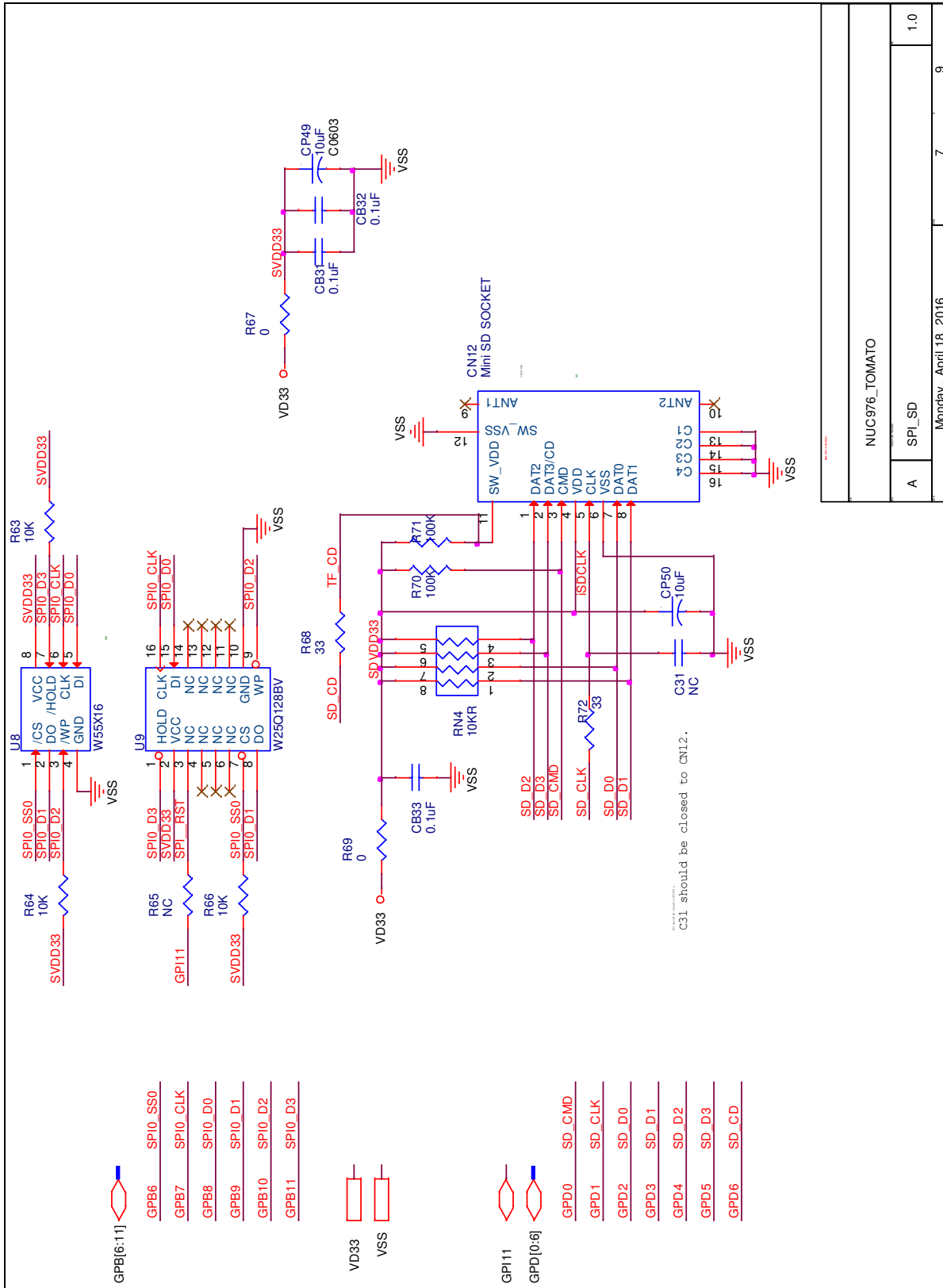
NUC976_TOMATO	
A	RMII
Monday, April 18, 2016	
6	9
1.0	





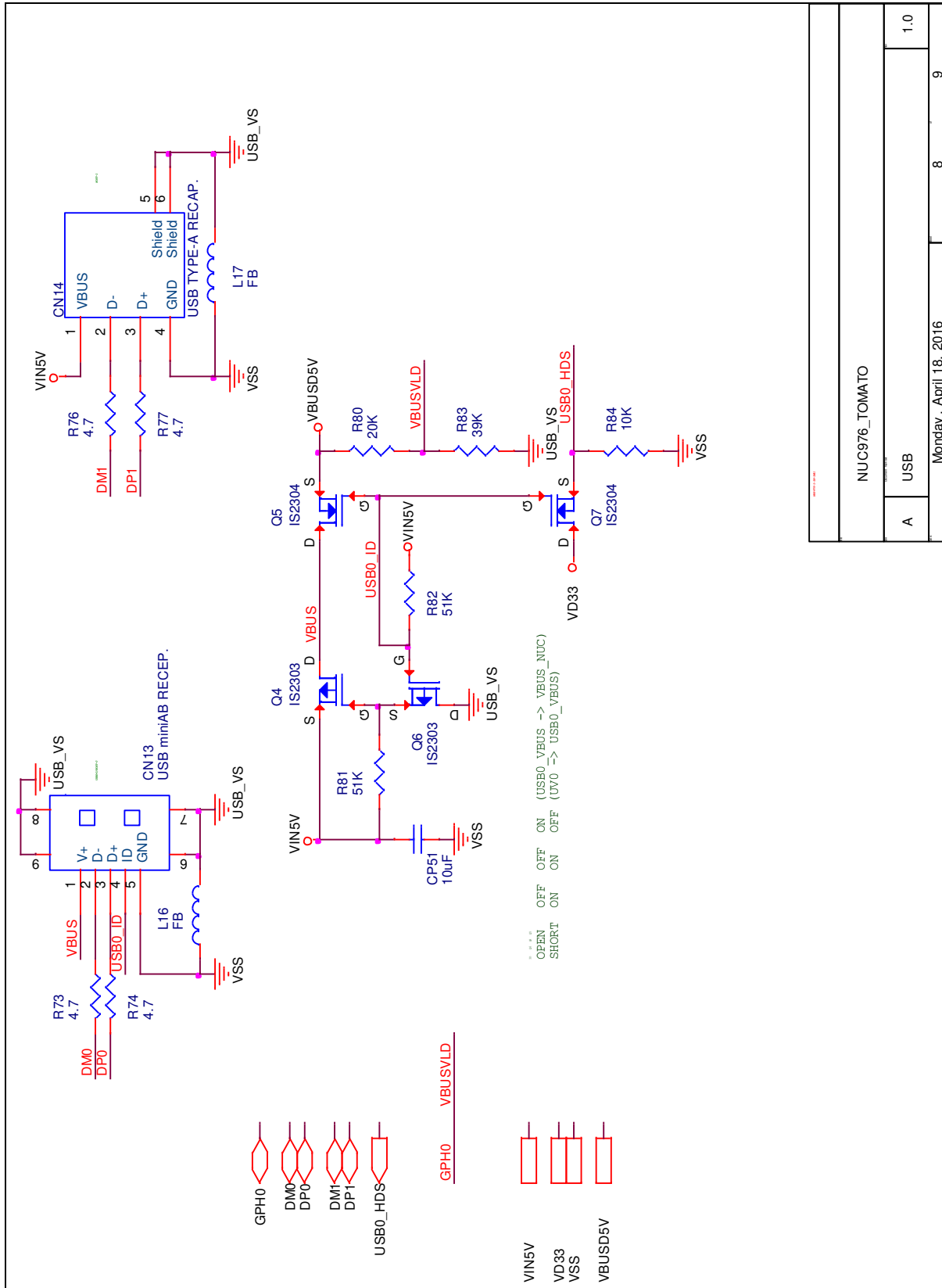
NUC976_TOMATO	
A	POWER
Friday, April 15, 2016	
5	9
1.0	

Figure 5-3 Power Schematic



NUC976_TOMATO	
A	SPI_SD
Monday, April 18, 2016	
7	9
1.0	

Figure 5-4 MicroSD Schematic



NUC976_TOMATO	
A	USB
Monday, April 18, 2016	
8	9
1.0	

Figure 5-5 USB Schematic

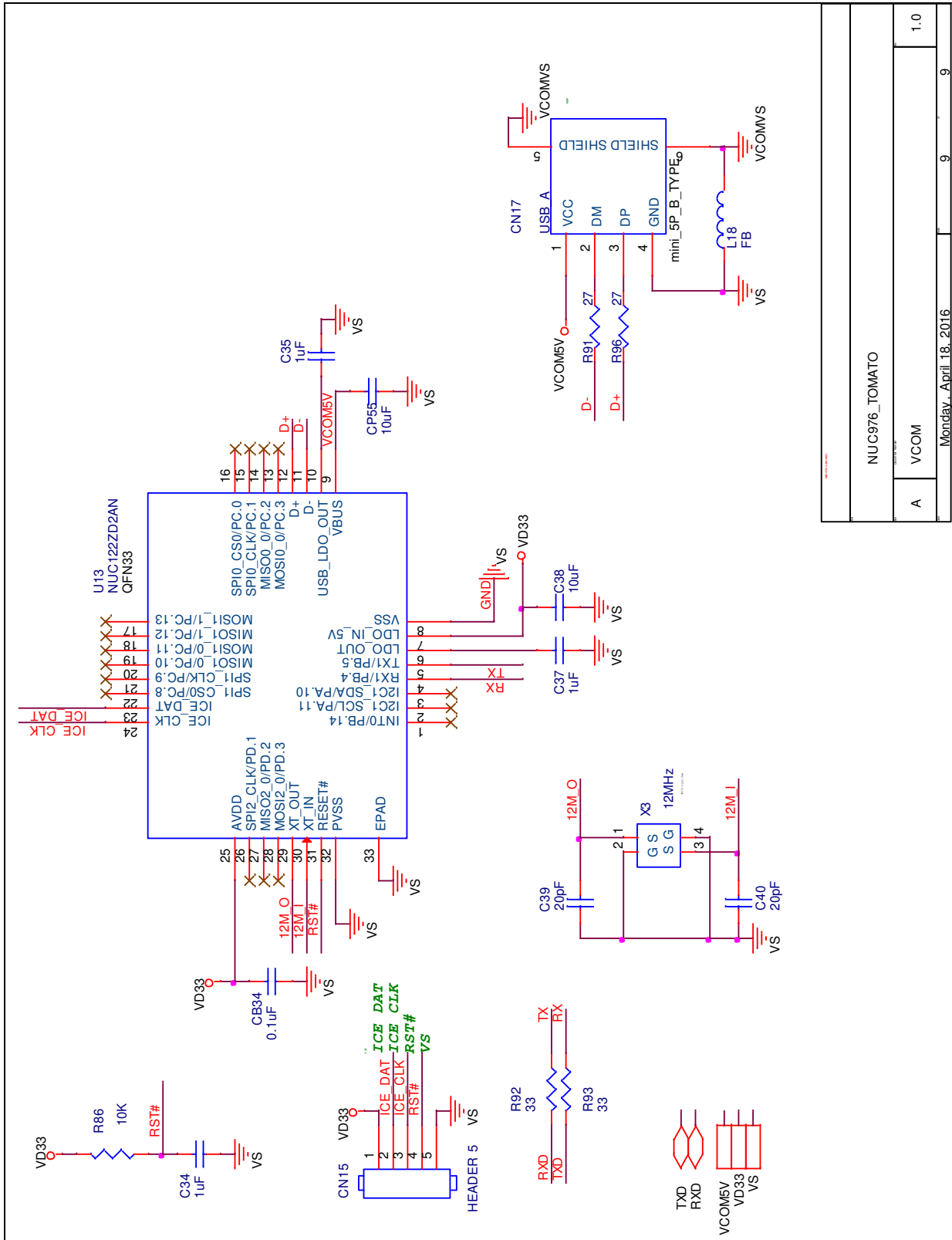


Figure 5-6 USB to UART Schematic

NUC976_TOMATO	
A	VCOM
Monday, April 18, 2016	
9	9
1.0	

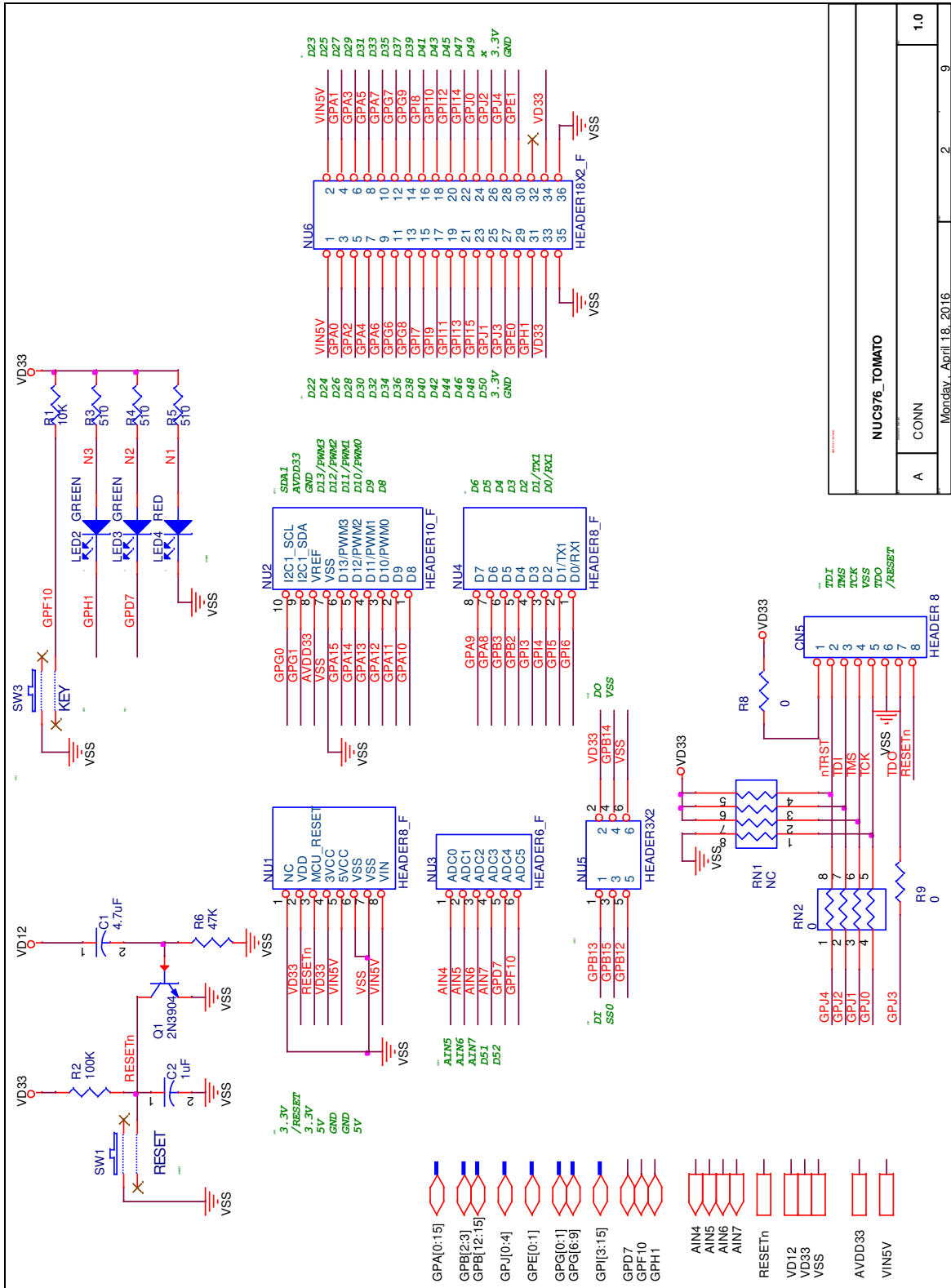


Figure 5-7 Connector Schematic

## 6.2 Python API

### 6.2.1 I<sup>2</sup>C (pysmbus)

- `write_byte_data(i2c_addr, register, value)` - I<sup>2</sup>C writes a byte
- `read_byte_data(i2c_addr, register, value)` - I<sup>2</sup>C reads a byte

Example:

```
from smbus import SMBus
bus = SMBus(1) # Opens /dev/i2c-1
bus.write_byte_data(0x54, 0x13, 0xff)
```

### 6.2.2 SPI (spidev)

- `open(bus, devcie)` - Open a SPI device
- `readbytes(n)` - read n byte from SPI
- `writebytes(list of values)` - write a list to SPI
- `xfer(list of values[, speed_hz, delay_usec, bits_per_word])` - set a SPI device and send start.
- `close()` - Cloce SPI device

Example:

```
import spidev
spi = spidev.SpiDev()
spi.open(bus, device)
to_send = [0x01, 0x02, 0x03]
spi.xfer(to_send, 1000)
```

### 6.2.3 GPIO (gpio)

- `setup(channel, inout)` - Set gpio channel
- `setmode(version)` - Set Tomato board version (beta or v1)
- `input(channel)` - read gpio
- `output(channel)` - write gpio

Example:

```
import gpio as GPIO
GPIO.setmode(GPIO.TOMATOV1_BOARD)
GPIO.setup(LED2_PIN, GPIO.OUT)
GPIO.setup(LED3_PIN, GPIO.OUT)
for x in range(5):
    GPIO.output(LED2_PIN, GPIO.LOW)
    time.sleep(1)
    GPIO.output(LED2_PIN, GPIO.HIGH)
    time.sleep(1)
```

### 6.2.4 PWM (pwm)

- start(channel, duty cycle, frequency) - Set PWM channel
- ChangeFrequency(channel, frequency) - change PWM frequency
- ChangeDutyCycle(channel, duty cycle) - change PWM duty
- stop(channel) - close PWM channel

Example:

```
import pwm as PWM
PWMCH=0
PWM.start(PWM_CH, 50, 70000)
PWM.ChangeFrequency(PWM_CH, 100000)
PWM.ChangeDutyCycle(PWM_CH, 30)
PWM.stop(PWM_CH)
```

### 6.2.5 ADC (adc)

- adc\_get(channel) - read adc channel

Example:

```
import adc as ADC
ADC_0 = 4
print "ADC=%d"%ADC.get(ADC_0)
```

### 6.2.6 UART(pyserial)

- Serial(port=None, baudrate=9600, bytesize=EIGHTBITS, parity=PARITY\_NONE, stopbits=STOPBITS\_ONE, timeout=None, xonxoff=False, rtscts=False, write\_timeout=None, dsrdtr=False, inter\_byte\_timeout=None) - UART set
- open() - open UART port
- read(size) - read data
- readline() - read a line data
- write(data) - write data
- close() - close UART

Example:

```
import serial
ser = serial.Serial('/dev/ttyS1', 115200, timeout=0)
print ser.readline()
```

## 7 REVISION HISTORY

Date	Revision	Description
2016.09.30	1.00	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.

---

*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.*