QMK Keyboard

Example Code Introduction for 32-bit NuMicro® Family

Document Information

Application	This example code uses QMK firmware keyboard open source to implement a keyboard on NUC029.
BSP Version	NUC029xGE_Series_BSP_CMSIS_V3.00.004
Hardware	NuMaker-NUC029SGE V1.0

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 and 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

1. Overview

QMK (Quantum Mechanical Keyboard) is an open source community centered around developing computer input devices, such as keyboards, mice, and MIDI devices. A keyboard product can use QMK firmware as its official and default firmware, or support updating to QMK firmware. Then users can modify the keyboard firmware itself and make their own customized keyboard. Besides the open source, QMK also provides development tools, for example, QMK MSYS, QMK configurator, and QMK Toolbox, etc. This example code uses the QMK keyboard firmware and QMK tools to implement a simulated keyboard.

1.1 Principle

The simulated keyboard on NUC029xGE includes a keyboard firmware itself and a firmware update flow. The keyboard firmware is in APROM and the firmware update flow is responsible for bootloader in LDROM. Figure 1-1 shows NUC029xGE memory map in this example code.

The APROM code is mainly executed after system start-up and it is recognized as an USB HID keyboard device on computer. This example code uses two buttons on NuTFT_LCM board, SW1 and SW2, to simulate two keys on keyboard. SW1 is key **Esc** and SW2 is key **1**.

The firmware update flow is responsible for LDROM bootloader. The way to enter bootloader is: hold on the key **Esc** and press **RESET** button on NuMaker-NUC029SGE board. The APROM code is executed again and check if the key **Esc** is pressed, then jump to LDROM bootloader code. It is recognized as a USB DFU device on computer and user can update firmware by DFU protocol.

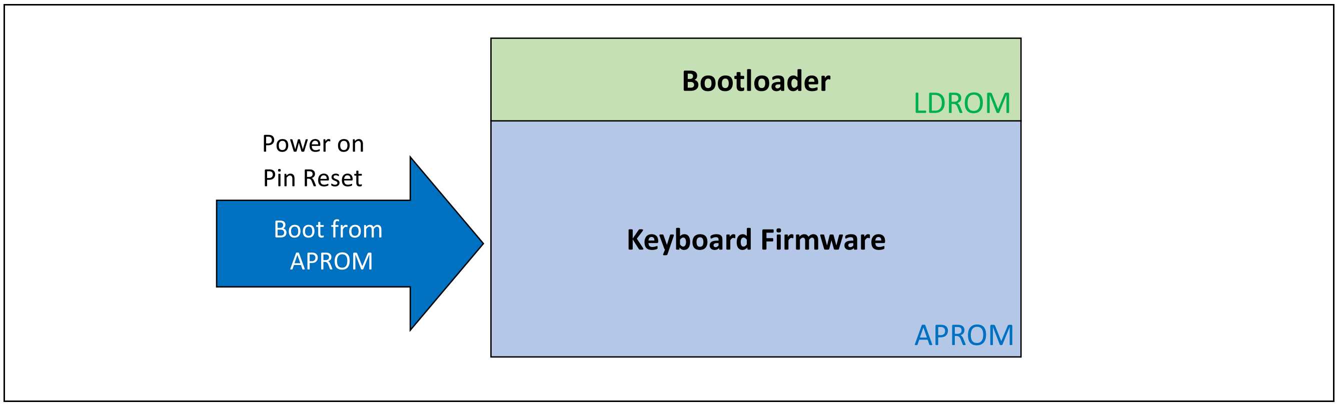
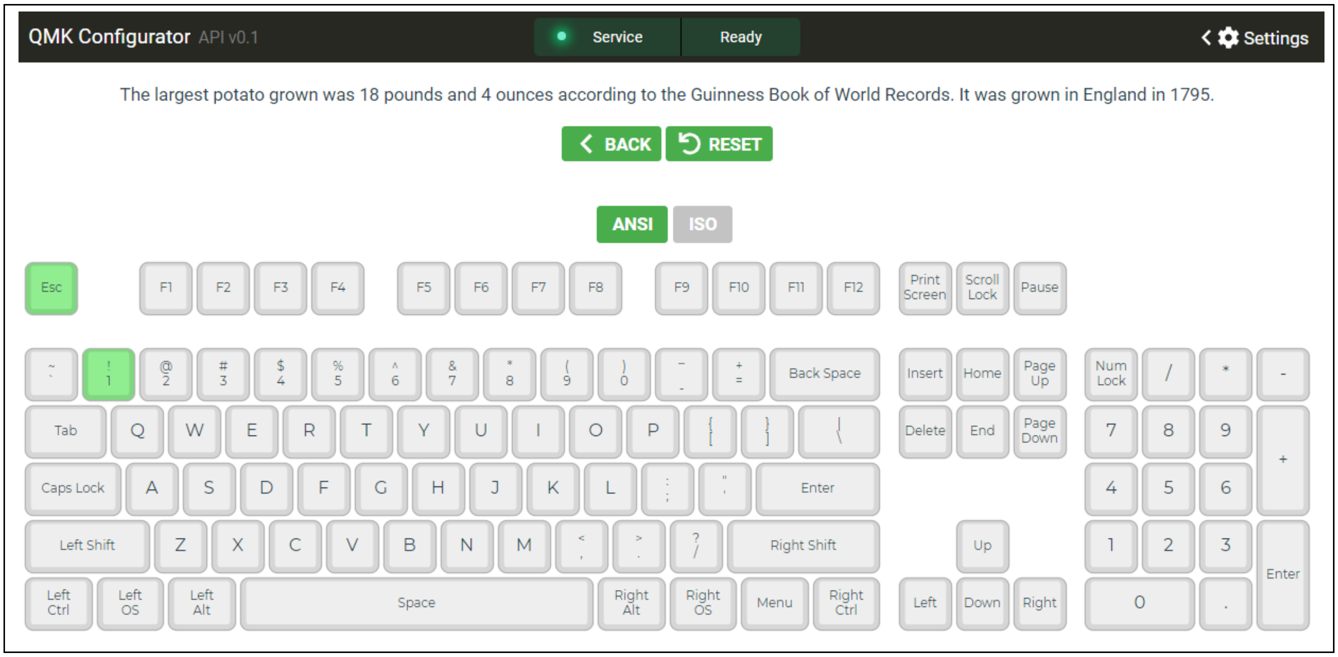


Figure 1-1 NUC029xGE Memory Map

1.2 Demo Result

To use keyboard test tool provided by QMK Configurator to test keyboard function, visit the website and press the two buttons on NuTFT_LCM board. You can see the keys **Esc** and **1** are pressed. Figure 1-2 shows the QMK Configurator keyboard test result.



The largest potato grown was 18 pounds and 4 ounces according to the Guinness Book of World Records. It was grown in England in 1795.

Figure 1-2 QMK Configurator Keyboard Test Result

2. Code Description

2.1 Main Function

Initialize and setup:

- platform_setup() is used to setup platform NUC029xGE hardware, including: system clock, GPIO, USB and systick.
- protocol_setup() is used to setup USB function.
- keyboard_setup() is used to setup keyboard function, and setup flash if VIA function is enabled.
- protocol_init() is used to initial and start USB, VIA and bootmagic function.

Main loop:

- protocol_task() is responsible for keyboard matrix scanning and USB service.
- housekeeping_task() is responsible for other tasks which need to be executed periodically, for example, HIRC48 trim.

qmk_firmware/quantum/main.c

```
int main(void) {
    /* Initial and setup */
    platform_setup();
    protocol_setup();
    keyboard_setup();

    protocol_init();

    /* Main loop */
    while (true) {
        protocol_task();
        housekeeping_task();
    }
}
```

2.2 Key

This example code uses two buttons, SW1 and SW2, on NuTFT_LCM board to simulate two keys on keyboard. SW1 is connected to PC.4 on NuMaker-NUC029SGE board and SW2 is connected to PC.5. In config.h, define DIRECT_PIN value as {{C4, C5}}. In nuc029xgex.h, define LAYOUT including two components, k00 and k01. In keymap.json, define the two components of LAYOUT as key code KC_ESCAPE and KC_1. The key code is defined in keycodes.h file and user can modify them to other desired key codes.

qmk_firmware\keyboards\handwired\numaker\nuc029xgex\config.h

```
#define DIRECT_PINS { { C4, C5 } }
```

qmk_firmware\keyboards\handwired\numaker\nuc029xgex\nuc029xgex.h

```
#define LAYOUT( k00, k01 ) { { k00, k01} }
```

qmk_firmware\keyboards\handwired\numaker\nuc029xgex\keymaps\default\keymap.json

```
{
    "keyboard": "handwired/numaker/nuc029xgex",
    "keymap": "default",
    "layout": "LAYOUT",
    "layers": [
        [
            "KC_ESCAPE",
            "KC_1"
        ]
    ]
}
```

qmk_firmware\quantum\keycodes.h

```
enum qk_keycode_defines {
// Keycodes

.
.
.

    KC_1 = 0x001E,
    KC_2 = 0x001F,
    KC_3 = 0x0020,
    KC_4 = 0x0021,
    KC_5 = 0x0022,
    KC_6 = 0x0023,
    KC_7 = 0x0024,
    KC_8 = 0x0025,
    KC_9 = 0x0026,
    KC_0 = 0x0027,
    KC_ENTER = 0x0028,
    KC_ESCAPE = 0x0029,

.
.
.

}
```

2.3 VIA

The VIA function can be used to modify keymap. To use this function, set VIA_ENABLE value as yes in rules.mk. The VIA function records keymap in non-volatile memory, such as EEPROM or embedded Flash. In rules.mk, set this example code to use wear leveling method to record keymap in embedded Flash. In config.h, define BACKING_STORE_WRITE_SIZE as 4 bytes which is the size written to Flash once, define NUC029xGE Flash size as 256 KB and enable APROM access.

qmk_firmware\keyboards\handwired\numaker\nuc029xgex\keymaps\via\rules.mk

```
VIA_ENABLE = yes
EEPROM_DRIVER = wear_leveling
```

```
WEAR_LEVELING_DRIVER = embedded_flash
```

qmk_firmware\keyboards\handwired\numaker\nuc029xgex\keymaps\via\config.h

```
#define BACKING_STORE_WRITE_SIZE     (4) /* 4 bytes */
#define WEAR_LEVELING_EFL_FLASH_SIZE (256<<10) /* 256KB */

#define NUC029_CONFIG_ENABLED FALSE
#define NUC029_EFL_ACCESS_APROM TRUE
#define NUC029_EFL_ACCESS_DATAFLASH FALSE
#define NUC029_EFL_ACCESS_LDROM FALSE
#define NUC029_EFL_ACCESS_CONFIG FALSE
```

2.4 Bootmagic

QMK provides bootmagic function to switch to bootloader mode. To use this function, set BOOTMAGIC_ENABLE value as yes in rules.mk. After keyboard is connected to USB, QMK firmware will detect if a specified key is pressed or not. The default specified key is **Esc** at position (0,0). If the **Esc** key is pressed, then clear EEPROM data and switch to bootloader mode. The switch process is implemented in nuc029xge.c file bootloader_jump() function.

qmk_firmware\keyboards\handwired\numaker\nuc029xgex\rules.mk

```
BOOTMAGIC_ENABLE = yes
```

qmk_firmware\quantum\bootmagic\bootmagic_lite.c

```
__attribute__((weak)) void bootmagic_lite(void) {
    // We need multiple scans because debouncing can't be turned off.
    matrix_scan();
    wait_ms(30);
    matrix_scan();

    // If the configured key (commonly Esc) is held down on power up,
    // reset the EEPROM valid state and jump to bootloader.
    // This isn't very generalized, but we need something that doesn't
    // rely on user's keymaps in firmware or EEPROM.
    uint8_t row = BOOTMAGIC_LITE_ROW;
    uint8_t col = BOOTMAGIC_LITE_COLUMN;

    if (matrix_get_row(row) & (1 << col)) {
        bootmagic_lite_reset_eeprom();

        // Jump to bootloader.
        bootloader_jump();
    }
}
```

qmk_firmware\keyboards\handwired\numaker\nuc029xgex\nuc029xgex.c

```
void bootloader_jump(void)
{
    /* Unlock register write protect function */
    SystemUnlockReg();

    /* Enable FMC ISP clock */
    CLK->AHBCLK |= CLK_AHBCLK_ISPCKEN_Msk;

    /* Set boot selection as booting from LDROM */
    FMC->ISPCTL |= FMC_ISPCTL_BS_Msk;
    /* Lock register write protect function */
    LOCKREG();

    /* System reset */
    NVIC_SystemReset();
}
```

2.5 Bootloader

In LDROM bootloader code, check if the **Esc** key is pressed, which means that user specifies it to switch to bootloader mode, and then keep in LDROM to wait firmware update. If the **Esc** key is not pressed, then run APROM keyboard code.

NUC029xGE_BSP\SampleCode\ISP\ISP_DFU\main.c

```
void GotoAprom(void)
{
    /* Reset system and boot from APROM */
    FMC->ISPCTL &= ~(FMC_ISPCTL_ISPEN_Msk | FMC_ISPCTL_BS_Msk);
    SCB->AIRCR = (V6M_AIRCR_VECTKEY_DATA | V6M_AIRCR_SYSRESETREQ);
}

int32_t main(void)
{
                                              .
                                              .
                                              .

    /* Keep in LDROM if ESC key is pressed. Otherwise, run APROM keyboard. */
    if( PC4 == 1 ) goto _APROM;

                                              .
                                              .
                                              .

_APROM:

    /* Reset system and boot from APROM */
    GotoAprom();

    /* Trap the CPU */
    while(1);
}
```

3. Software and Hardware Requirements

3.1 Software Requirements

- GitHub Repository
  - qmk/qmk_firmware
  - OpenNuvoton/NUC029xGE_BSP v3.00.004
- Development Tool
  - Git 2.37.1
  - QMK MSYS 1.7.2
  - QMK Toolbox (Nuvoton Edition)
  - Nuvoton NuMicro ICP Programming Tool 3.10
  - Keil uVersion 5.37

3.2 Hardware Requirements

- Circuit Components
  - NuMaker-NUC029SGE V1.0
  - NuTFT_LCM V1.4
- Pin Connect
  - Connect NuTFT_LCM as upper board and NuMaker-NUC029SGE as lower board.
  - Connect NuMaker-NUC029SGE side USB to PC.

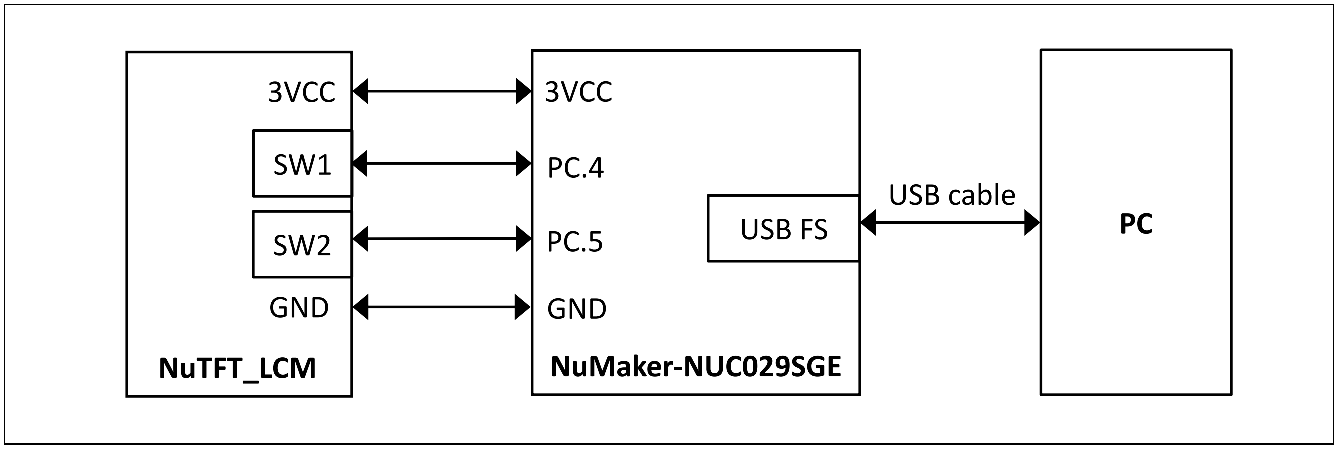


Figure 3-1 NuMaker-NUC029SGE Pin Connect

4. Directory Information

The directory structure is shown below.

🗁 EC_NUC029_QMK_Keyboard_V1.00

🗁 git_patch	Git patch file
🗁 qmk_tool	The executable file of QMK Toolbox (Nuvoton Edition)
🗁 via_json	The JSON file for VIA Tool

Figure 4-1 Directory Structure

5. Example Code Execution

This section will introduce how to download and compile keyboard and bootloader code, how to download code on board, firmware update flow and finally introduce VIA tool usage.

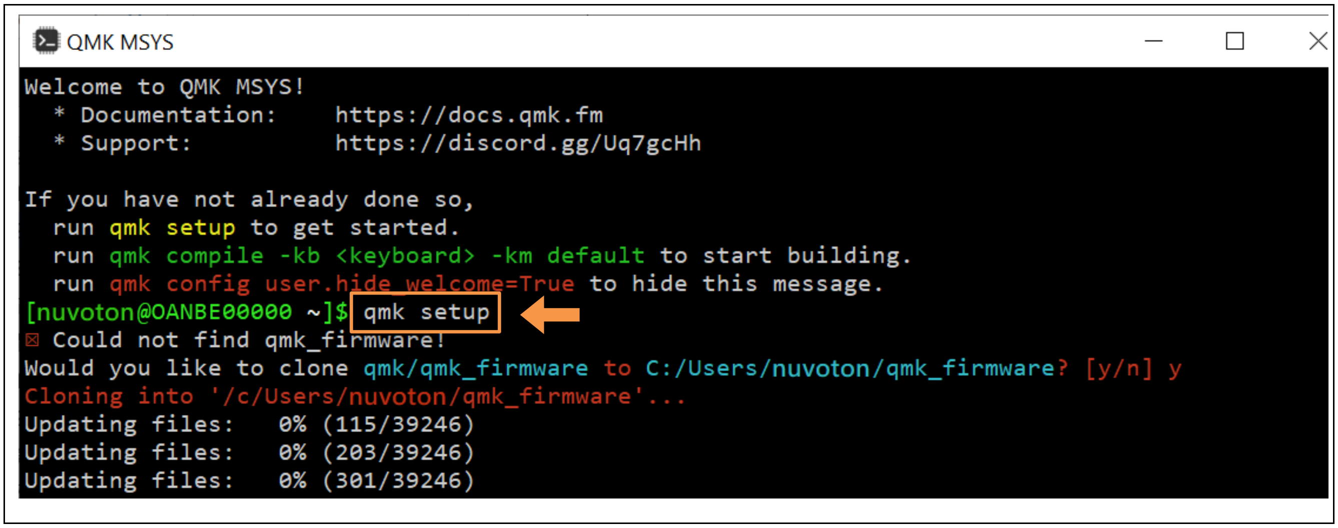
5.1 Keyboard Code

5.1.1 Download Keyboard Code

This example code is modified from QMK keyboard firmware and uses QMK MSYS as compiling tool. First, prepare development environment and download keyboard code:

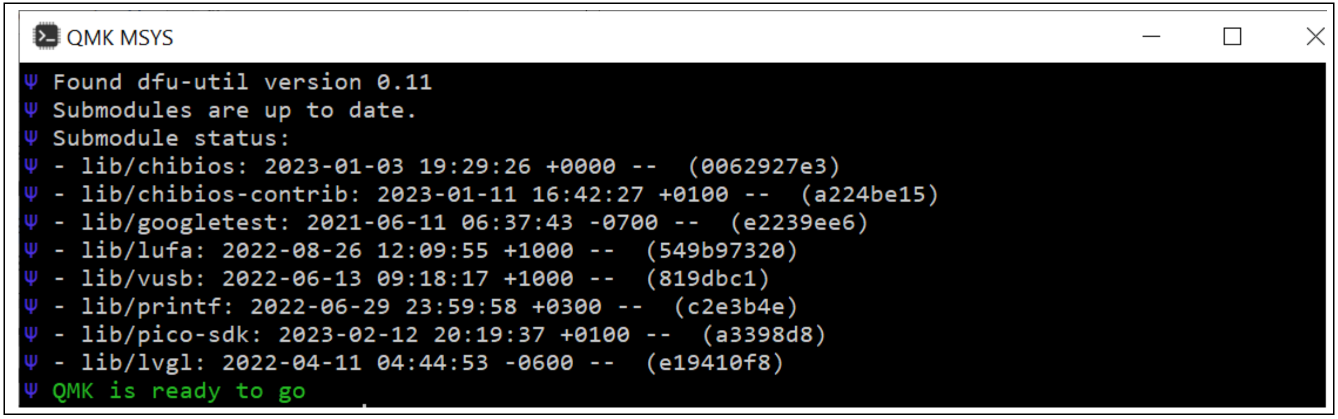
1. Install QMK MSYS.
2. Open QMK MSYS and run qmk setup command, to check development environment and download qmk_firmware code from GitHub.

```
qmk setup
```



```
Welcome to QMK MSYS!
  * Documentation:    https://docs.qmk.fm
  * Support:          https://discord.gg/Uq7gcHh

If you have not already done so,
  run qmk setup to get started.
  run qmk compile -kb <keyboard> -km default to start building.
  run qmk config user.hide_welcome=True to hide this message.
[nuvoton@OANBE00000 ~]$ qmk setup
☒ Could not find qmk_firmware!
Would you like to clone qmk/qmk_firmware to C:/Users/nuvoton/qmk_firmware? [y/n] y
Cloning into '/c/Users/nuvoton/qmk_firmware'...
Updating files:   0% (115/39246)
Updating files:   0% (203/39246)
Updating files:   0% (301/39246)
```

Figure 5-1 Run qmk Setup Command



```
Ψ Found dfu-util version 0.11
Ψ Submodules are up to date.
Ψ Submodule status:
Ψ - lib/chibios: 2023-01-03 19:29:26 +0000 --  (0062927e3)
Ψ - lib/chibios-contrib: 2023-01-11 16:42:27 +0100 --  (a224be15)
Ψ - lib/googletest: 2021-06-11 06:37:43 -0700 --  (e2239ee6)
Ψ - lib/lufa: 2022-08-26 12:09:55 +1000 --  (549b97320)
Ψ - lib/vusb: 2022-06-13 09:18:17 +1000 --  (819dbc1)
Ψ - lib/printf: 2022-06-29 23:59:58 +0300 --  (c2e3b4e)
Ψ - lib/pico-sdk: 2023-02-12 20:19:37 +0100 --  (a3398d8)
Ψ - lib/lvgl: 2022-04-11 04:44:53 -0600 --  (e19410f8)
Ψ QMK is ready to go
```

Figure 5-2 Run qmk Setup Command Finished

Next, add NUC029xGE related files into qmk_firmware code by git apply patch method.

1. Install Git.
2. Put qmk_firmware git patch file of this example code under *qmk_firmware* directory just downloaded.
3. Put chibios-contrib git patch file of this example code under *qmk_firmware\lib\chibios-contrib* directory just downloaded.

- 🗁 EC_NUC029_QMK_Keyboard_V1.00
  - 🗁 git_patch
    - 🗎 0001-update-qmk_firmware-for-NUC029xGE.patch

Figure 5-3 Example Code qmk_firmware Git Patch File Location

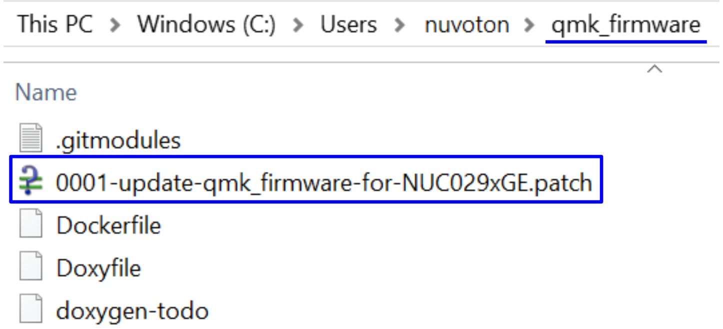

This PC › Windows (C:) › Users › nuvoton › qmk_firmware
Name
.gitmodules
0001-update-qmk_firmware-for-NUC029xGE.patch
Dockerfile
Doxyfile
doxygen-todo

Figure 5-4 Put qmk_firmware Git Patch File into qmk_firmware Directory

- 🗁 EC_NUC029_QMK_Keyboard_V1.00
  - 🗁 git_patch
    - 🗎 0002-update-chibios_contrib-for-NUC029xGE.patch

Figure 5-5 Example Code chibios_contrib Git Patch File Location

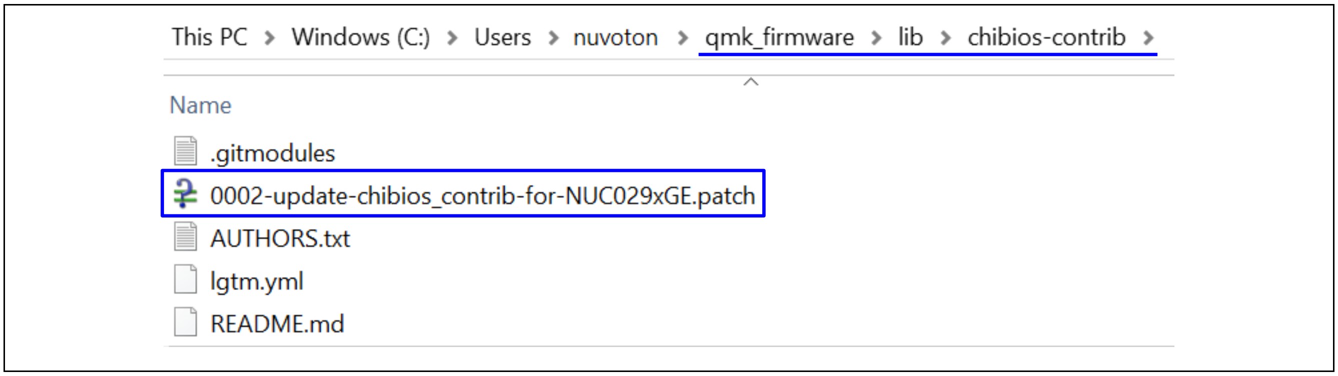

This PC > Windows (C:) > Users > nuvoton > qmk_firmware > lib > chibios-contrib >
Name
.gitmodules
0002-update-chibios_contrib-for-NUC029xGE.patch
AUTHORS.txt
lgtm.yml
README.md

Figure 5-6 Put chibios-contrib Git Patch File into qmk_firmware\lib\chibios-contrib Directory

4. Under qmk_firmware directory, right-click and select **Git Base Here** to open Git.

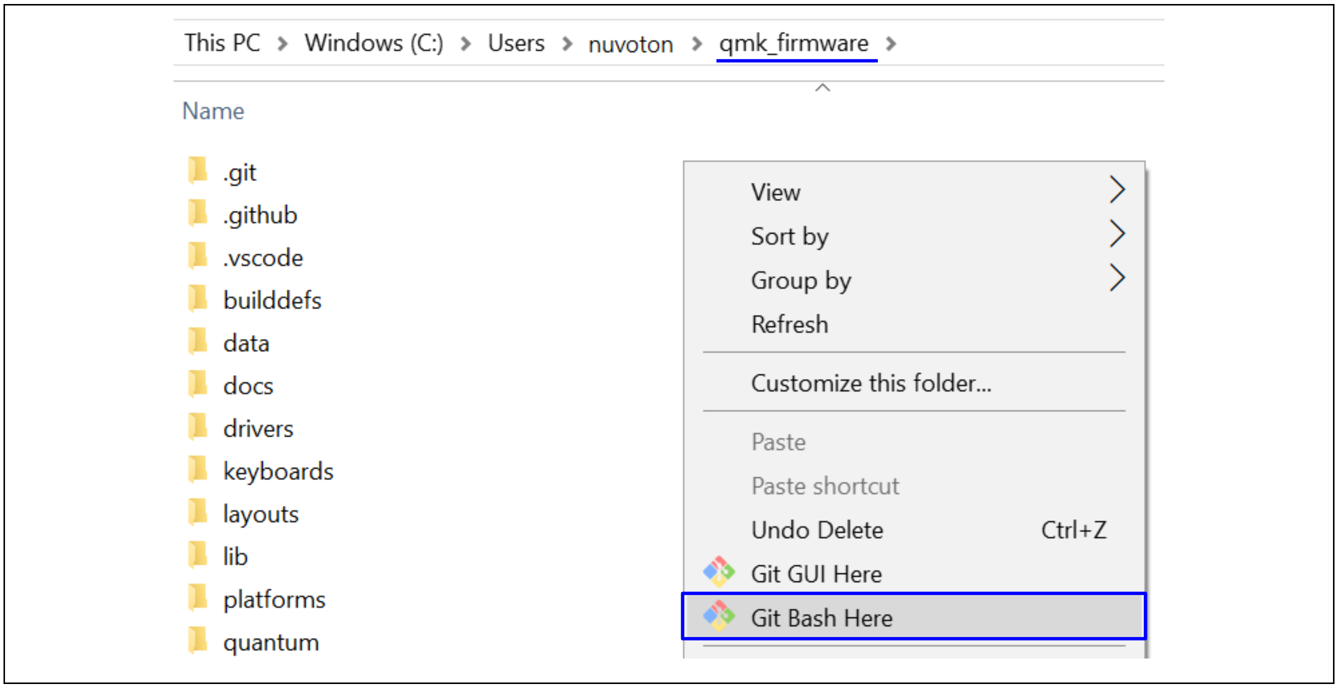


Figure 5-7 Open Git

5. In opened MINGW64 windows, run git apply patch command to add NUC029xGE related files into qmk_firmware code.

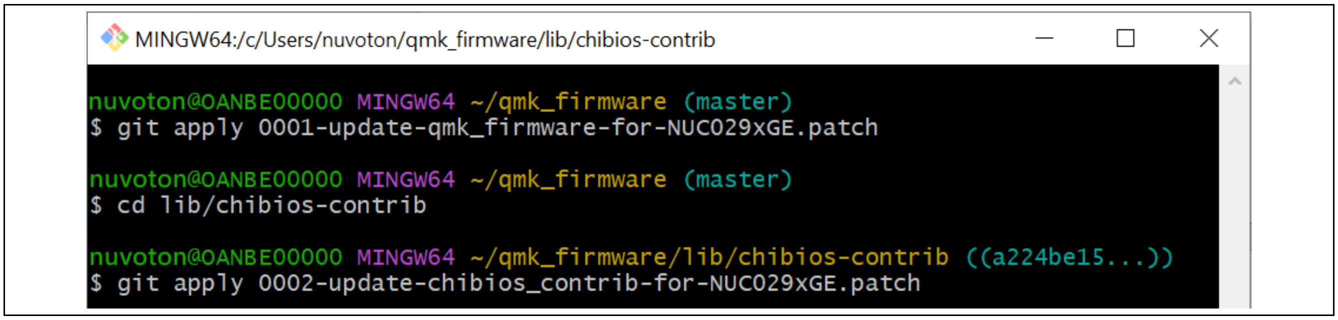
```
git apply 0001-update-qmk_firmware-for-NUC029xGE.patch
```

6. Switch working directory to qmk_firmware/lib/chibios-contrib.

```
cd lib/chibios-contrib
```

7. Run git apply patch command to add NUC029xGE related files into chibios-contrib code.

```
git apply 0002-update-chibios_contrib-for-NUC029xGE.patch
```



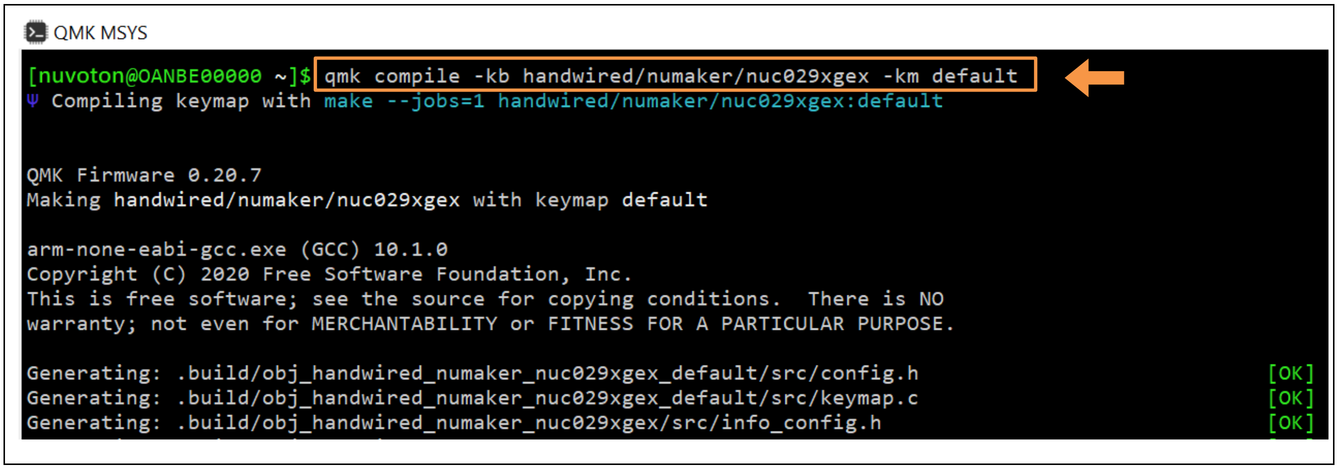
```
nuvoton@OANBE00000 MINGW64 ~/qmk_firmware (master)
$ git apply 0001-update-qmk_firmware-for-NUC029xGE.patch

nuvoton@OANBE00000 MINGW64 ~/qmk_firmware (master)
$ cd lib/chibios-contrib

nuvoton@OANBE00000 MINGW64 ~/qmk_firmware/lib/chibios-contrib ((a224be15...))
$ git apply 0002-update-chibios_contrib-for-NUC029xGE.patch
```

Figure 5-8 Add NUC029xGE Related Files into qmk_firmware and chibios-contrib Code

5.1.2 Compile Keyboard Code

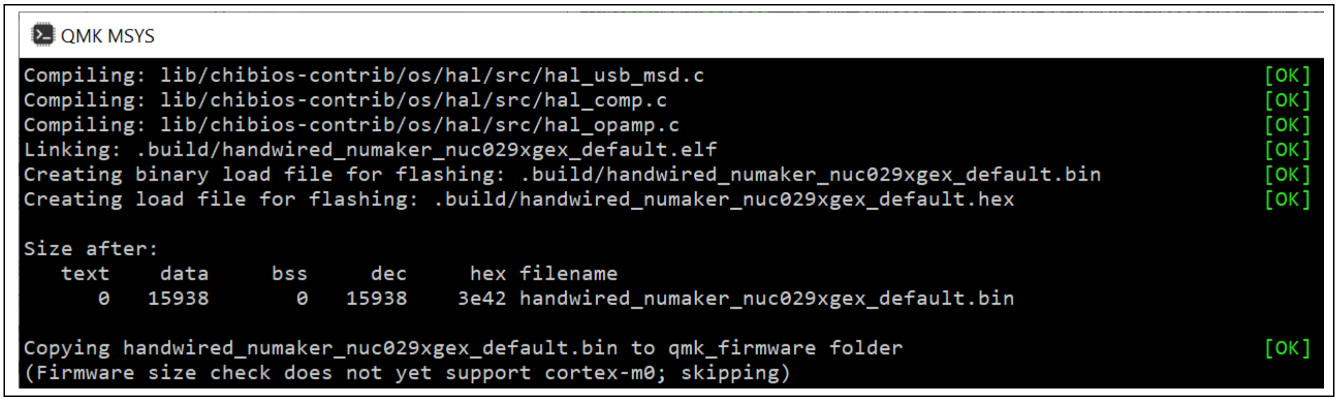
After adding NUC029xGE files, start to compile the keyboard code. In QMK MSYS, run qmk compile command to compile keyboard code, and the firmware file will be added into qmk_firmware directory after compiling process is finished.

```
qmk compile -kb handwired/numaker/nuc029xgex -km default
```



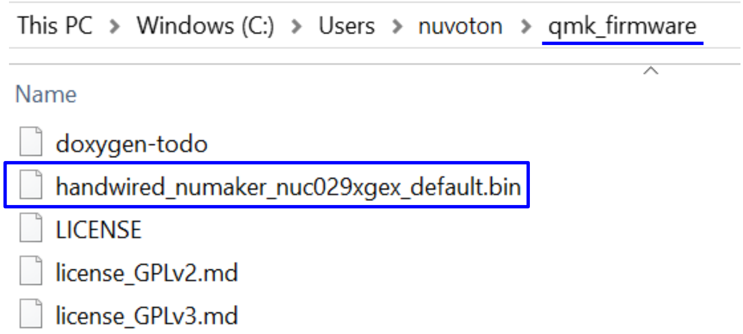
```
[nuvoton@OANBE00000 ~]$ qmk compile -kb handwired/numaker/nuc029xgex -km default
Ψ Compiling keymap with make --jobs=1 handwired/numaker/nuc029xgex:default


QMK Firmware 0.20.7
Making handwired/numaker/nuc029xgex with keymap default

arm-none-eabi-gcc.exe (GCC) 10.1.0
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Generating: .build/obj_handwired_numaker_nuc029xgex_default/src/config.h        [OK]
Generating: .build/obj_handwired_numaker_nuc029xgex_default/src/keymap.c        [OK]
Generating: .build/obj_handwired_numaker_nuc029xgex/src/info_config.h           [OK]
```

Figure 5-9 Compile Keyboard Code



```
Compiling: lib/chibios-contrib/os/hal/src/hal_usb_msd.c                                    [OK]
Compiling: lib/chibios-contrib/os/hal/src/hal_comp.c                                       [OK]
Compiling: lib/chibios-contrib/os/hal/src/hal_opamp.c                                      [OK]
Linking: .build/handwired_numaker_nuc029xgex_default.elf                                   [OK]
Creating binary load file for flashing: .build/handwired_numaker_nuc029xgex_default.bin    [OK]
Creating load file for flashing: .build/handwired_numaker_nuc029xgex_default.hex           [OK]

Size after:
   text    data     bss     dec     hex filename
      0   15938       0   15938    3e42 handwired_numaker_nuc029xgex_default.bin

Copying handwired_numaker_nuc029xgex_default.bin to qmk_firmware folder                    [OK]
(Firmware size check does not yet support cortex-m0; skipping)
```

Figure 5-10 Compile Keyboard Code Finished



This PC > Windows (C:) > Users > nuvoton > qmk_firmware

Name

doxygen-todo
handwired_numaker_nuc029xgex_default.bin
LICENSE
license_GPLv2.md
license_GPLv3.md

Figure 5-11 Keyboard Code Firmware File

5.2 Bootloader Code

5.2.1 Download Bootloader Code

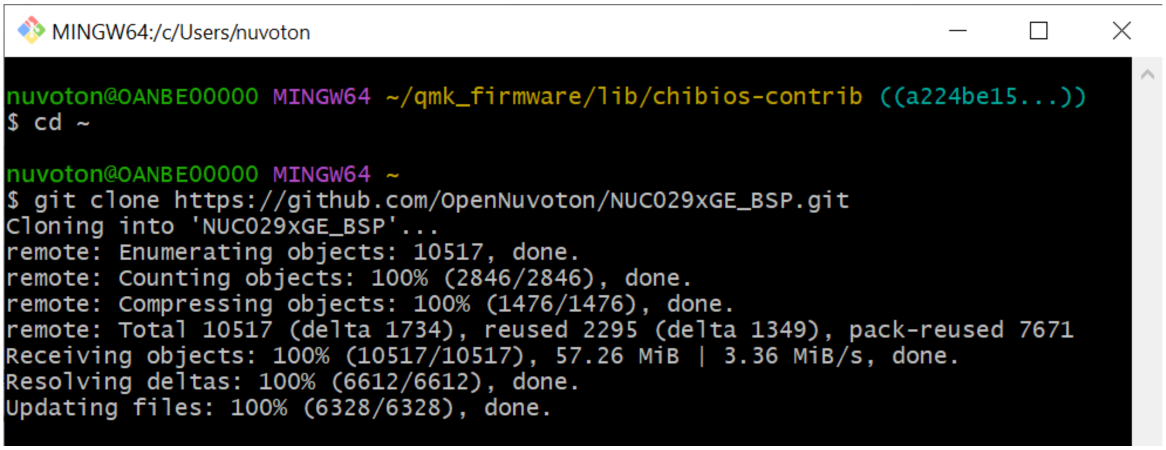
The bootloader of this example code is modified from ISP_DFU sample code in NUC029xGE BSP. Download NUC029xGE BSP from GitHub, switch code version to v3.00.004 and then add the modified files by git apply patch method.

1. In MINGW64 windows, switch back to working directory.

```
cd ~
```

2. Run git clone command to download NUC029xGE BSP from GitHub.

```
git clone https://github.com/OpenNuvoton/NUC029xGE_BSP.git
```



MINGW64:/c/Users/nuvoton

```
nuvoton@OANBE00000 MINGW64 ~/qmk_firmware/lib/chibios-contrib ((a224be15...))
$ cd ~

nuvoton@OANBE00000 MINGW64 ~
$ git clone https://github.com/OpenNuvoton/NUC029xGE_BSP.git
Cloning into 'NUC029xGE_BSP'...
remote: Enumerating objects: 10517, done.
remote: Counting objects: 100% (2846/2846), done.
remote: Compressing objects: 100% (1476/1476), done.
remote: Total 10517 (delta 1734), reused 2295 (delta 1349), pack-reused 7671
Receiving objects: 100% (10517/10517), 57.26 MiB | 3.36 MiB/s, done.
Resolving deltas: 100% (6612/6612), done.
Updating files: 100% (6328/6328), done.
```

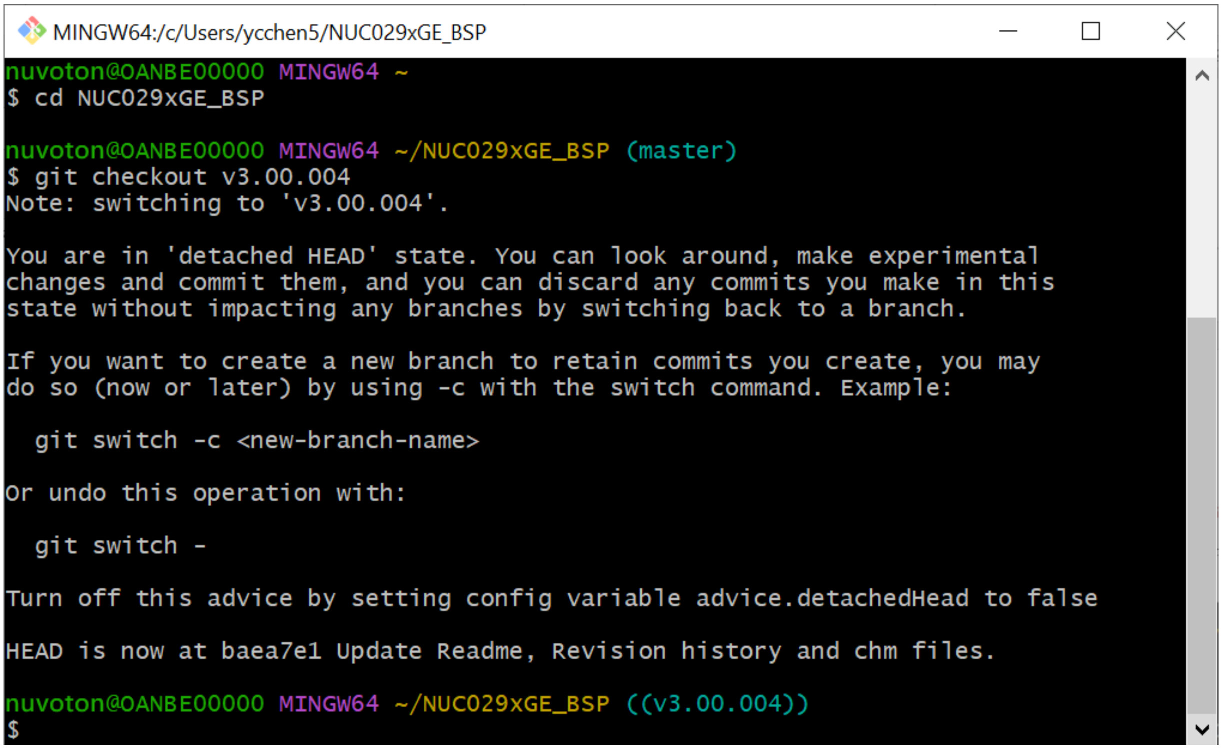
Figure 5-12 Download NUC029xGE_BSP Code

3. Switch to NUC029xGE_BSP directory.

```
cd NUC029xGE_BSP
```

4. Switch NUC029xGE_BSP version to v3.00.004.

```
git checkout v3.00.004
```



```
nuvoton@OANBE00000 MINGW64 ~
$ cd NUC029xGE_BSP

nuvoton@OANBE00000 MINGW64 ~/NUC029xGE_BSP (master)
$ git checkout v3.00.004
Note: switching to 'v3.00.004'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at baea7e1 Update Readme, Revision history and chm files.

nuvoton@OANBE00000 MINGW64 ~/NUC029xGE_BSP ((v3.00.004))
$
```

Figure 5-13 Switch NUC029xGE_BSP Version to v3.00.004

5. Put ISP_DFU git patch file of this example code under NUC029xGE_BSP directory just downloaded.

- EC_NUC029_QMK_Keyboard_V1.00
  - git_patch
    - 0003-update-ISP_DFU-for-NUC029xGE.patch

Figure 5-14 Example Code ISP_DFU Git Patch File Location

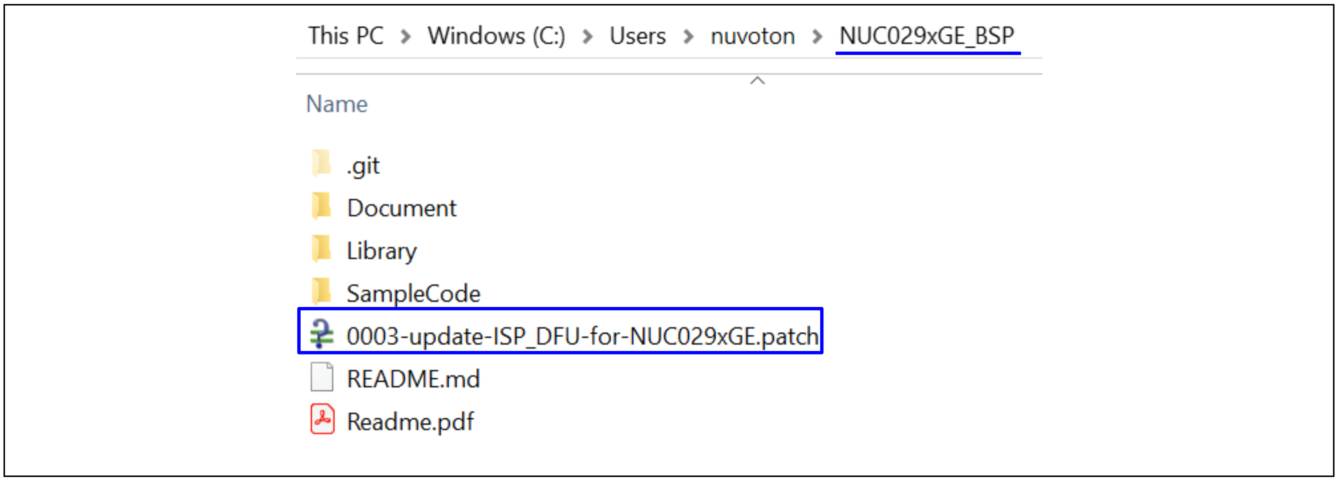
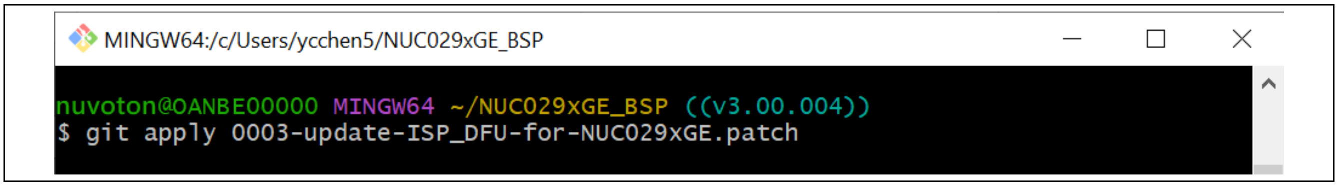

This PC > Windows (C:) > Users > nuvoton > NUC029xGE_BSP
Name
.git
Document
Library
SampleCode
0003-update-ISP_DFU-for-NUC029xGE.patch
README.md
Readme.pdf

Figure 5-15 Put ISP_DFU Git Patch File into NUC029xGE_BSP Directory

6. In MINGW64 windows, run git apply patch command to add ISP_DFU sample code modified files into NUC029xGE_BSP code.

```
git apply 0003-update-ISP_DFU-for-NUC029xGE.patch
```

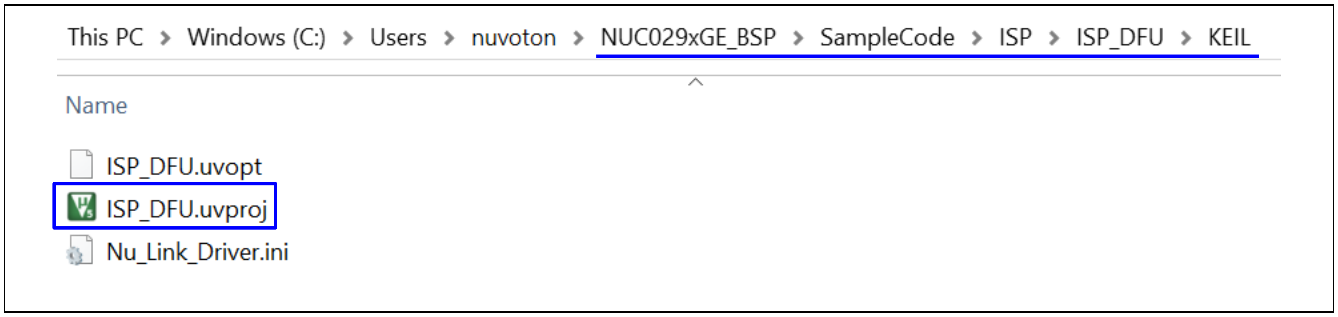


```
nuvoton@OANBE00000 MINGW64 ~/NUC029xGE_BSP ((v3.00.004))
$ git apply 0003-update-ISP_DFU-for-NUC029xGE.patch
```

Figure 5-16 Add ISP_DFU Sample Code Modified Files into NUC029xGE_BSP Code

5.2.2 Build Bootloader Code

1. Double-click on ISP_DFU sample code Keil project to open it. The project path is shown in Figure 5-17.


This PC > Windows (C:) > Users > nuvoton > NUC029xGE_BSP > SampleCode > ISP > ISP_DFU > KEIL

Name

ISP_DFU.uvopt
ISP_DFU.uvproj
Nu_Link_Driver.ini

Figure 5-17 ISP_DFU Sample Code Keil Project Path

2. In the opened Keil project, click **Rebuild** button on the top screen to build project. The ISP_DFU sample code firmware file is created into obj directory after the building process is finished.

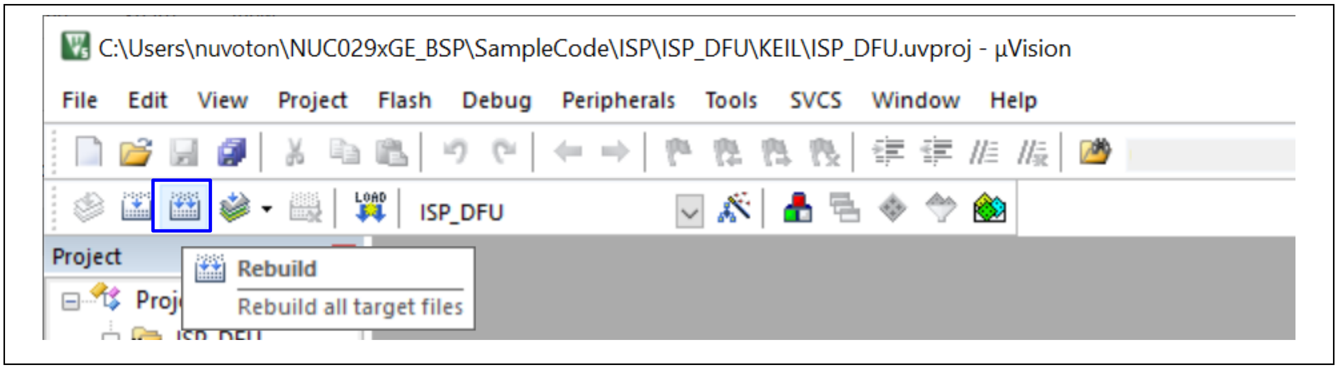
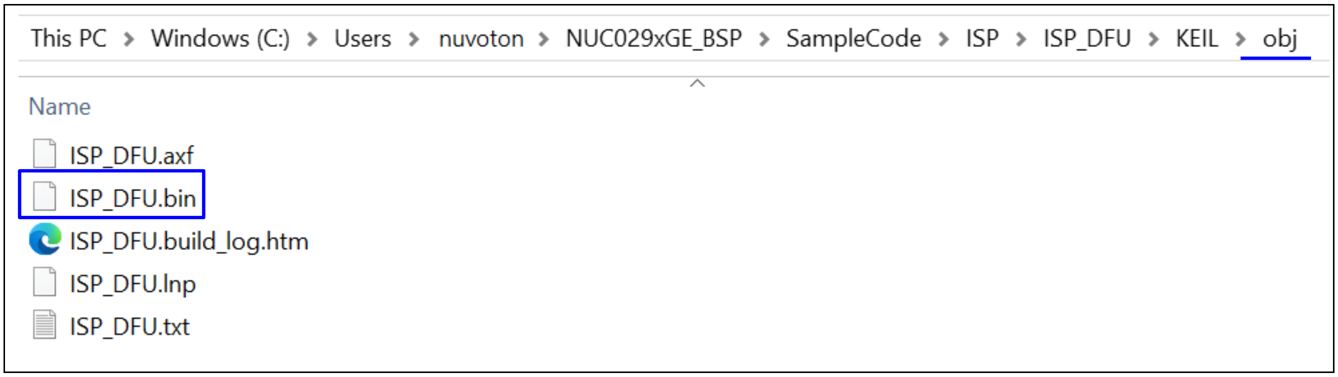

C:\Users\nuvoton\NUC029xGE_BSP\SampleCode\ISP\ISP_DFU\KEIL\ISP_DFU.uvproj - µVision

Figure 5-18 Build ISP_DFU sample code


This PC > Windows (C:) > Users > nuvoton > NUC029xGE_BSP > SampleCode > ISP > ISP_DFU > KEIL > obj

Name

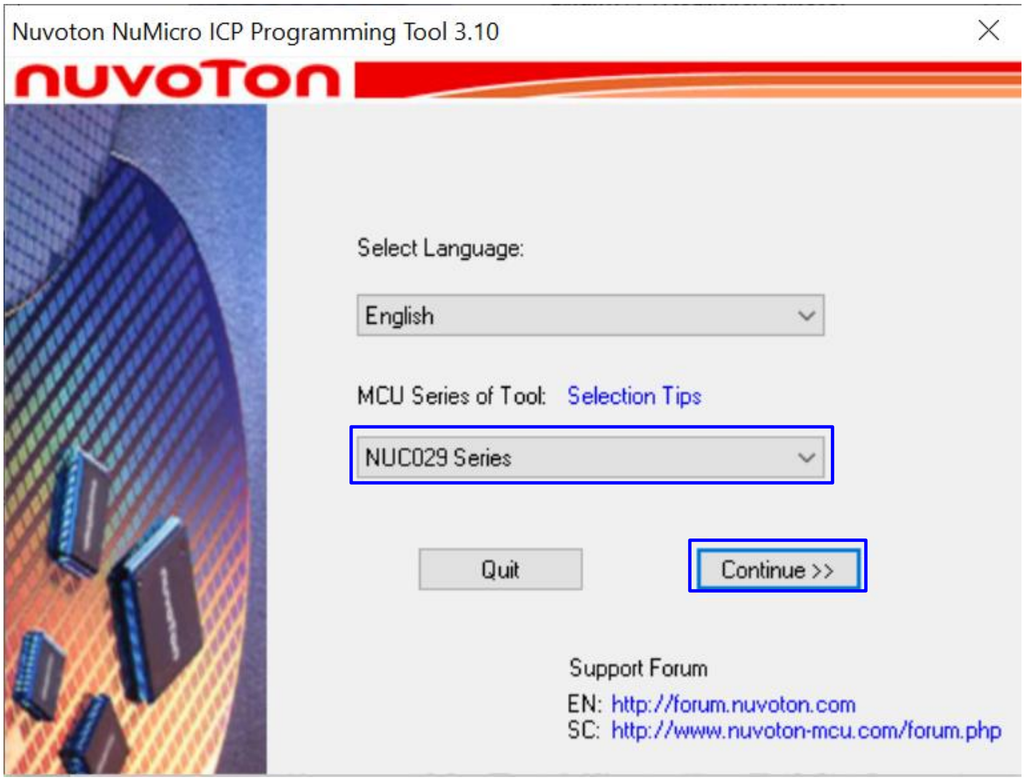
ISP_DFU.axf
ISP_DFU.bin
ISP_DFU.build_log.htm
ISP_DFU.lnp
ISP_DFU.txt

Figure 5-19 ISP_DFU Firmware File

5.3 Download Code

Next, download the keyboard and bootloader firmware on NuMaker-NUC029SGE board through NuMicro ICP Programming Tool.

1. Open NuMicro ICP Programming Tool. In **MCU Series of Tool** pull-down menu, select **NUC029 Series** and click **Continue>>** button to continue.



Select Language: English

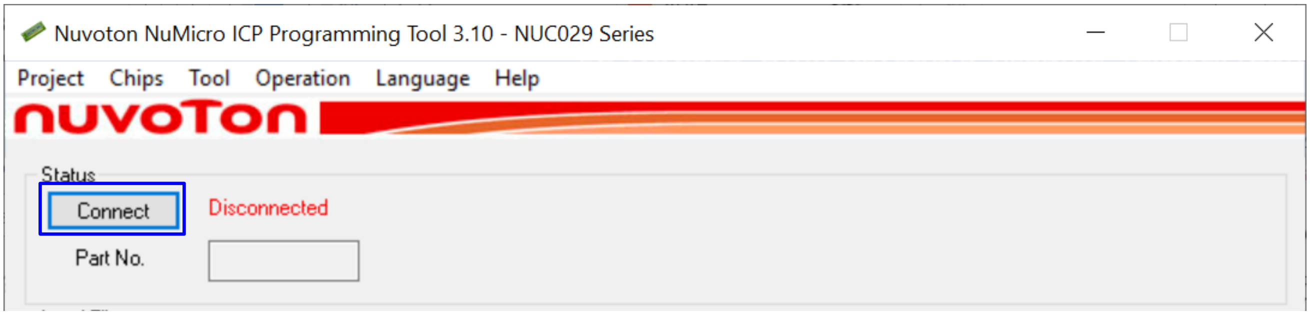
MCU Series of Tool: Selection Tips

NUC029 Series

Support Forum
EN: http://forum.nuvoton.com
SC: http://www.nuvoton-mcu.com/forum.php

Figure 5-20 NuMicro ICP Programming Tool

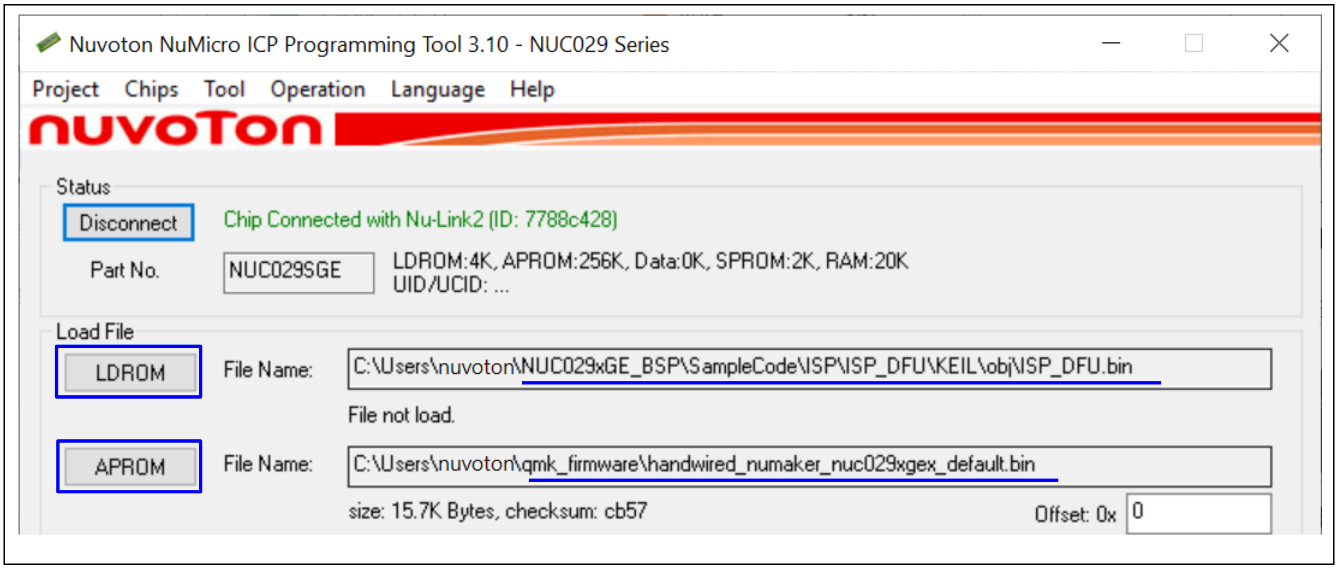
2. Connect NuMaker-NUC029SGE board Nu-Link2-Me side USB to computer. In **Status** field, click **Connect** button to connect NuMaker-NUC029SGE board.



Disconnected

Figure 5-21 Connect NuMaker-NUC029SGE Board

3. After chip is connected, in **Load File** field, select **LDROM** firmware file as bootloader code and **APROM** firmware file as keyboard code.



Chip Connected with Nu-Link2 (ID: 7788c428)

Part No. NUC029SGE

LDROM:4K, APROM:256K, Data:0K, SPROM:2K, RAM:20K
UID/UCID: ...

File Name: C:\Users\nuvoton\NUC029xGE_BSP\SampleCode\ISP\ISP_DFU\KEIL\obj\ISP_DFU.bin

File not load.

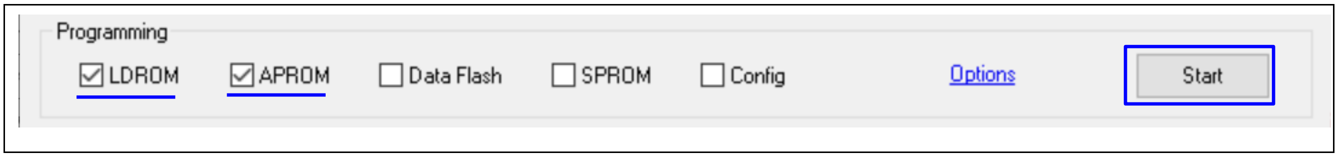
File Name: C:\Users\nuvoton\qmk_firmware\handwired_numaker_nuc029xgex_default.bin

size: 15.7K Bytes, checksum: cb57

Offset: 0x 0

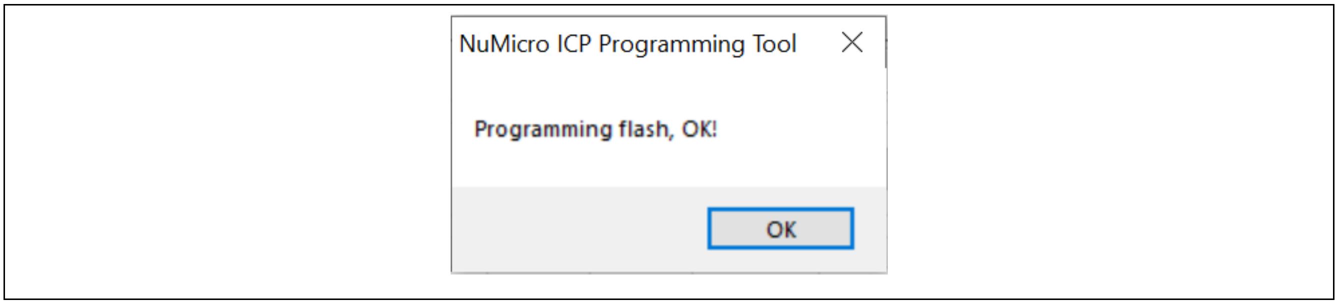
Figure 5-22 Select LDROM and APROM Firmware File

4. In the **Programming** field, select download region, includes **LDROM** and **APROM**, and then click **Start** button to start programming. Finally, show the programming OK message.



Programming: ☑ LDROM ☑ APROM ☐ Data Flash ☐ SPROM ☐ Config

Figure 5-23 Start Programming



Programming flash, OK!

Figure 5-24 Programming Flash OK

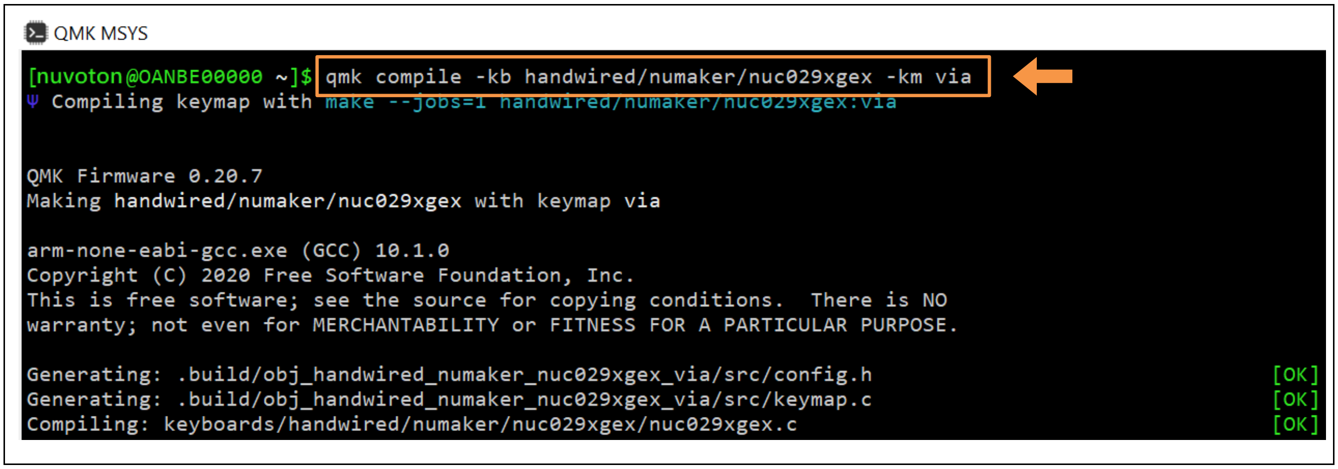
After programming is finished, connect NuMaker-NUC029xGE side USB to computer. It will be recognized as a USB HID keyboard device. Press the buttons on NuTFT_LCM board and they can be used as keys on keyboard.

5.4 Update Firmware

QMK provides QMK Toolbox tool to update firmware. The following will demonstrate how to update QMK keyboard firmware that supports VIA function by QMK Toolbox (Nuvoton Edition):

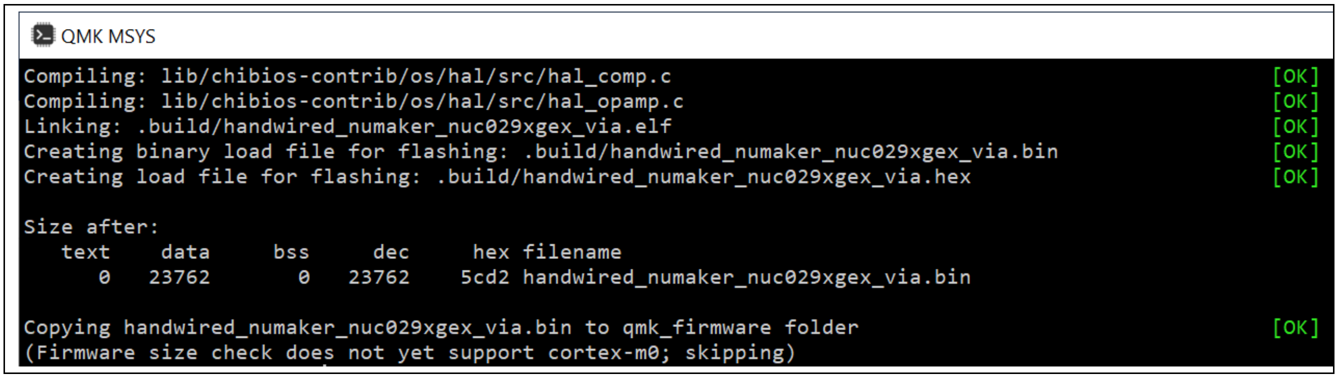
1. Open QMK MSYS, run qmk compile command to compile the QMK keyboard firmware with VIA function. The firmware file will be in qmk_firmware directory after compiling process is finished.

```
qmk compile -kb handwired/numaker/nuc029xgex -km via
```



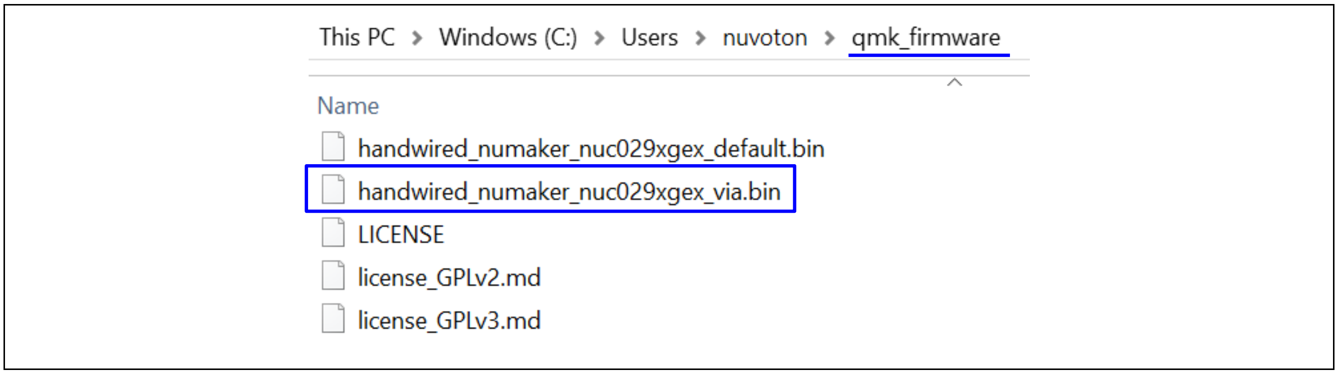
```
[nuvoton@OANBE00000 ~]$ qmk compile -kb handwired/numaker/nuc029xgex -km via
Ψ Compiling keymap with make --jobs=1 handwired/numaker/nuc029xgex:via


QMK Firmware 0.20.7
Making handwired/numaker/nuc029xgex with keymap via

arm-none-eabi-gcc.exe (GCC) 10.1.0
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Generating: .build/obj_handwired_numaker_nuc029xgex_via/src/config.h                 [OK]
Generating: .build/obj_handwired_numaker_nuc029xgex_via/src/keymap.c                 [OK]
Compiling: keyboards/handwired/numaker/nuc029xgex/nuc029xgex.c                       [OK]
```

Figure 5-25 Compile Keyboard Firmware with VIA Function



```
Compiling: lib/chibios-contrib/os/hal/src/hal_comp.c                                  [OK]
Compiling: lib/chibios-contrib/os/hal/src/hal_opamp.c                                 [OK]
Linking: .build/handwired_numaker_nuc029xgex_via.elf                                  [OK]
Creating binary load file for flashing: .build/handwired_numaker_nuc029xgex_via.bin   [OK]
Creating load file for flashing: .build/handwired_numaker_nuc029xgex_via.hex          [OK]

Size after:
   text    data     bss     dec     hex filename
      0   23762       0   23762    5cd2 handwired_numaker_nuc029xgex_via.bin

Copying handwired_numaker_nuc029xgex_via.bin to qmk_firmware folder                   [OK]
(Firmware size check does not yet support cortex-m0; skipping)
```

Figure 5-26 Compile Keyboard Firmware with VIA Function Finished



handwired_numaker_nuc029xgex_default.bin
handwired_numaker_nuc029xgex_via.bin
LICENSE
license_GPLv2.md
license_GPLv3.md

Figure 5-27 Keyboard Firmware with VIA Function

2. To switch to bootloader mode, connect NuMaker-NUC029SGE side USB to computer. Hold on SW1 button on NuMaker-NUC029SGE board and press **RESET** button.

3. After entering bootloader, it will be recognized as a Nu_DFU device, and then user can release SW1 button. If Nu_DFU device cannot be recognized, user needs to install Nu_DFU device driver. The file path is shown in Figure 5-29.

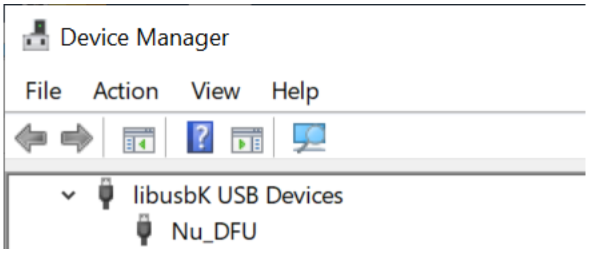


Figure 5-28 Nu_DFU Device

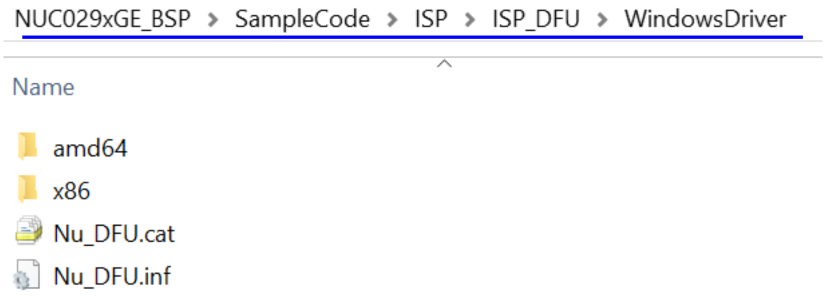
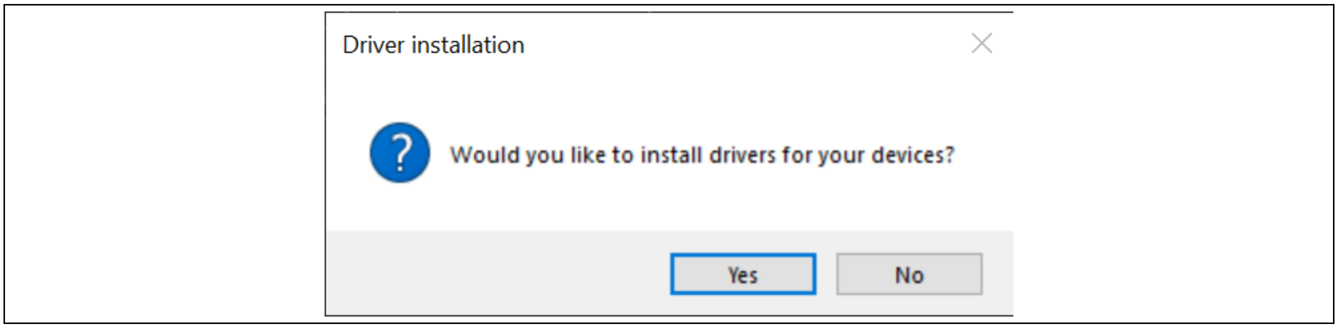


Figure 5-29 Nu_DFU Device Driver File Path

4. Open QMK Toolbox (Nuvoton Edition) in this example code. The file path is shown in Figure 5-30. If a message about asking to install drivers for devices is shown after opening QMK Toolbox (Nuvoton Edition), as shown in Figure 5-31, note that the driver that is asked to be installed at this time is applicable to other QMK supported devices, but not for the Nu_DFU device. The driver of Nu_DFU device needs to be installed manually, but user can decide whether to install other device drivers here.

🗁 EC_NUC029_QMK_Keyboard_V1.00
    🗁 qmk_tool
        🗎 qmk_toolbox - nuvoton_edition.exe

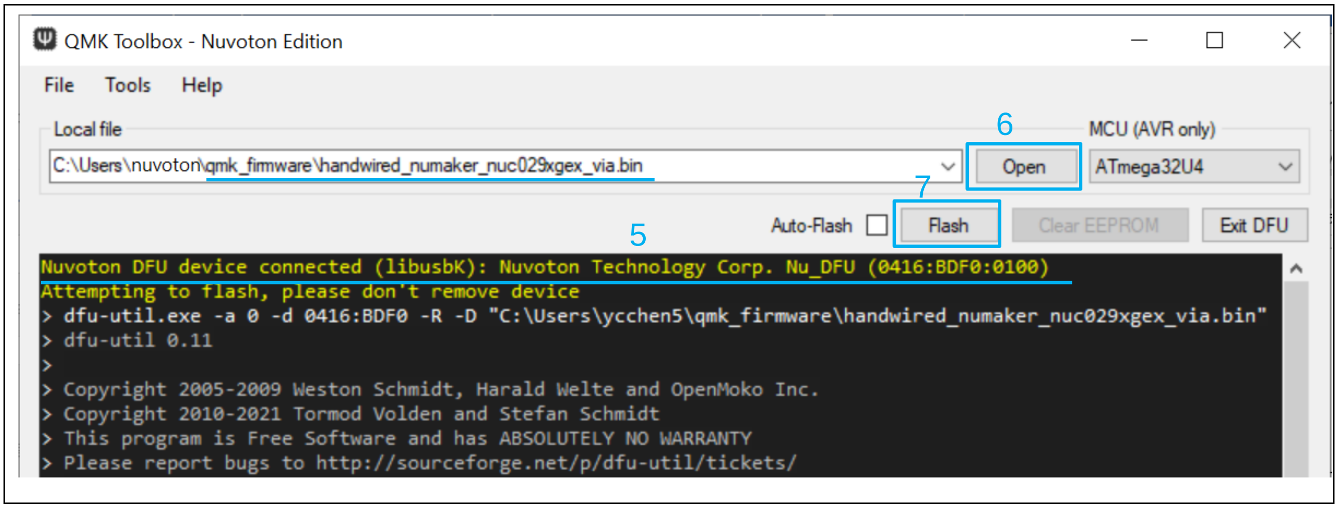
Figure 5-30 QMK Toolbox (Nuvoton Edition) File Path



Would you like to install drivers for your devices?

Figure 5-31 Ask to Install Drivers for Devices (But Not for Nu_DFU Device)

5. QMK Toolbox (Nuvoton Edition) shows that the Nuvoton DFU Device is detected.
6. Click **Open** button to select the firmware file to be updated.
7. Click **Flash** button to start updating firmware, and show a completed message when update firmware process is finished.

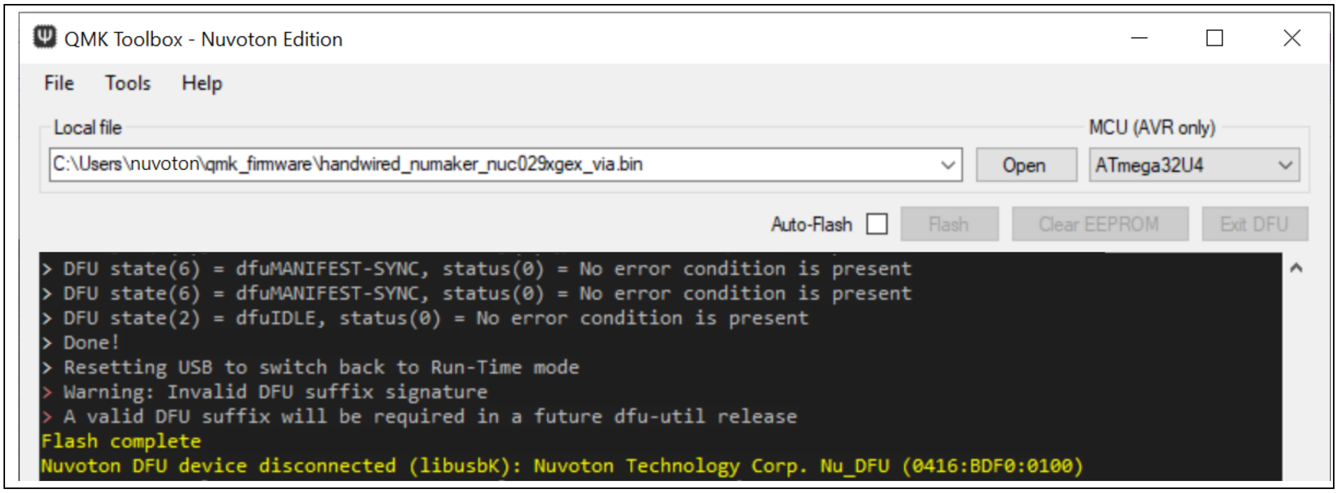


Local file: C:\Users\nuvoton\qmk_firmware\handwired_numaker_nuc029xgex_via.bin

MCU (AVR only): ATmega32U4

```
Nuvoton DFU device connected (libusbK): Nuvoton Technology Corp. Nu_DFU (0416:BDF0:0100)
Attempting to flash, please don't remove device
> dfu-util.exe -a 0 -d 0416:BDF0 -R -D "C:\Users\ycchen5\qmk_firmware\handwired_numaker_nuc029xgex_via.bin"
> dfu-util 0.11
>
> Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
> Copyright 2010-2021 Tormod Volden and Stefan Schmidt
> This program is Free Software and has ABSOLUTELY NO WARRANTY
> Please report bugs to http://sourceforge.net/p/dfu-util/tickets/
```

Figure 5-32 Update Firmware by QMK Toolbox (Nuvoton Edition)



Local file: C:\Users\nuvoton\qmk_firmware\handwired_numaker_nuc029xgex_via.bin

MCU (AVR only): ATmega32U4

```
> DFU state(6) = dfuMANIFEST-SYNC, status(0) = No error condition is present
> DFU state(6) = dfuMANIFEST-SYNC, status(0) = No error condition is present
> DFU state(2) = dfuIDLE, status(0) = No error condition is present
> Done!
> Resetting USB to switch back to Run-Time mode
> Warning: Invalid DFU suffix signature
> A valid DFU suffix will be required in a future dfu-util release
Flash complete
Nuvoton DFU device disconnected (libusbK): Nuvoton Technology Corp. Nu_DFU (0416:BDF0:0100)
```

Figure 5-33 Update Firmware by QMK Toolbox (Nuvoton Edition) Finished

5.5 VIA

VIA is a tool that can modify keymap directly without compile and update firmware again. It has to be used with the QMK keyboard firmware that supports VIA function.

1. Open VIA Web App, select **SETTINGS** page on the top menu.

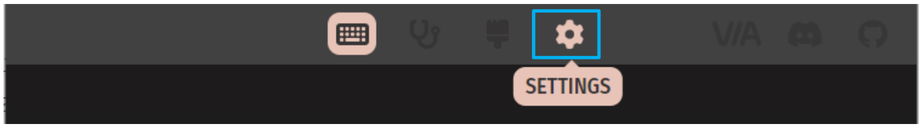
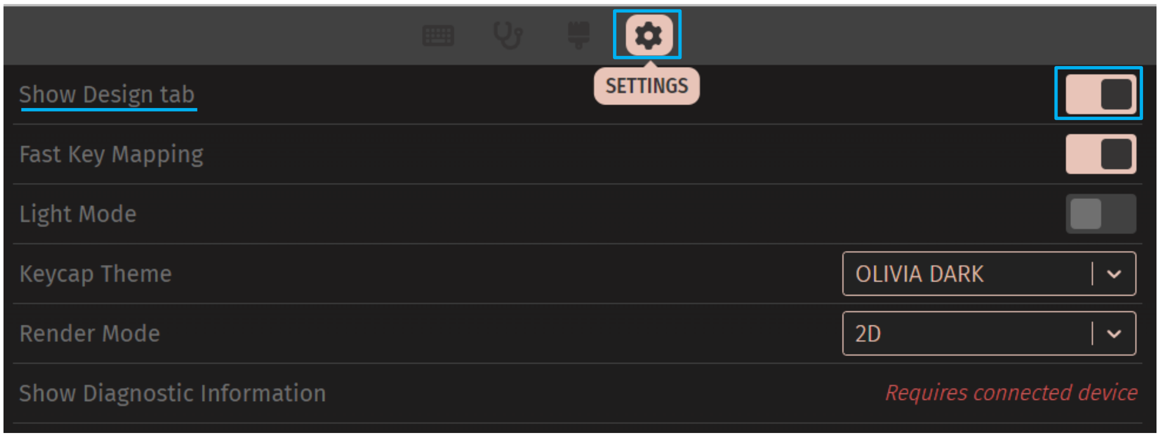


Figure 5-34 VIA Web App

2. In **SETTINGS** page as shown in Figure 5-35, enable **Show Design tab** function. An additional **DESIGN** page will be shown on the top menu.



Show Design tab
Fast Key Mapping
Light Mode
Keycap Theme OLIVIA DARK
Render Mode 2D
Show Diagnostic Information *Requires connected device*

Figure 5-35 VIA Tool SETTINGS Page

3. In **DESIGN** page as shown in Figure 5-36, disable **Use V2 definitions (deprecated)** function.

4. Click **Load** button of **Load Draft Definition** function and select the NUC029xGE JSON file for VIA tool. The file path is shown in Figure 5-37.

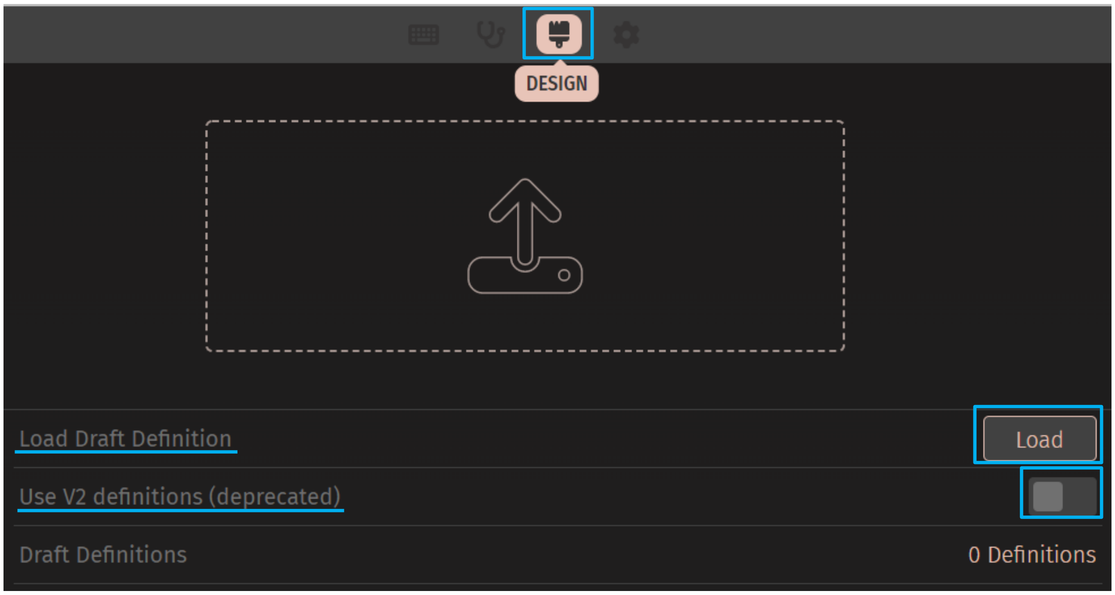
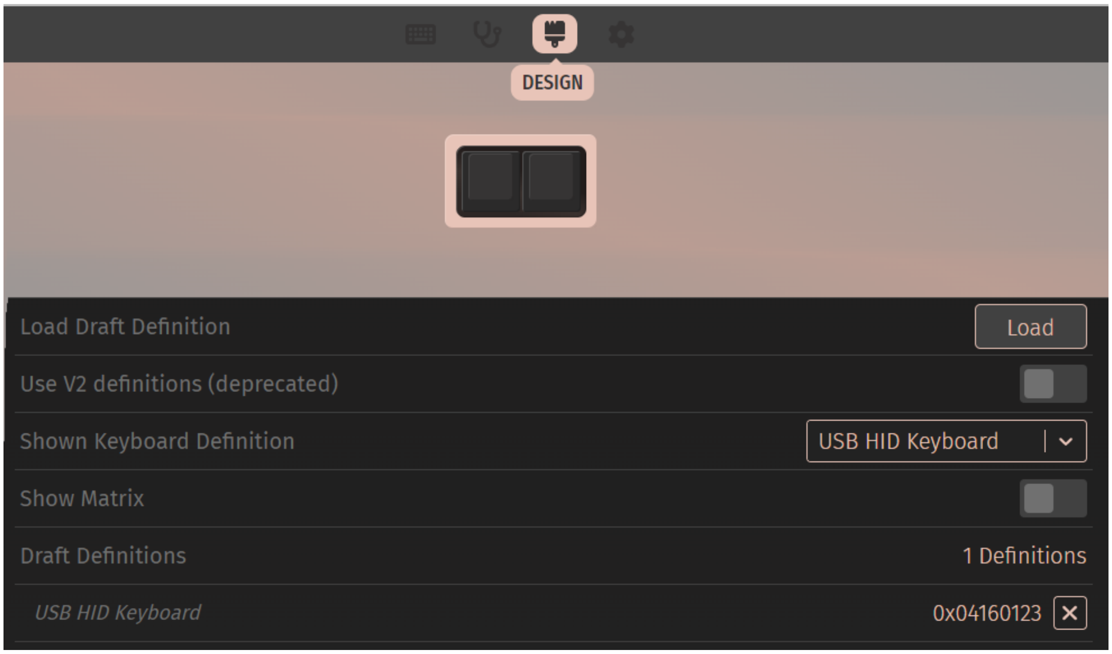


Figure 5-36 VIA Tool DESIGN Page

- 🗁 EC_NUC029_QMK_Keyboard_V1.00
  - 🗁 via_json
    - 🗎 NUC029xGE_QMK_VIA_V3.json

Figure 5-37 NUC029xGE JSON File for VIA Tool File Path

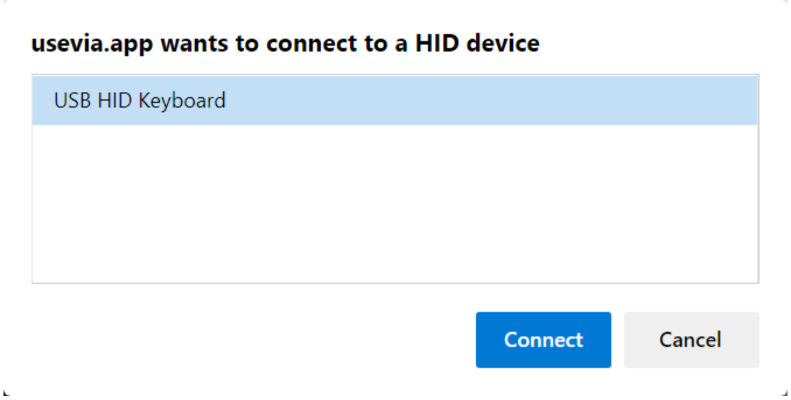
5. After adding the JSON file, there are two keys in NUC029xGE design as shown in **DESIGN** page.



Load Draft Definition
Use V2 definitions (deprecated)
Shown Keyboard Definition: USB HID Keyboard
Show Matrix
Draft Definitions: 1 Definitions
USB HID Keyboard 0x04160123

Figure 5-38 NUC029xGE Has Two Keys in Design

6. If a HID device that supports VIA tool is recognized, a window will pop up to ask to connect to the device. Select the HID device and click **Connect**.


usevia.app wants to connect to a HID device

USB HID Keyboard

Figure 5-39 Connect HID Device

7. If the connection window does not pop up, user can try to connect HID device manually by click **Authorize device +** button in **CONFIGURE** page.

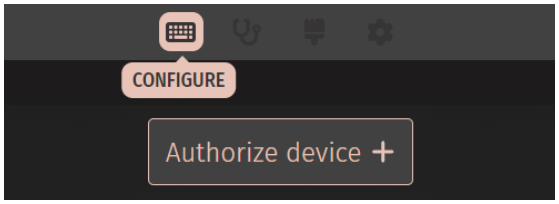


Figure 5-40 Click **Authorize device +** Button to Connect HID Device Manually

8. After the HID device is connected, there are two keys in design, **Esc** and **1**, as shown in **CONFIGURE** page.

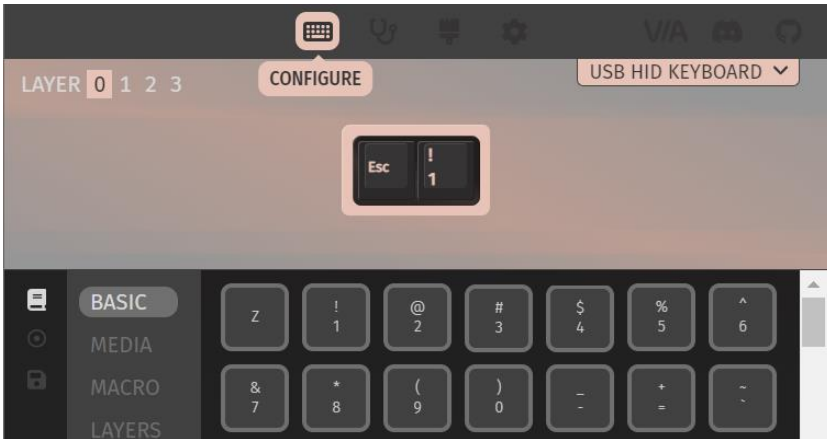


Figure 5-41 Key **Esc** and **1**

9. To modify the keymap, first, click the key to be modified such as key “**1**”. Key “**1**” is flashing when it is selected. Then click the target key you want to modify listed below, such as key “**4**”, as shown in Figure 5-42.

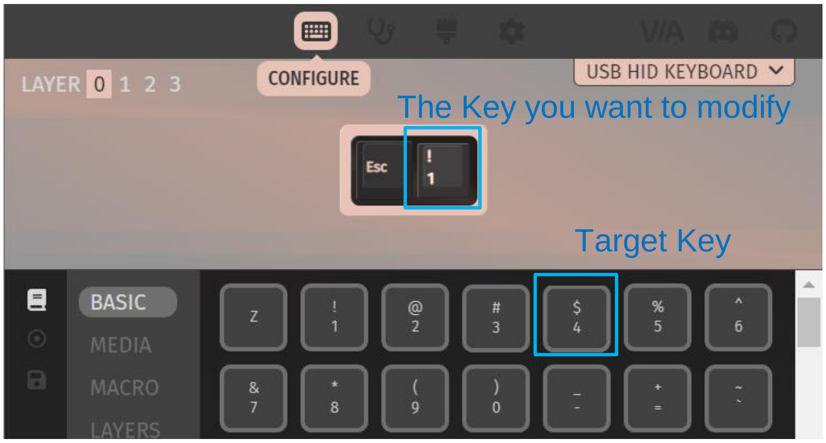


Figure 5-42 Modify Keymap

10. Then, it can be seen on the screen that the second key has changed from original “**1**” to “**4**”, as shown in Figure 5-43.

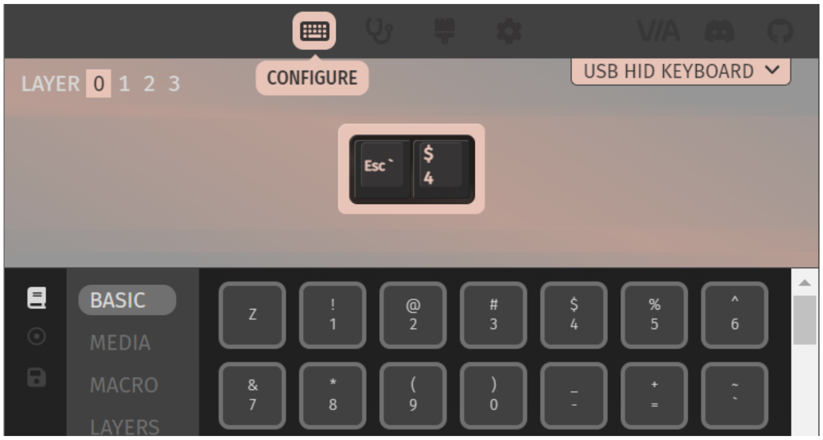


Figure 5-43 Key **Esc** and **4**

11. The VIA tool can be also used to test keyboard function. User can test the key result before and after modification directly in **KEY TESTER** page on the top menu. You can see the two keys changed to **Esc** and **4**, as shown in Figure 5-44.

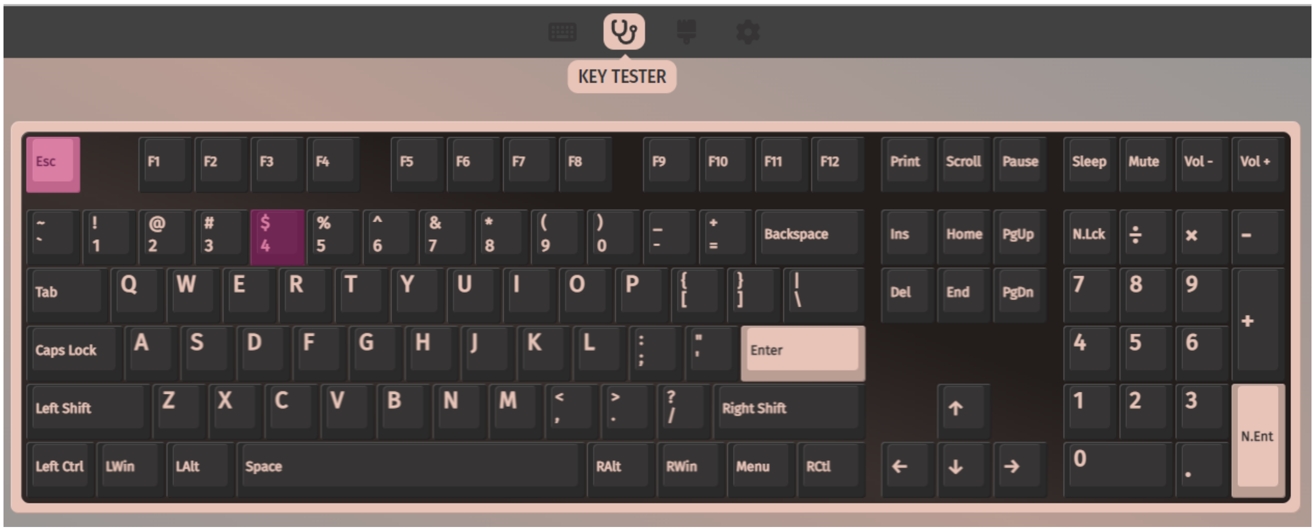


Figure 5-44 VIA Tool Key Test Result

6. Revision History

Date	Revision	Description
2023.05.23	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.