QMK键盘

NuMicro® 32位系列微控制器范例代码介绍

文件信息

应用简述	本范例代码使用开放的 QMK 键盘韧体原始码在 NUC029 实作键盘
BSP 版本	NUC029xGE_Series_BSP_CMSIS_V3.00.004
开发平台	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. 概述

QMK(Quantum Mechanical Keyboard)是一个社区维护用于开发计算机输入设备的开源软件，像是键盘、鼠标、MIDI 等设备。市售键盘可以直接使用 QMK 韧体作为产品，或支持更新为 QMK 韧体，用户就可以自行修改功能，制作专属个人客制化的键盘。除了开放的原始码，QMK 也提供其他协助程序开发的工具，例如：QMK MSYS、QMK Configurator，QMK Toolbox 等。本范例代码将使用 QMK 键盘韧体以及搭配 QMK 开发工具，在 NUC029xGE 上实作一个仿真的键盘。

1.1 原理

QMK 在 NUC029xGE 上实作仿真的键盘，包括一个键盘韧体本身和一个更新韧体的流程，键盘韧体放在 APROM，更新韧体由 bootloader 负责，放在 LDROM。如图 1-1 所示本范例代码在 NUC029xGE 的内存映像。

系统开机后，主要执行 APROM 键盘程序，在计算机上识别为一个 USB HID 键盘设备。本范例代码利用 NuTFT_LCM 板子上的两个按钮 SW1 和 SW2 仿真键盘的两个按键，SW1 为按键 **Esc**，SW2 为按键 **1**。

韧体更新由 LDROM bootloader 负责，进入 bootloader 的方式为：按住按键 **Esc** 后，再按一下 NuMaker-NUC029SGE 板子上的 RESET 按钮。重跑 APROM 程序，检查按键 **Esc** 如果有被按下，则切换到 LDROM 执行 bootloader 程序，在计算机上识别为一个 USB DFU 设备，之后就可以透过 DFU 协议更新韧体。

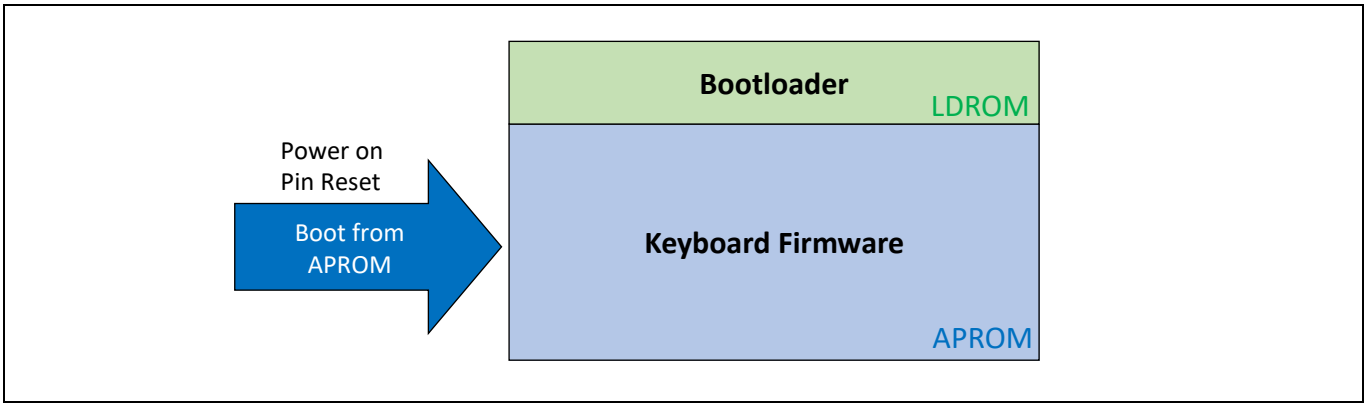

图 1-1 NUC029xGE 内存映像

1.2 执行结果

透过 QMK Configurator 提供的键盘测试工具来测试键盘功能，进入[键盘测试网页](#)，按下 NuTFT_LCM 板子上的两个按钮，就可以看到结果显示按键 **Esc** 和 **1** 被按下，如图 1-2 所示 QMK Configurator 键盘测试结果。

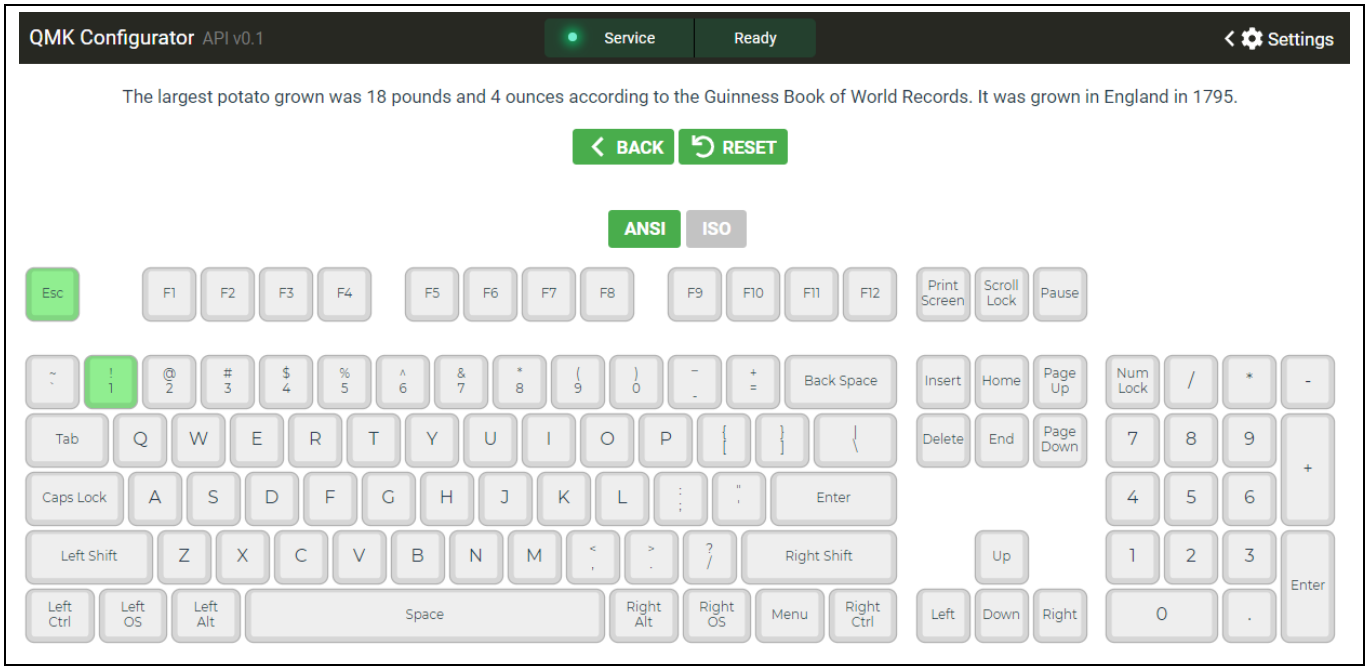

图 1-2 QMK Configurator 键盘测试结果

2. 代码介绍

2.1 主要函数

初始化设定：

- platform_setup()设定和开发平台 NUC029xGE 相关的硬件，包括系统时钟、GPIO、USB 和 systick。
- protocol_setup()设定和 USB 相关的功能。
- keyboard_setup()设定和键盘相关的功能，开启 VIA 功能时需要设定 flash。
- protocol_init()初始化和启用 USB、VIA 和 bootmagic 功能。

主要循环：

- protocol_task()负责键盘矩阵扫描和 USB 服务。
- housekeeping_task()负责其他需要周期的被执行的程序，例如：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 按键

本范例代码使用 NuTFT_LCM 板子上的两个按钮仿真键盘的两个按键，SW1 按钮连接至 NuMaker-NUC029SGE 板子上的 PC.4，SW2 按钮连接至 PC.5。在 config.h 定义 DIRECT_PIN 值为 {{C4, C5}}，在 nuc029xgex.h 定义 LAYOUT 包括两个元素 k00 与 k01，在 keymap.json 定义 LAYOUT 的两个元素各代表按键码[KC_ESCAPE]和[KC_1]。按键码定义于 keycodes.h，使用者也可以自行修改成其他想要的按键码。

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

VIA 功能可以用来修改按键定义，如果要使用 VIA 功能，在 rules.mk 设定 VIA_ENABLE 值为 yes。VIA 需要将按键定义纪录在非易失性存储器，像是 EEPROM 或 embedded flash，在 rules.mk 中设定本范例代码使用 wear leveling 方式记录按键定义于 embedded flash。在 config.h 设定 BACKING_STORE_WRITE_SIZE 是一次写入 flash 的数据为 4 bytes，定义 NUC029xGE 的 flash 大小为 256KB，允许 APROM 存取。

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 提供切换到 bootloader 模式的功能，在 rules.mk 设定 BOOTMAGIC_ENABLE 值为 yes。键盘接上 USB 后，QMK 程序会侦测指定的按键有没有被按下，默认指定的按键为位置(0,0)的[ESC]，如果[ESC]有被按下，则清除 EEPROM 数据并切换到 bootloader 模式，切换的程序在 nuc029xgex.c 的 bootloader_jump()里实现。

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

在 LDROM bootloader 程序里检查如果 **Esc** 按键有被按下，表示使用者指定要切换到 bootloader 模式，则留在 LDROM 等待韧体更新，如果 **Esc** 按键没有被按下，则执行 APROM 键盘程序。

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. 软件与硬件需求

3.1 软件需求

- GitHub Repository
  - qmk/qmk_firmware
  - OpenNuvoton/NUC029xGE_BSP v3.00.004
- 开发工具
  - 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 硬件需求

- 电路组件
  - NuMaker-NUC029SGE V1.0
  - NuTFT_LCM V1.4
- 示意图
  - 将 NuTFT_LCM 上板和 NuMaker-NUC029SGE 下板相接
  - 将 NuMaker-NUC029SGE 端的 USB 连接到计算机

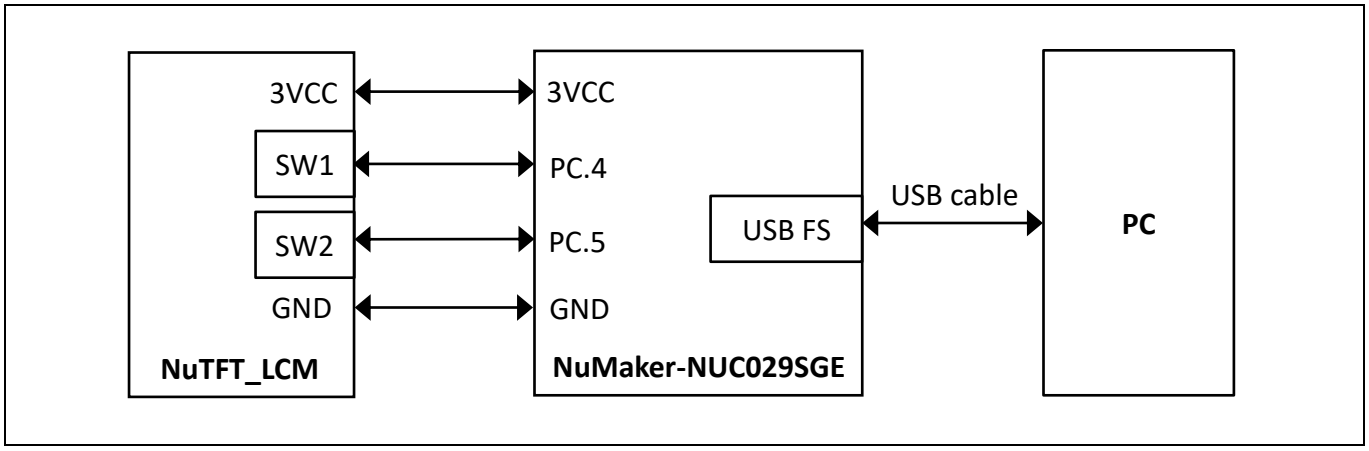

图 3-1 NuMaker-NUC029SGE 接线示意图

4. 目录信息

- 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

图 4-1 目录信息

5. 范例程序执行

本章节将介绍：如何下载编译键盘和 bootloader 程序，编译好的程序如何烧录到板子上，接下来更新韧体的流程，最后介绍 VIA 工具的使用。

5.1 键盘程序

5.1.1 下载键盘程序

本范例代码修改自 QMK 键盘韧体以及使用 QMK MSYS 为编译程序的工具，首先建立开发环境和下载键盘程序：

1. 安装 QMK MSYS。
2. 开启 QMK MSYS 执行 qmk setup 命令，检查开发环境并从 GitHub 下载 qmk_firmware 程序。

```
qmk setup
```

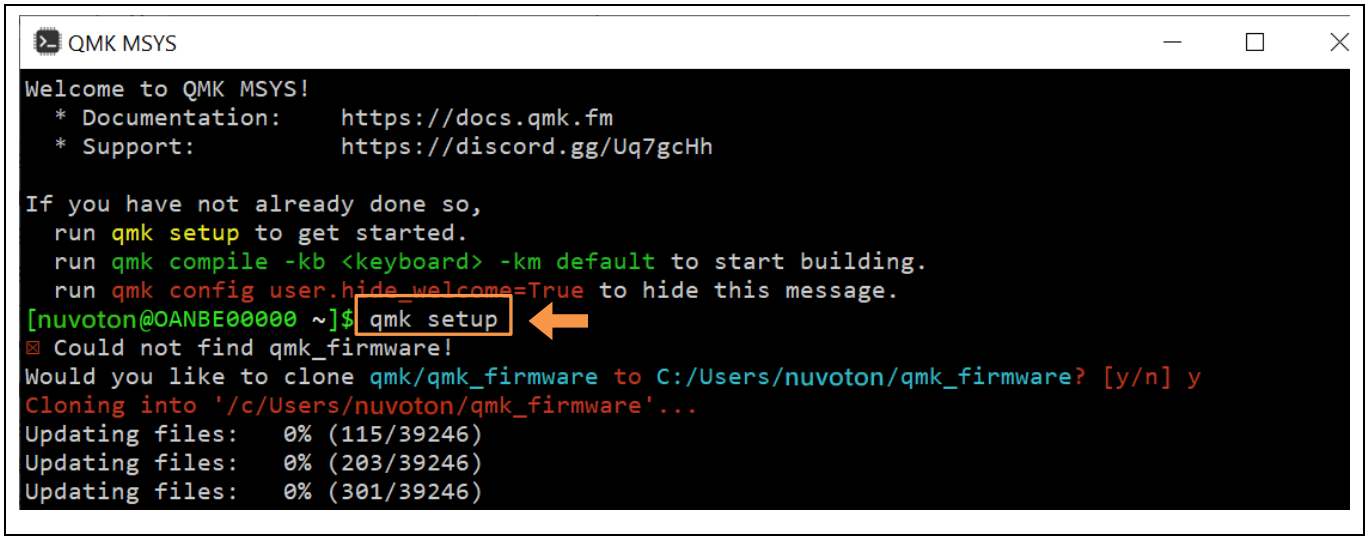

图 5-1 执行 qmk setup 命令

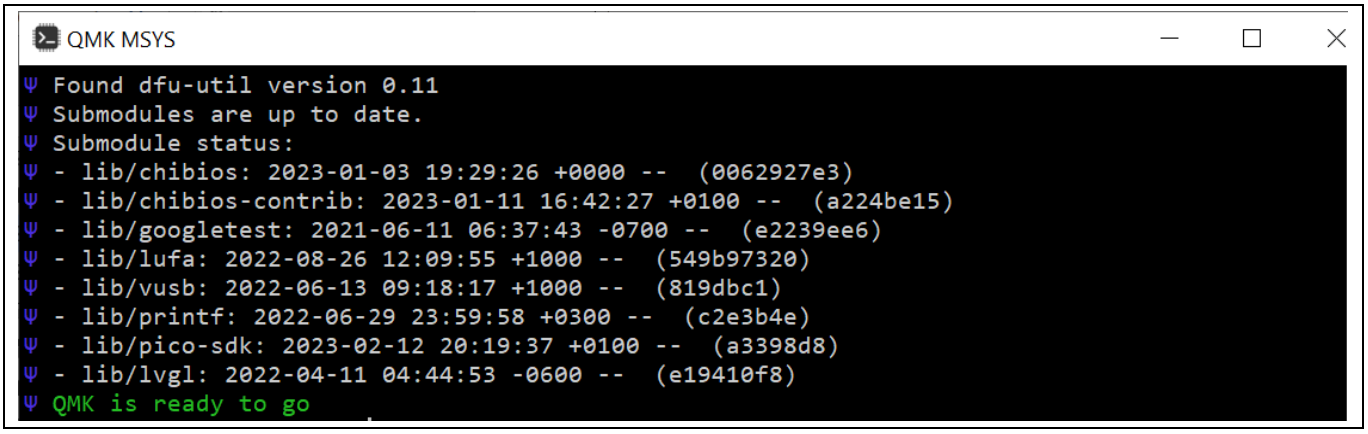

图 5-2 执行 qmk setup 命令完成

接下来透过 git apply patch 的方式加入 NUC029xGE 的相关档案到 qmk_firmware 程序：

1. 安装 Git。
2. 将本范例代码 qmk_firmware git 补丁放在刚才下载的程序 qmk_firmware 目录底下。
3. 将本范例代码 chibios-contrib git 补丁放在刚才下载的程序 qmk_firmware\lib\chibios-contrib 目录底下。

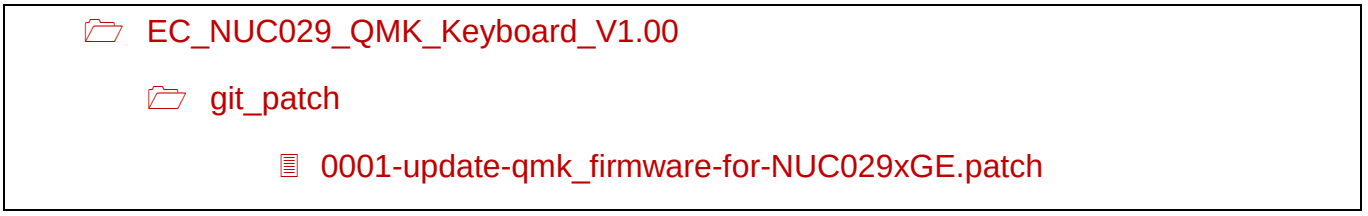

图 5-3 本范例代码 qmk_firmware git 补丁档案位置

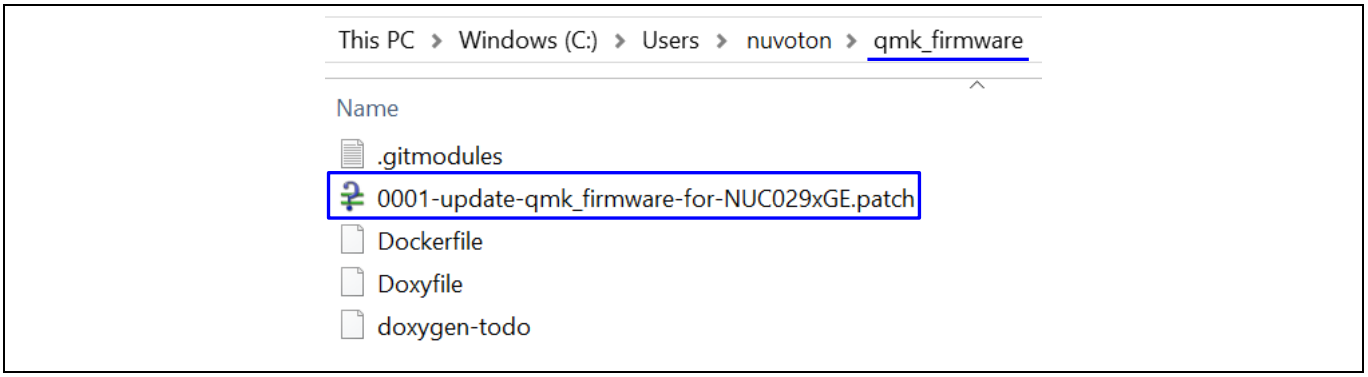

图 5-4 将 qmk_firmware git 补丁放在 qmk_firmware 目录底下

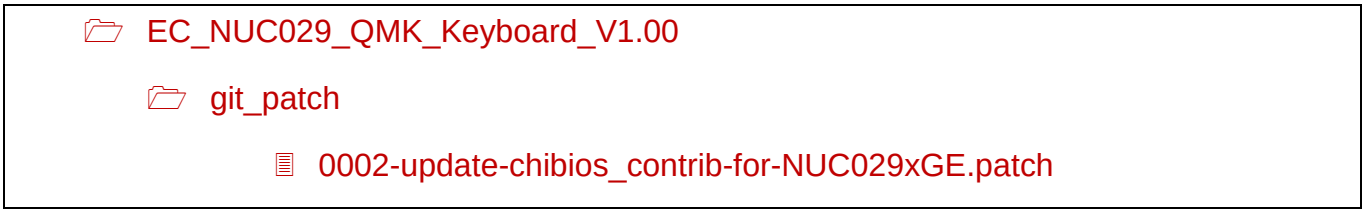

图 5-5 本范例代码 chibios-contrib git 补丁档案位置

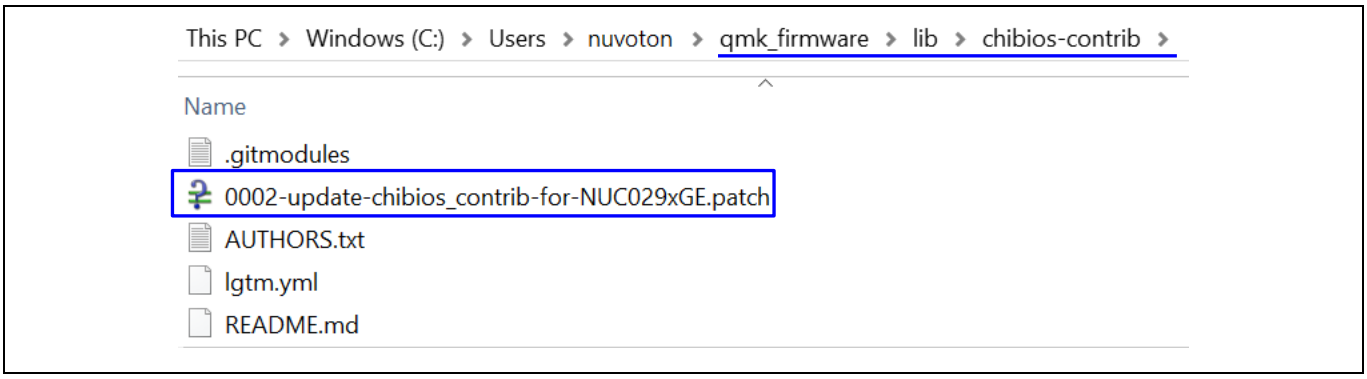

图 5-6 将 chibios-contrib git 补丁放在 qmk_firmware\lib\chibios-contrib 目录底下

4. 在 qmk_firmware 目录底下，右键单击，选择 [Git Bash Here]，开启 Git。

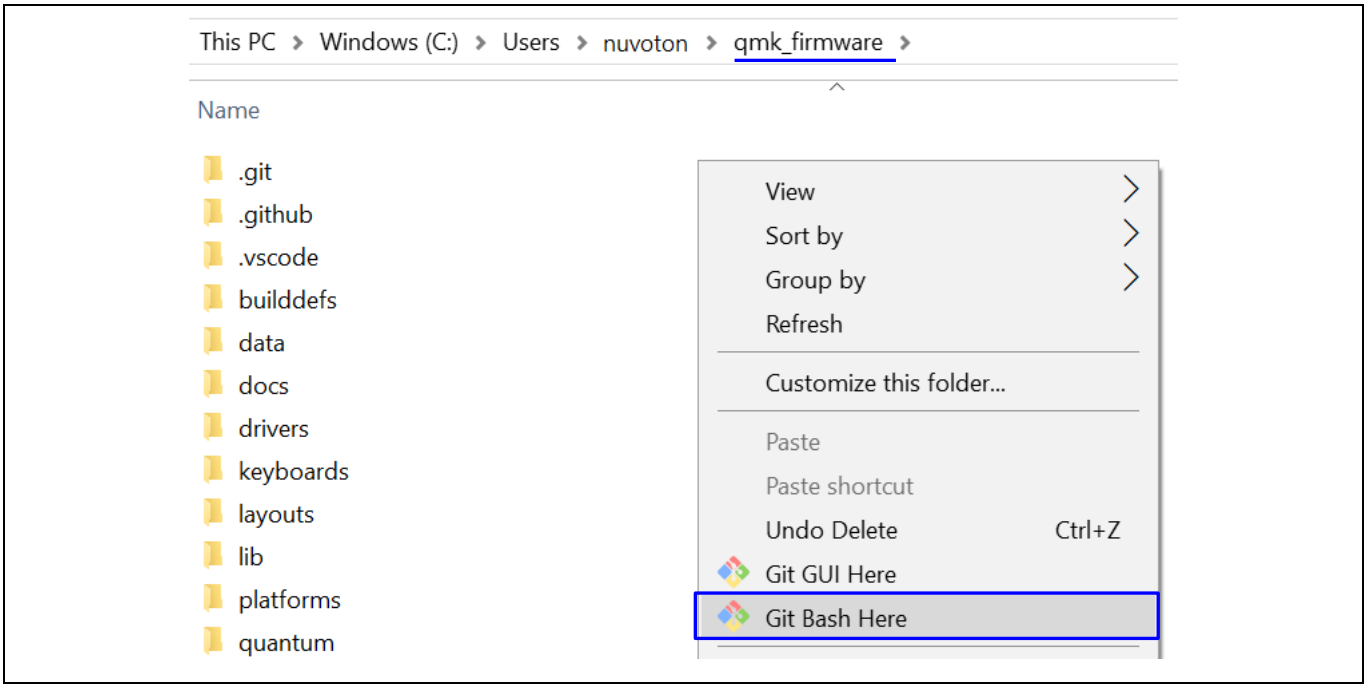

图 5-7 开启 Git

5. 在开启的 MINGW64 窗口中执行 git apply patch 命令，加入 NUC029xGE 的相关档案到 qmk_firmware 程序。

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

6. 切换工作目录到 qmk_firmware/lib/chibios-contrib。

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

7. 执行 git apply patch 命令，加入 NUC029xGE 的相关档案到 chibios-contrib 程序。

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

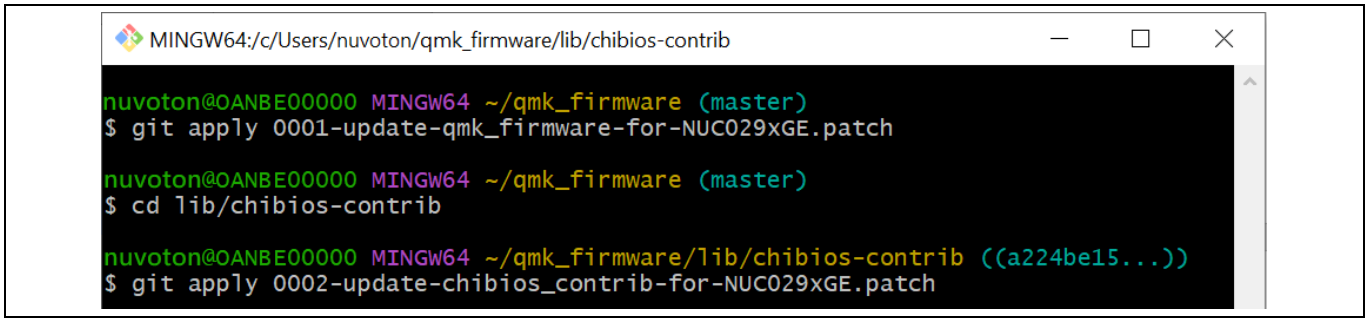

图 5-8 加入 NUC029xGE 的相关档案到 qmk_firmware 和 chibios-contrib 程序

5.1.2 编译键盘程序

加入 NUC029xGE 的相关档案之后开始编译程序，在 QMK MSYS 执行 qmk compile 命令，编译完成后产生的韧体档案会新增在 qmk_firmware 目录底下。

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

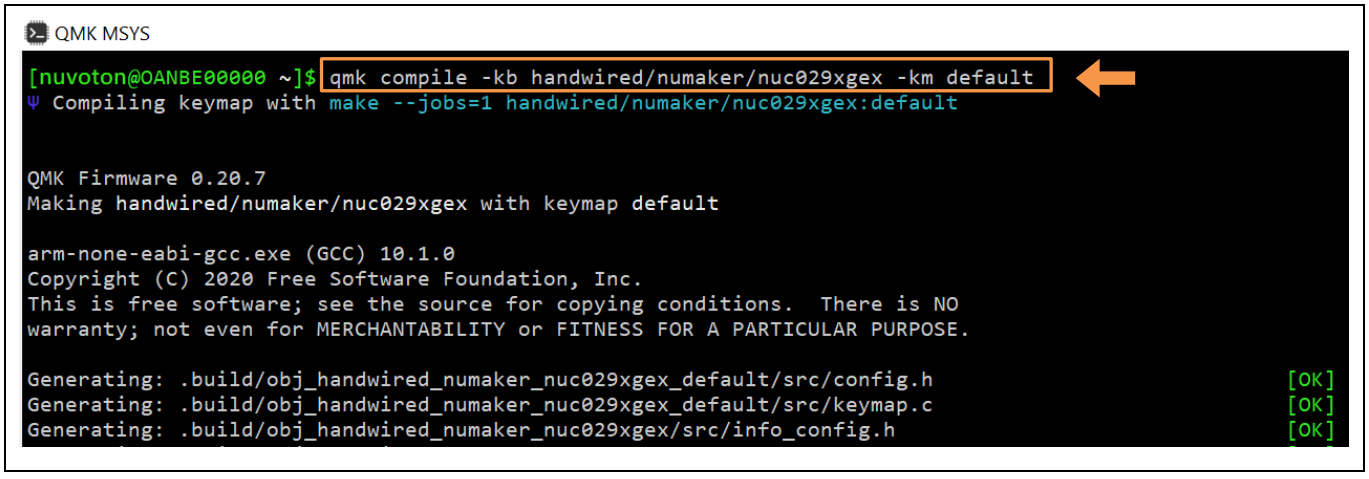

图 5-9 编译键盘程序

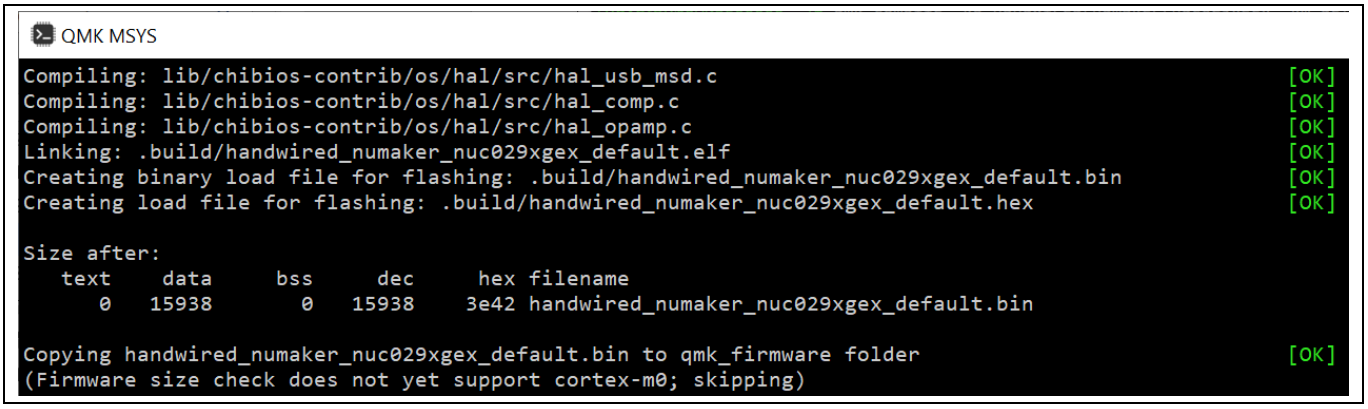

图 5-10 编译键盘程序完成

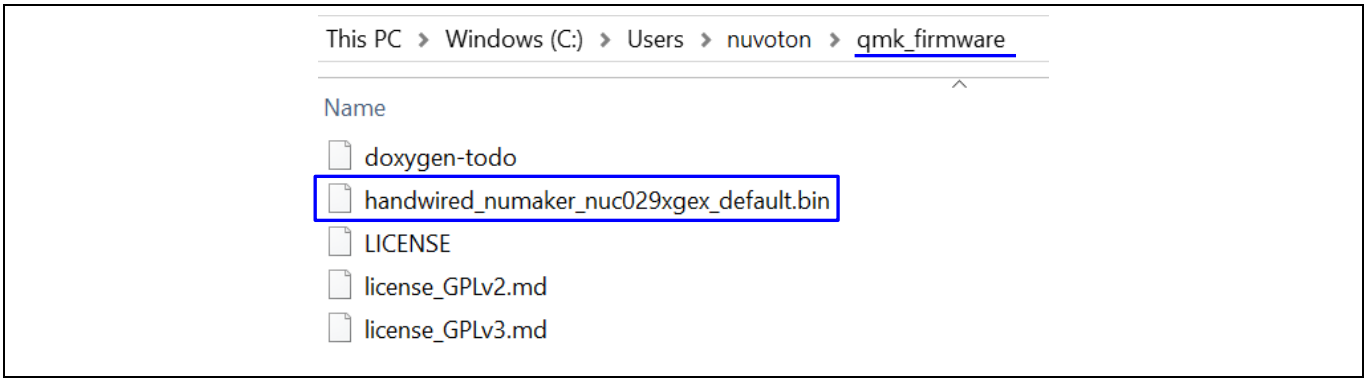

图 5-11 键盘程序韧体档案

5.2 Bootloader 程序

5.2.1 下载 Bootloader 程序

本范例代码的 bootloader 修改自 NUC029xGE BSP 的 ISP_DFU 范例程序，先从 GitHub 下载 NUC029xGE BSP 程序，切换至版本 v3.00.004，然后再透过 git apply patch 的方式加入修改的档案。

1. 在 MINGW64 窗口中，切换回工作目录。

```
cd ~
```

2. 执行 git clone 命令，从 GitHub 下载 NUC029xGE BSP 程序。

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

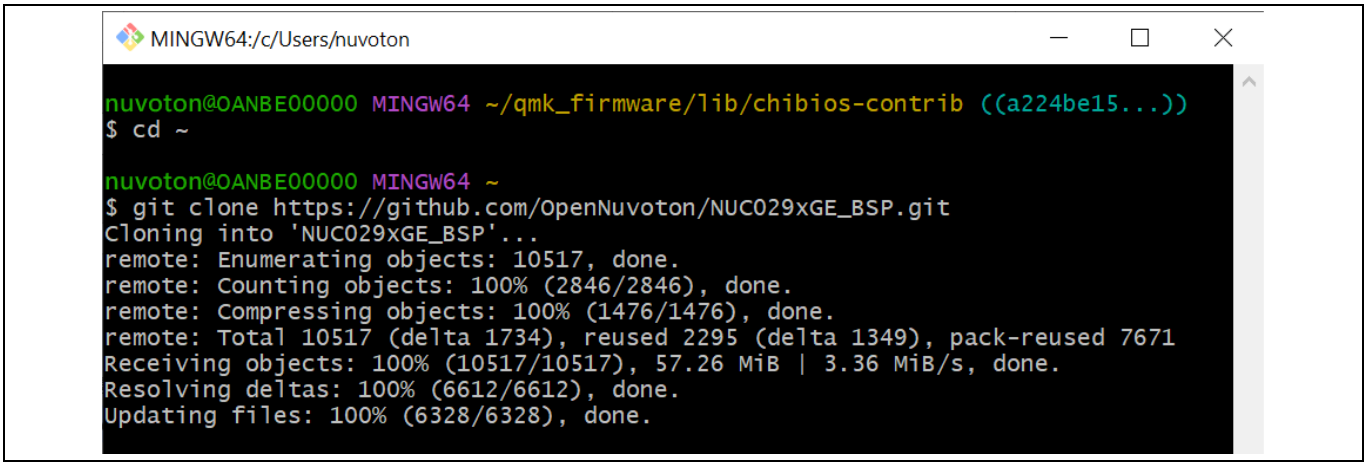

图 5-12 下载 NUC029xGE_BSP 程序

3. 切换到 NUC029xGE_BSP 目录。

```
cd NUC029xGE_BSP
```

4. 切换到 NUC029xGE_BSP 版本 v3.00.004。

```
git checkout v3.00.004
```

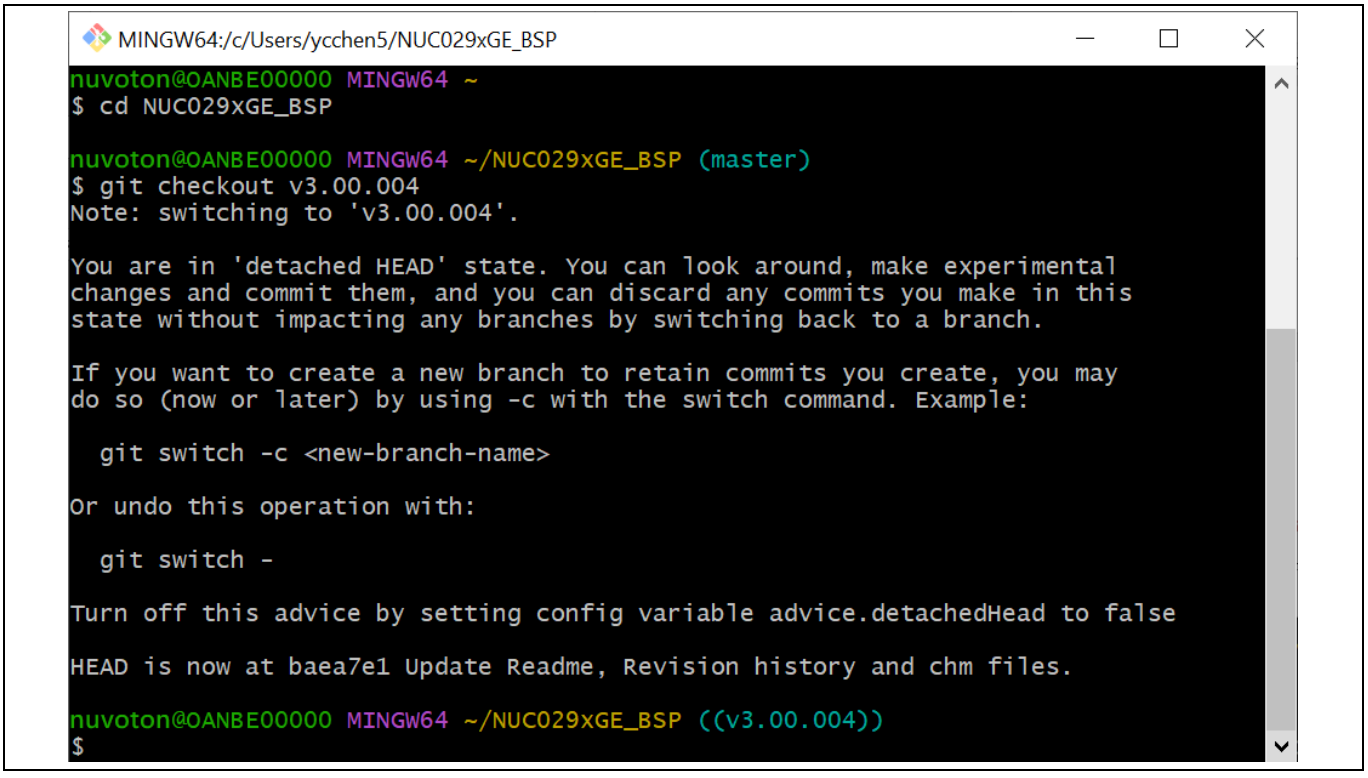

图 5-13 切换至 NUC029xGE_BSP 程序版本 v3.00.004

5. 将本范例代码 ISP_DFU git 补丁放在刚才下载的程序 NUC029xGE_BSP 目录底下。

图 5-14 本范例代码 ISP_DFU git 补丁档案位置

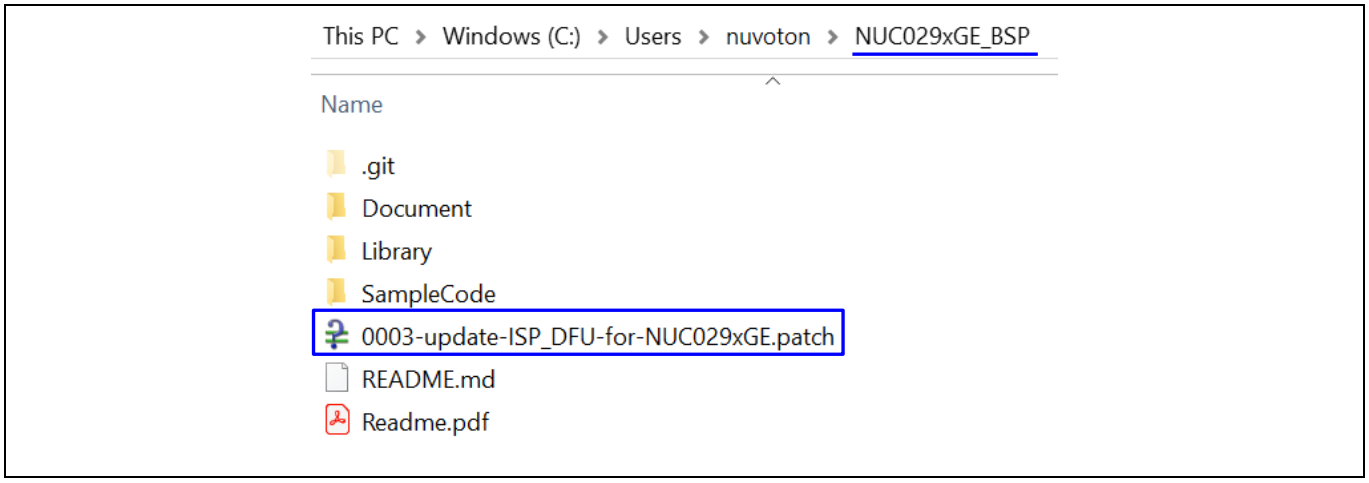

图 5-15 将 ISP_DFU git 补丁放在 NUC029xGE_BSP 目录底下

6. 在 MINGW64 窗口中，执行 git apply patch 命令，加入修改的 ISP_DFU 范例程序档案到 NUC029xGE_BSP 程序。

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

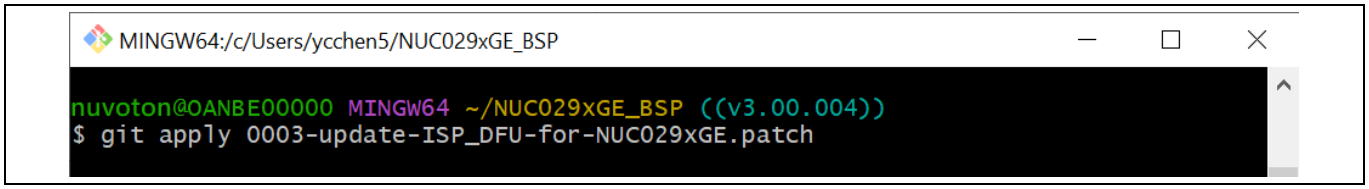

图 5-16 加入 ISP_DFU 范例程序的修改档案到 NUC029xGE_BSP 程序

5.2.2 编译 Bootloader 程序

1. 双击开启 NUC029xGE BSP 的 ISP_DFU 范例程序 Keil 项目，档案路径如图 5-17 所示。

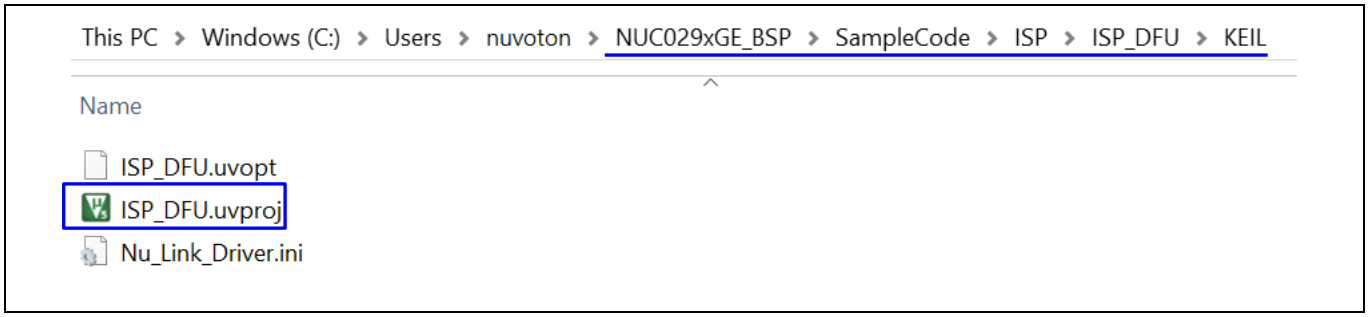

图 5-17 ISP_DFU 范例程序 Keil 项目路径

2. 在开启的 Keil 项目中，按下画面上方的[**Rebuild** 按钮编译程序，编译完成后产生范例程序 ISP_DFU 韧体档案 ISP_DFU.bin 于 obj 目录下。

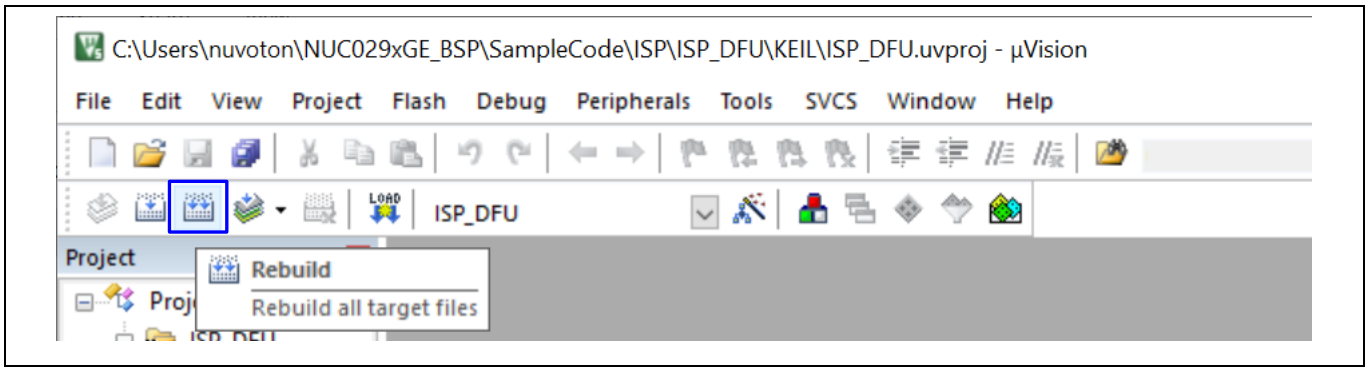

图 5-18 编译 ISP_DFU 范例程序

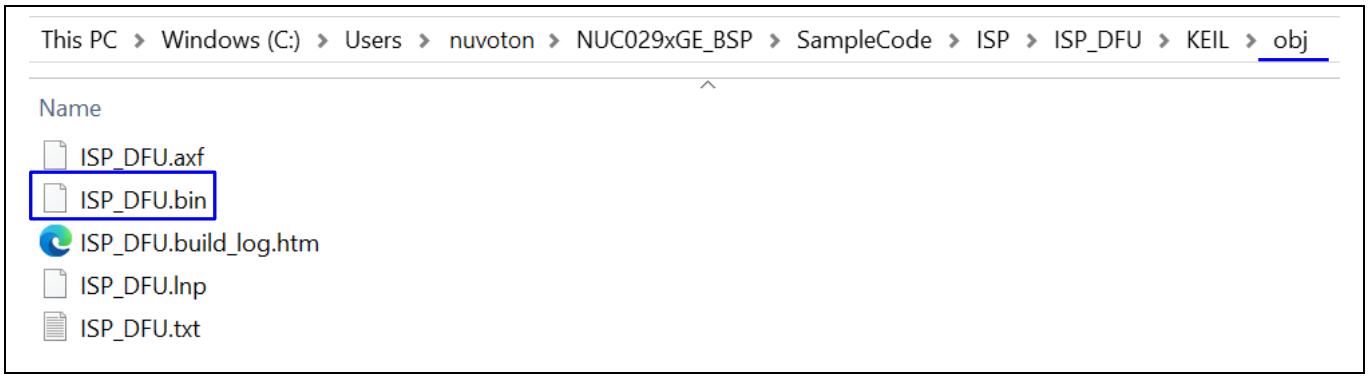

图 5-19 ISP_DFU 范例程序韧体档案

5.3 烧录代码

接下来我们要透过 Nuvoton NuMicro ICP Programming Tool 将编译好的键盘和 bootloader 韧体烧录到 NuMaker-NUC029SGE 板子上。

1. 开启 Nuvoton NuMicro ICP Programming Tool，在[MCU 工具系列]下拉选单中选择[NUC029 系列]，按[继续>>]按钮继续。

图 5-20 Nuvoton NuMicro ICP Programming Tool

2. 将 NuMaker-NUC029SGE 板子上 Nu-Link2-Me 端的 USB 连接至计算机，在[连线状态检测]栏位按[连接]按钮，连接 NuMaker-NUC029SGE 板子。

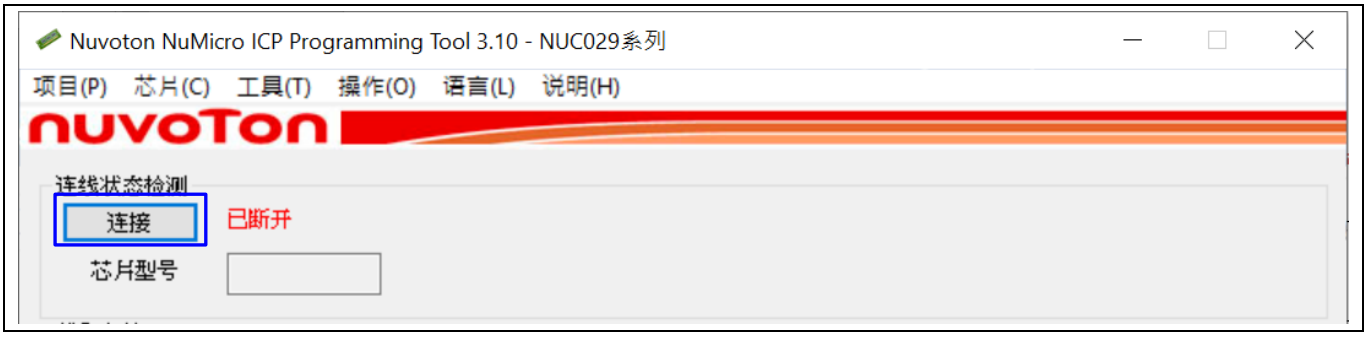

图 5-21 连接 NuMaker-NUC029SGE 板子

3. 连线成功后，在[加载文件]栏位选择[LDROM]的韧体档案为 bootloader 程序，[APROM]的韧体档案为键盘程序。

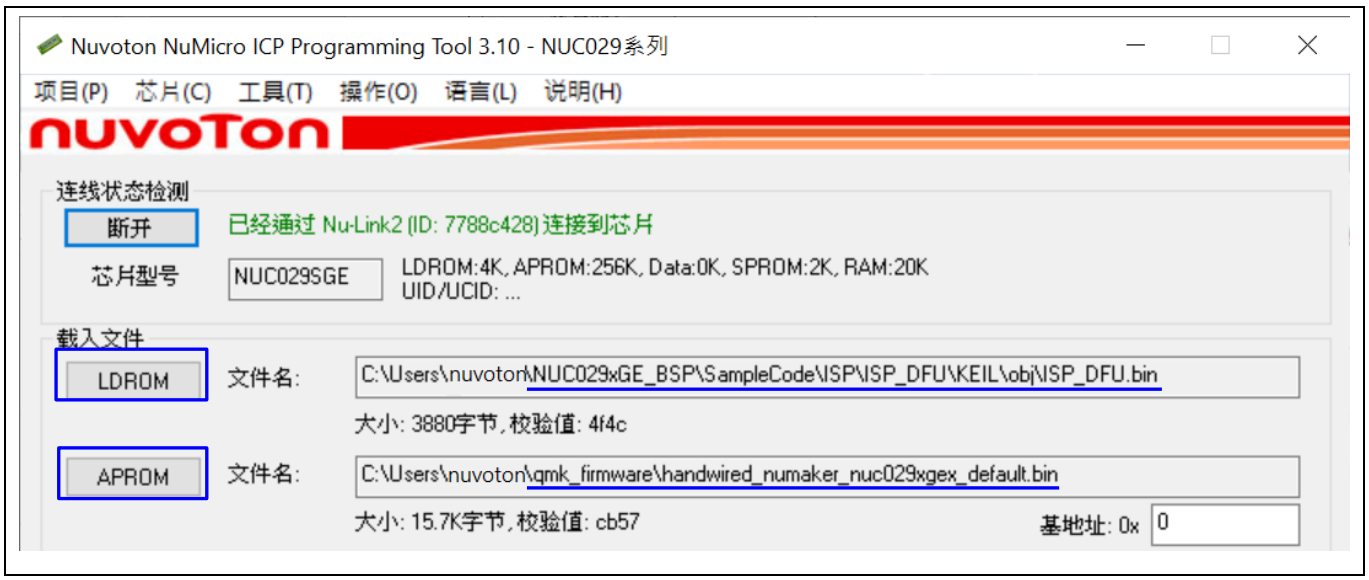

图 5-22 选择 LDROM 和 APROM 程序韧体档案

4. 在最底下的[编程]栏位，选择要编程的区域包括[LDROM] 和[APROM]，然后按[开始]按钮，开始编程，最后显示编程完成信息。

图 5-23 开始编程

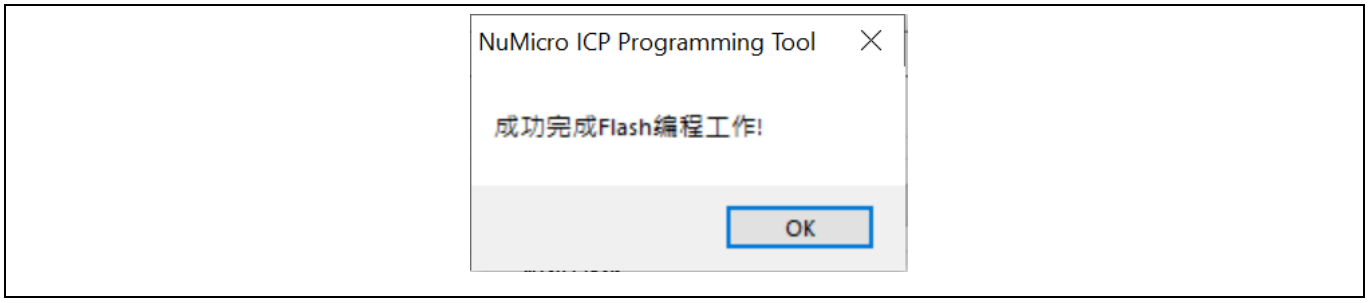

图 5-24 编程完成

编程完成后，将 NuMaker-NUC029SGE 端的 USB 连接至计算机，会在计算机上识别为一个 USB HID 键盘设备，按下 NuTFT_LCM 上的按钮就可以仿真键盘上的按键。

5.4 韧体更新

QMK 提供 QMK Toolbox 软件作为更新韧体的工具，以下将示范透过 QMK Toolbox (Nuvoton Edition) 更新键盘韧体为支持 VIA 功能的 QMK 键盘程序。

1. 开启 QMK MSYS，执行 qmk compile 命令，编译支持 VIA 功能的 QMK 键盘程序，编译完成后的韧体档案会新增在 qmk_firmware 目录底下。

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

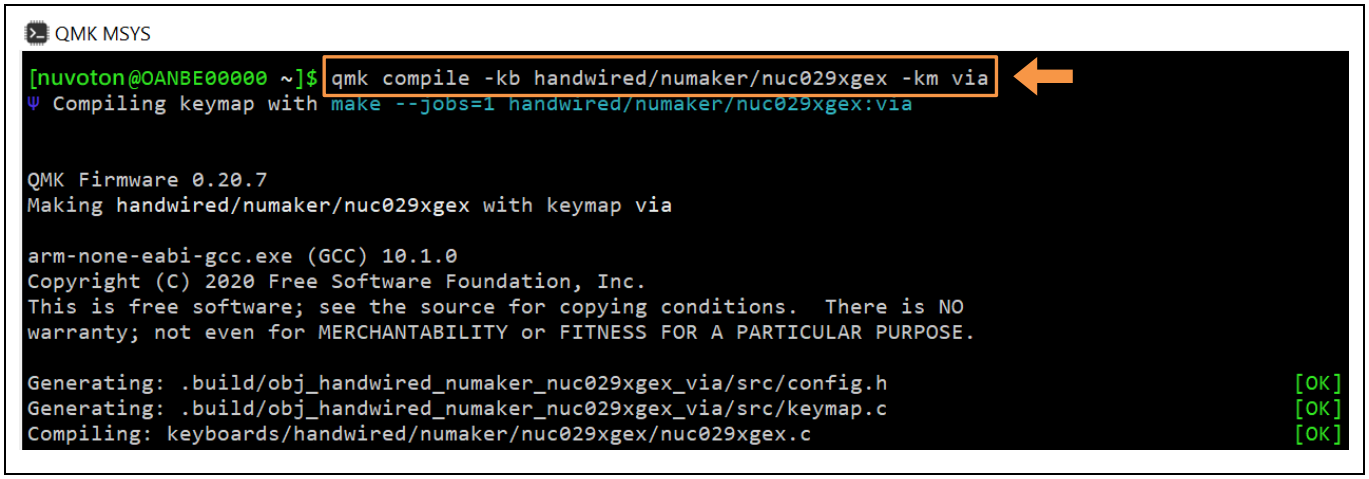

图 5-25 编译支持 VIA 功能的键盘韧体

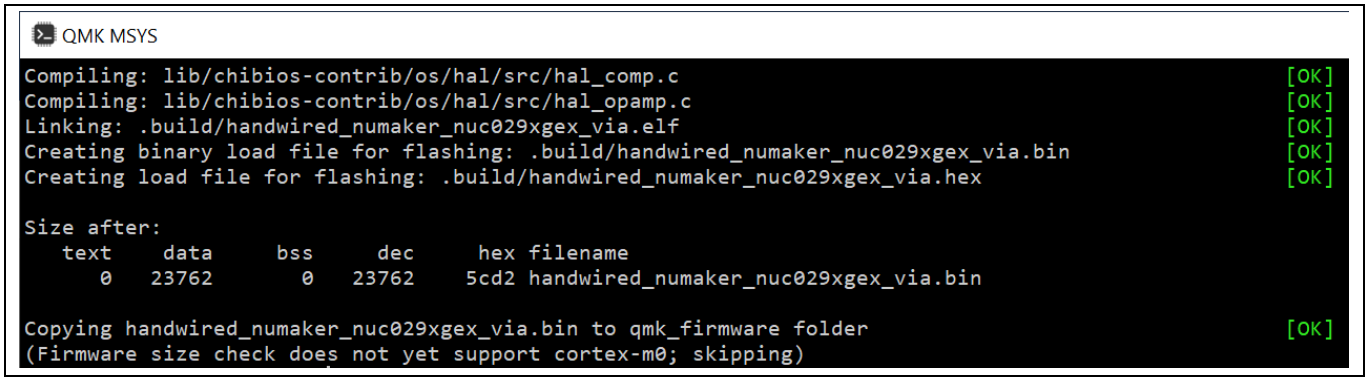

图 5-26 编译支持 VIA 功能键盘韧体完成

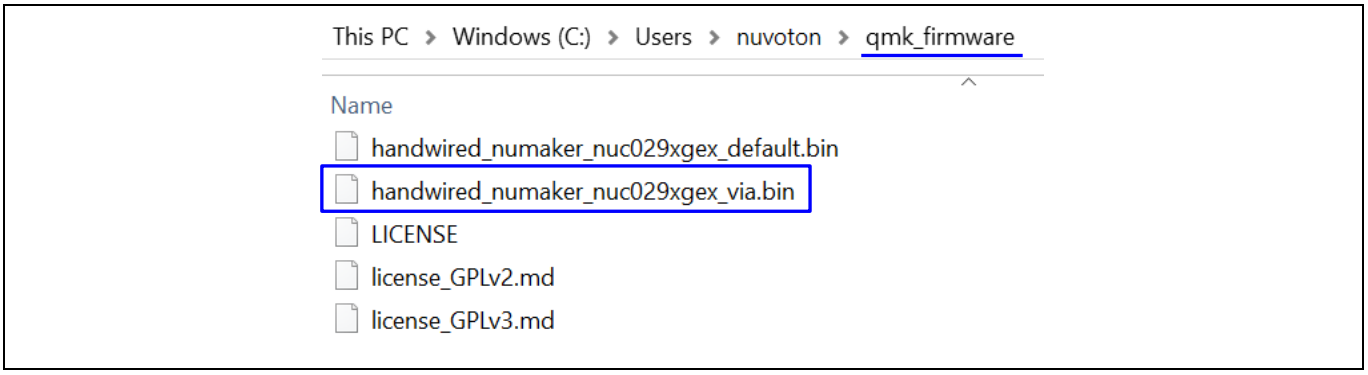

图 5-27 支持 VIA 功能键盘韧体档案

2. 将板子切换到 bootloader 模式：连接 NuMaker-NUC029SGE 端的 USB 至计算机，按住板子上的 SW1 按钮，然后再按一下 RESET 按钮。

3. 进入 bootloader 会被识别为一个 Nu_DFU 装置，识别到 Nu_DFU 装置后即可放开 SW1 按钮。如果无法识别到 Nu_DFU 装置，则需要安装驱动，档案路径如图 5-29 所示。

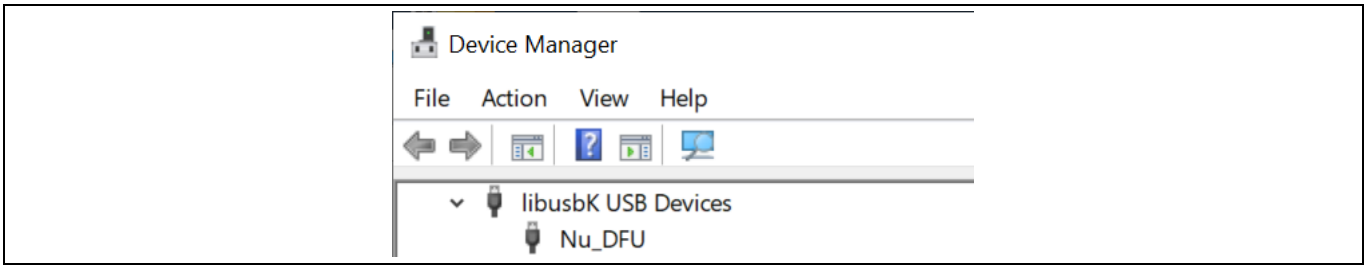

图 5-28 Nu_DFU 装置

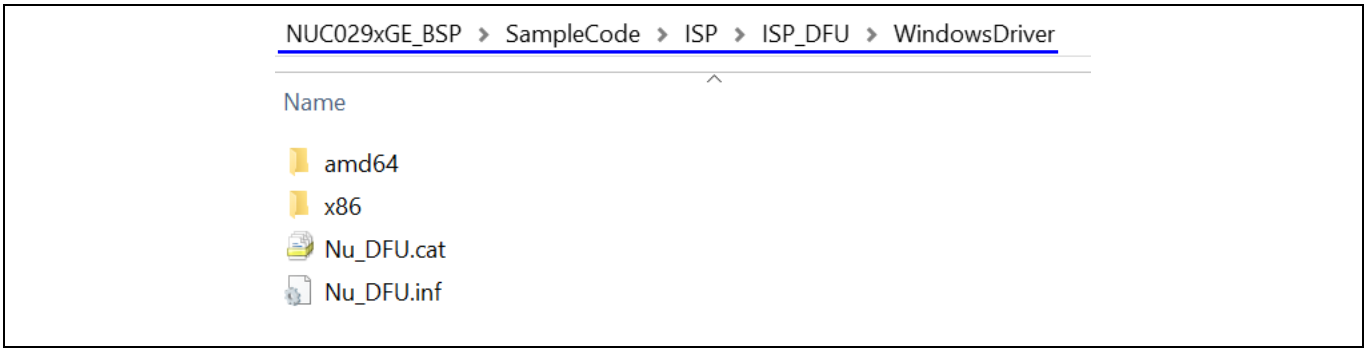

图 5-29 Nu_DFU 装置驱动档案路径

4. 开启本范例代码里的 QMK Toolbox (Nuvoton Edition)，档案路径如图 5-30 所示。如果开启时显示如图 5-31 信息询问是否要安装装置驱动，注意此时询问安装的驱动适用于其他支持 QMK 的装置，但不包括我们现在要用的 Nu_DFU 装置，Nu_DFU 装置的驱动程序仍需要手动安装，使用者在此可以自行决定是否要安装其他装置驱动。

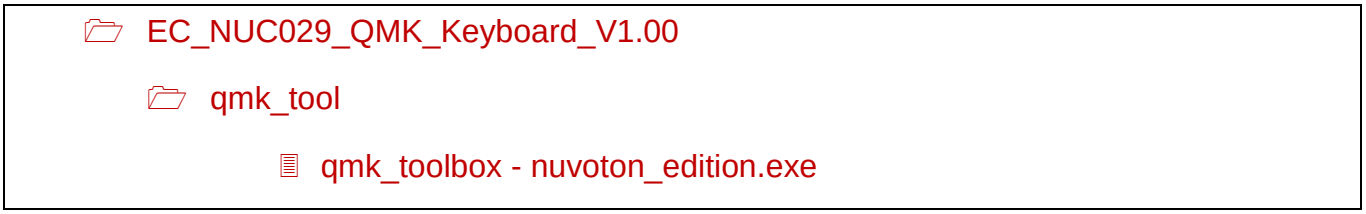

图 5-30 QMK Toolbox (Nuvoton Edition) 档案路径

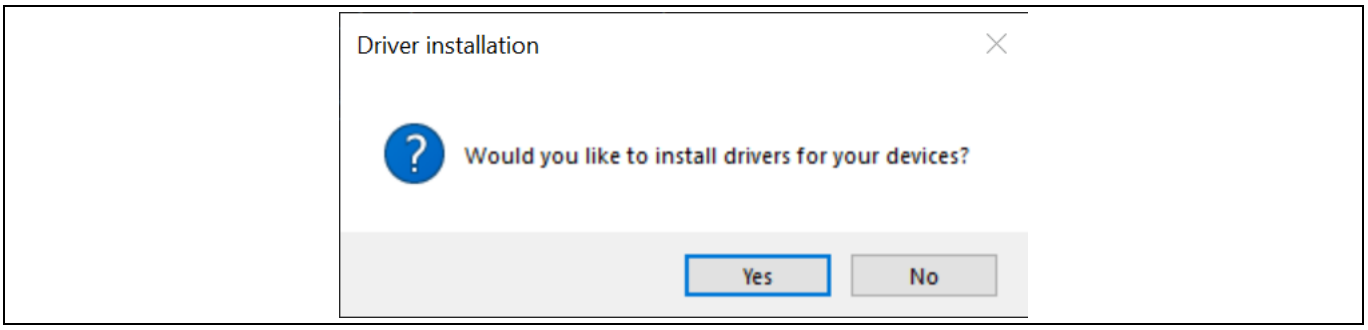

图 5-31 询问是否安装装置驱动(但不包括 Nu_DFU 装置)

5. QMK Toolbox (Nuvoton Edition) 显示侦测到 Nuvoton DFU Device 信息。

6. 按[Open]按钮选择要更新的韧体档案。

7. 按[Flash]按扭开始进行韧体更新，最后显示韧体更新完成信息。

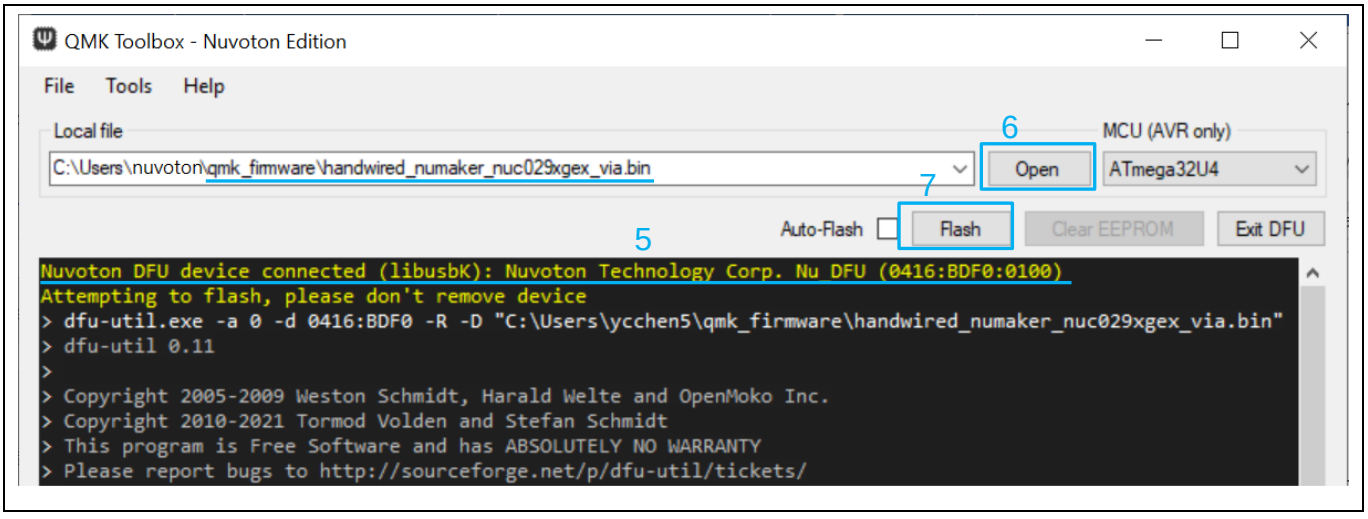

图 5-32 QMK Toolbox (Nuvoton Edition) 进行韧体更新

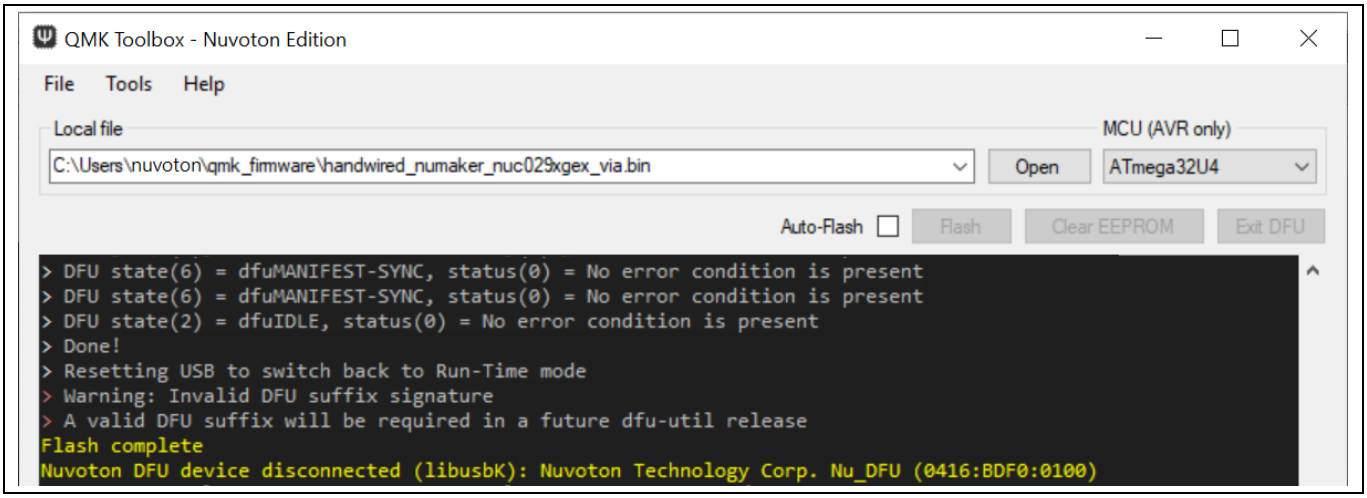

图 5-33 QMK Toolbox (Nuvoton Edition) 进行韧体更新完成

5.5 VIA

VIA 是一款可以在线及时修改按键定义的工具，而不需要再重新编译和更新韧体，使用时需搭配支持 VIA 功能的 QMK 键盘程序，注意 VIA 功能在默认的 QMK 键盘程序未被开启。

1. 开启 VIA 网页 App，在上方选单选择[SETTINGS]页面。

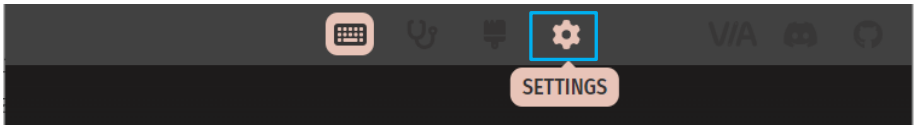

图 5-34 VIA 网页 App

2. 在 [SETTINGS]页面如图 5-35 所示，启用[Show Design tab]功能，上方选单即会多出[DESIGN]页面。

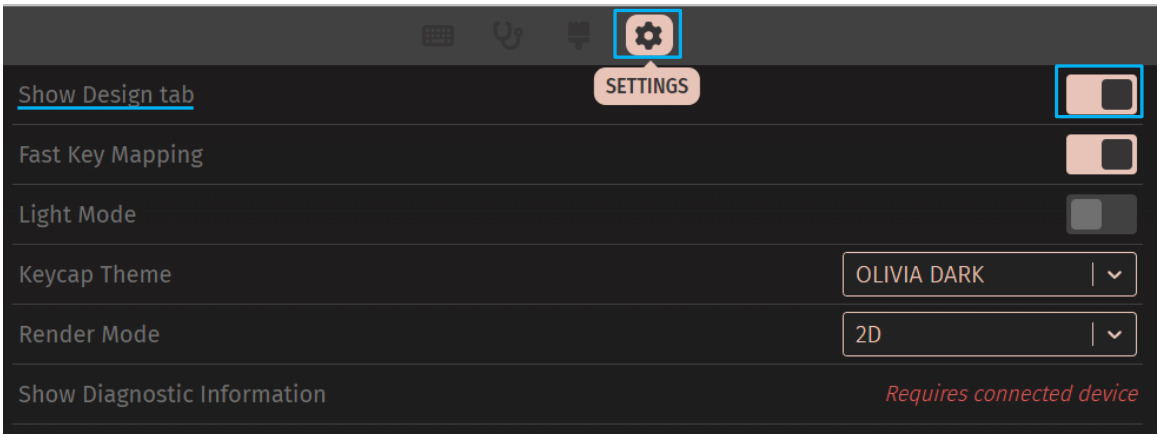

图 5-35 VIA 工具 [SETTINGS]页面

3. 在[DESIGN]页面如图 5-36 所示，关掉[Use V2 definitions (deprecated)]功能。

4. 按[Load Draft Definition]选项的[Load]按钮，选择本范例代码中给 VIA 工具使用的 NUC029xGE JSON 档案，档案路径如图 5-37 所示。

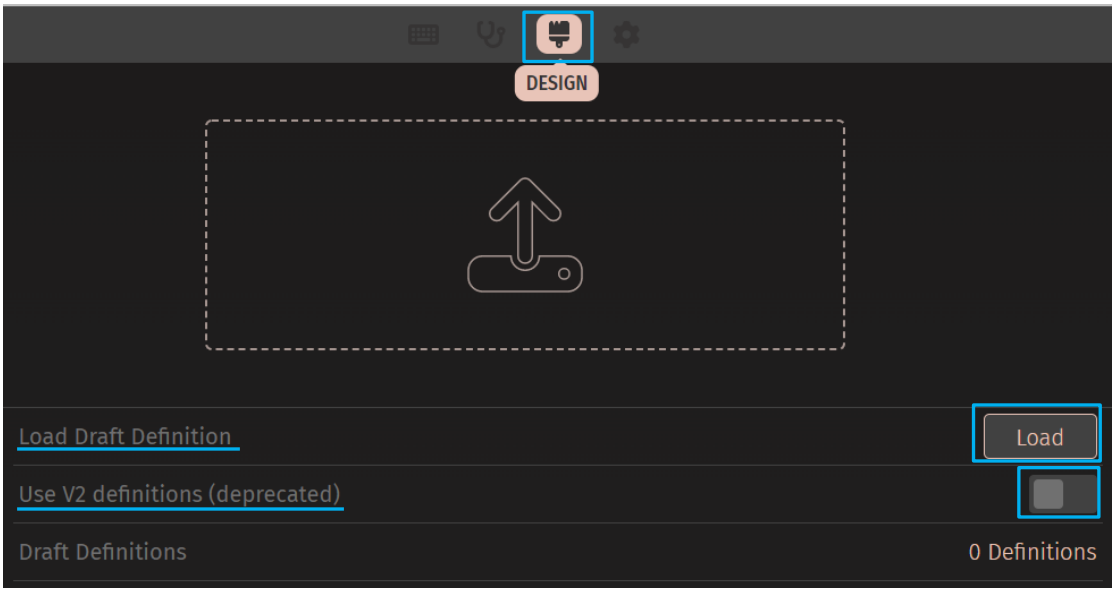

图 5-36 VIA 工具 [DESIGN]页面

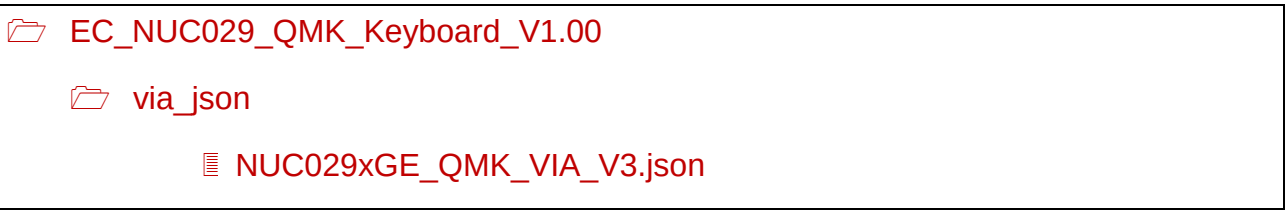

图 5-37 给 VIA 工具使用的 NUC029xGE JSON 档案路径

5. 加入 JSON 档案后，在[DESIGN]页面显示 NUC029xGE 的设计有两个按键。

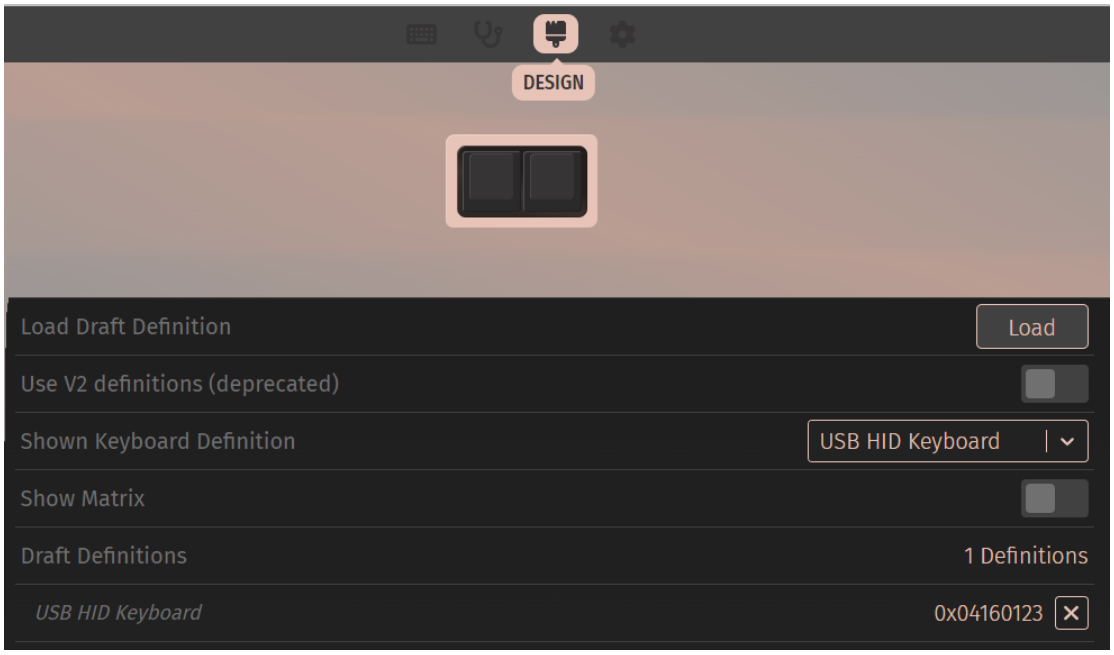

图 5-38 NUC029xGE 的设计有两个按键

6. 如果有认到支持 VIA 工具的 HID 设备，会跳出要求连接该设备的窗口，选择该设备后按[Connect]连接。

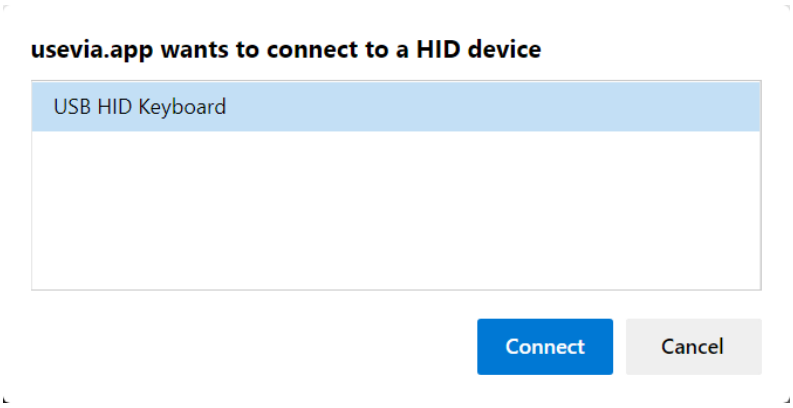

图 5-39 连接 HID 设备

7. 如果没有自动跳出连接窗口，也可以在[CONFIGURE]页面上按[Authorize device +]按钮尝试手动连接。

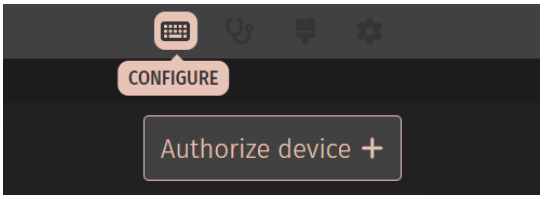

图 5-40 按[Authorize device +]按钮手动连接 HID 设备

8. 连接上 HID 设备之后，在[CONFIGURE]页面显示设计上的两个按键，分别是 [ESC]和 [1]。

图 5-41 按键[ESC]和[1]

9. 如果要修改按键定义，先点选要修改的按键例如按键[1]，当按键[1]被选中时会闪烁，接着再点选下方按键列表中想要更改的目标按键，例如按键[4]，如图 5-42 所示。

图 5-42 修改按键定义

10. 此时我们可以看到画面显示：第二颗按键已经从原本的按键[1]更改成按键[4]，如图 5-43 所示。

图 5-43 按键 [ESC]和[4]

11. VIA 工具也附有键盘测试工具，在上方选单中的[KEY TESTER]页面就可以直接测试按键修改前和修改后的结果，如图 5-44 所示，可以看到两颗按键已经修改成[ESC]和[4]。

图 5-44 VIA 键盘测试结果

6. 修订纪录

Date	Revision	Description
2023.05.23	1.00	初始发布。

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.