

## QMK鍵盤

NuMicro® 32位系列微控制器範例代碼介紹

### 文件資訊

應用簡述	本範例代碼使用開放的 QMK 鍵盤韌體原始碼在 NUC029 實作鍵盤
BSP 版本	NUC029xGE_Series_BSP_CMSIS_V3.00.004
開發平台	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](http://www.nuvoton.com)

## 1. 概述

[QMK](#)(Quantum Mechanical Keyboard)是一個社區維護的應用於電腦輸入設備的開放原始碼，像是鍵盤、滑鼠、MIDI 等設備。市售鍵盤可以直接使用 QMK 韌體作為產品，或支援更新為 QMK 韌體，使用者就可以自行修改功能，製作專屬個人客製化的鍵盤。除了開放的原始碼，QMK 也提供其他協助程式開發的工具，例如：QMK MSYS、QMK Configurator、QMK Toolbox 等。本範例代碼將使用 QMK 鍵盤韌體以及搭配 QMK 開發工具，在 NUC029xGE 上實作一個模擬的鍵盤。

### 1.1 原理

QMK 在 NUC029xGE 上實作模擬的鍵盤，包括一個鍵盤韌體本身和一個更新韌體的流程，鍵盤韌體放在 APROM，更新韌體由 bootloader 負責，放在 LDRUM，如圖 1-1 所示本範例代碼在 NUC029xGE 的記憶體映射。

系統開機後，主要執行 APROM 鍵盤程式，在電腦上識別為一個 USB HID 鍵盤設備。本範例代碼利用 NuTFT\_LCM 板子上的兩個按鈕 SW1 和 SW2 模擬鍵盤的兩個按鍵，SW1 為按鍵 **Esc**，SW2 為按鍵 **1**。

韌體更新由 LDRUM bootloader 負責，進入 bootloader 的方式為：按住按鍵 **Esc** 後，再按一下 NuMaker-NUC029SGE 板子上的 RESET 按鈕。重跑 APROM 程式，檢查按鍵 **Esc** 如果有被按下，則切換到 LDRUM 執行 bootloader 程式，在電腦上識別為一個 USB DFU 設備，之後就可以透過 DFU 協定更新韌體。

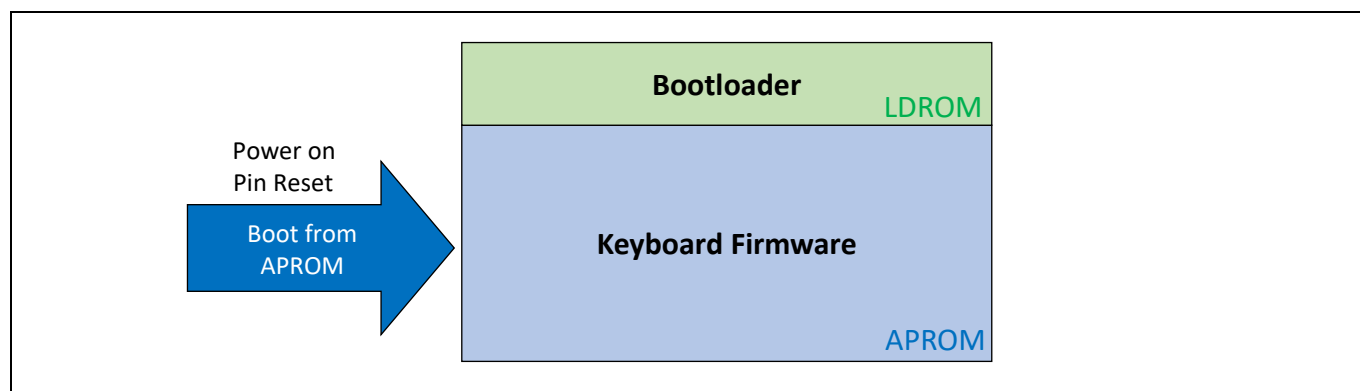


圖 1-1 NUC029xGE 記憶體映射

## 1.2 執行結果

透過 QMK Configurator 提供的鍵盤測試工具來測試鍵盤功能，進入[鍵盤測試網頁](#)，按下 NuTFT\_LCM 板子上的兩個按鈕，就可以看到結果顯示按鍵 **Esc** 和 **1** 被按下，如圖 1-2 所示 QMK Configurator 鍵盤測試結果。

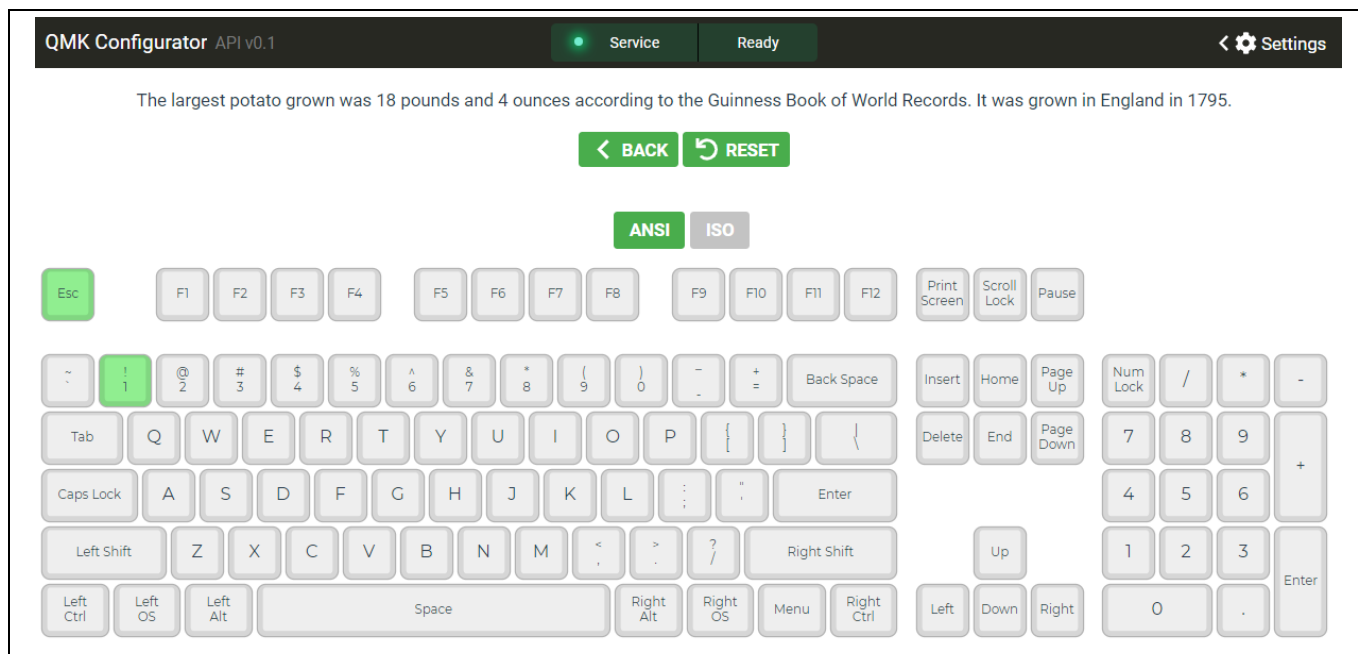


圖 1-2 QMK Configurator 鍵盤測試結果

## 2. 代碼介紹

### 2.1 主要函數

初始化設定：

- platform\_setup()設定和開發平台 NUC029xGE 相關的硬體，包括系統時鐘、GPIO、USB 和 systick。
- protocol\_setup()設定和 USB 相關的功能。
- keyboard\_setup()設定和鍵盤相關的功能，開啟 VIA 功能時需要設定 flash。
- protocol\_init()初始化和啟用 USB、VIA 和 bootmagic 功能。

主要迴圈：

- protocol\_task()負責鍵盤矩陣掃描和 USB 服務。
- housekeeping\_task()負責其他需要週期的被執行的程式，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 按鍵

本範例代碼使用 NuTFT\_LCM 板子上的兩個按鈕模擬鍵盤的兩個按鍵，SW1 按鈕連接至 NuMaker-NUC029SGE 板子上的 PC.4，SW2 按鈕連接至 PC.5。在 config.h 定義 DIRECT\_PIN 值為 {{C4, C5}}，在 nuc029xgex.h 定義 LAYOUT 包括兩個元素 k00 與 k01，在 keymap.json 定義 LAYOUT 的兩個元素各代表按鍵碼[KC\_ESCAPE]和[KC\_1]。按鍵碼定義於 keycodes.h，使用者也可以自行修改成其他想要的按鍵碼。

*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 功能可以用來修改按鍵定義，如果要使用 VIA 功能，在 `rules.mk` 設定 `VIA_ENABLE` 值為 `yes`。VIA 需要將按鍵定義紀錄在非揮發性記憶體，像是 EEPROM 或 embedded flash，在 `rules.mk` 中設定本範例代碼使用 `wear leveling` 方式記錄按鍵定義於 embedded flash。在 `config.h` 設定 `BACKING_STORE_WRITE_SIZE` 是一次寫入 flash 的資料為 4 bytes，定義 NUC029xGE 的 flash 大小為 256KB，允許 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 提供切換到 bootloader 模式的功能，在 rules.mk 設定 BOOTMAGIC\_ENABLE 值為 yes。鍵盤接上 USB 後，QMK 程式會偵測指定的按鍵有沒有被按下，預設指定的按鍵為位置(0,0)的 [ESC]，如果 [ESC] 有被按下，則清除 EEPROM 資料並切換到 bootloader 模式，切換的程式在 nuc029xgex.c 的 bootloader\_jump() 裡實現。

*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 LDR0M bootloader 程式裡檢查如果 **Esc** 按鍵有被按下，表示使用者指定要切換到 bootloader 模式，則留在 LDR0M 等待韌體更新，如果 **Esc** 按鍵沒有被按下，則執行 APROM 鍵盤程式。

*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->AIRCRCR = (V6M_AIRCRCR_VECTKEY_DATA | V6M_AIRCRCR_SYSRERESETREQ);
}

int32_t main(void)
{
    .
    .
    .

    /* Keep in LDR0M 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. 軟體與硬體需求

#### 3.1 軟體需求

- GitHub Repository
  - [gmk/gmk\\_firmware](#)
  - [OpenNuvoton/NUC029xGE\\_BSP](#) v3.00.004
- 開發工具
  - [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 硬體需求

- 電路元件
  - NuMaker-NUC029SGE V1.0
  - NuTFT\_LCM V1.4
- 示意圖
  - 將 NuTFT\_LCM 上板和 NuMaker-NUC029SGE 下板相接
  - 將 NuMaker-NUC029SGE 端的 USB 連接到電腦

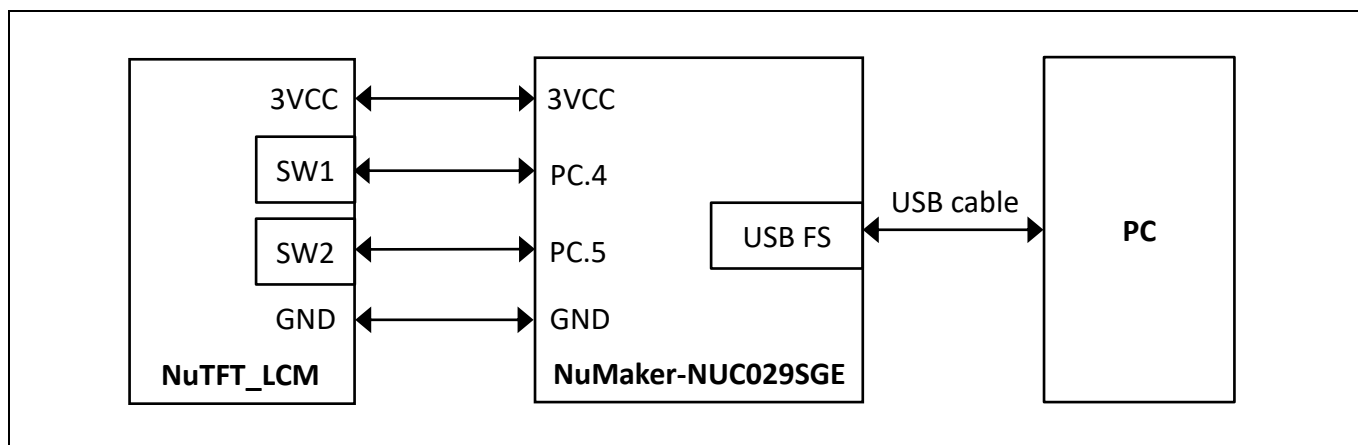


圖 3-1 NuMaker-NUC029SGE 接線示意圖



## 4. 目錄資訊





	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

圖 4-1 目錄資訊

## 5. 範例程式執行

本章節將介紹：如何下載編譯鍵盤和 bootloader 程式，編譯好的程式如何燒錄到板子上，接下來更新韌體的流程，最後介紹 VIA 工具的使用。

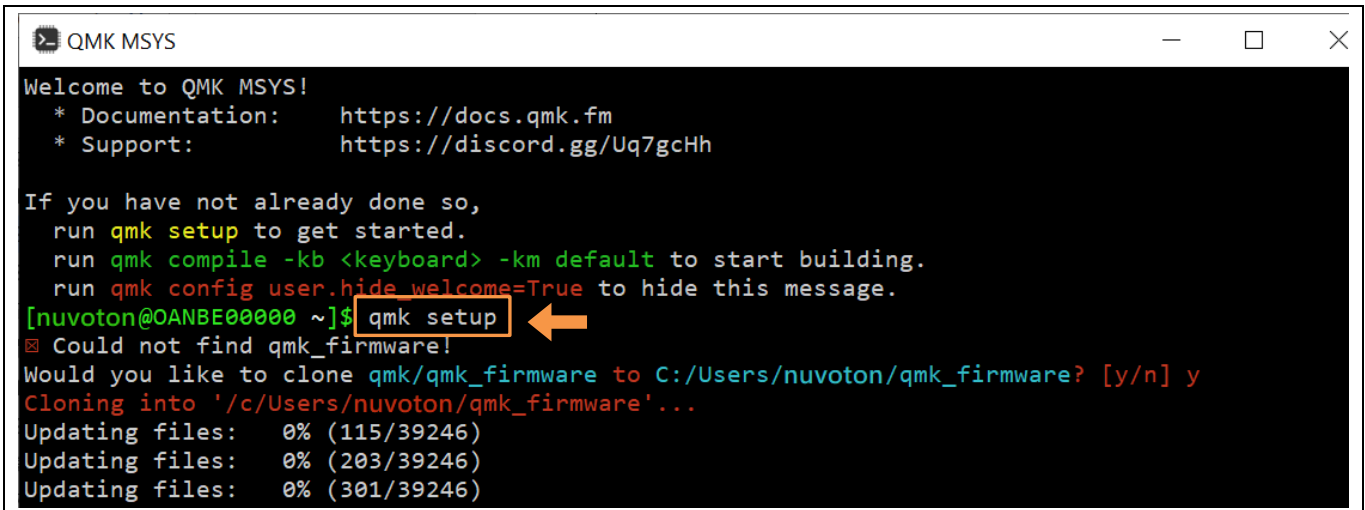
### 5.1 鍵盤程式

#### 5.1.1 下載鍵盤程式

本範例代碼修改自 QMK 鍵盤韌體以及使用 QMK MSYS 為編譯程式的工具，首先建立開發環境和下載鍵盤程式：

1. 安裝 [QMK MSYS](#)。
2. 開啟 QMK MSYS 執行 `qmk setup` 命令，檢查開發環境並從 GitHub 下載 `qmk_firmware` 程式。

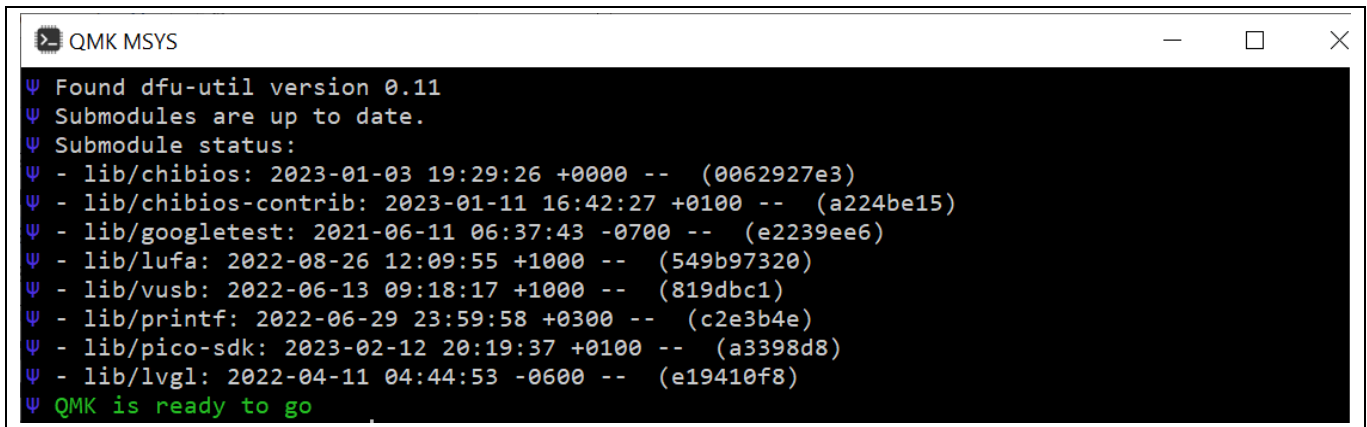
```
qmk setup
```



```
QMK MSYS
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)
```

圖 5-1 執行 `qmk setup` 命令



```

QMK MSYS
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
  
```

圖 5-2 執行 qmk setup 完成

接下來透過 git apply patch 的方式加入 NUC029xGE 的相關檔案到 qmk\_firmware 程式：

1. 安裝 [Git](#)。
2. 將本範例代碼 qmk\_firmware git 補丁放在剛才下載的程式 qmk\_firmware 目錄底下。
3. 將本範例代碼 chibios-contrib git 補丁放在剛才下載的程式 qmk\_firmware\lib\chibios-contrib 目錄底下。

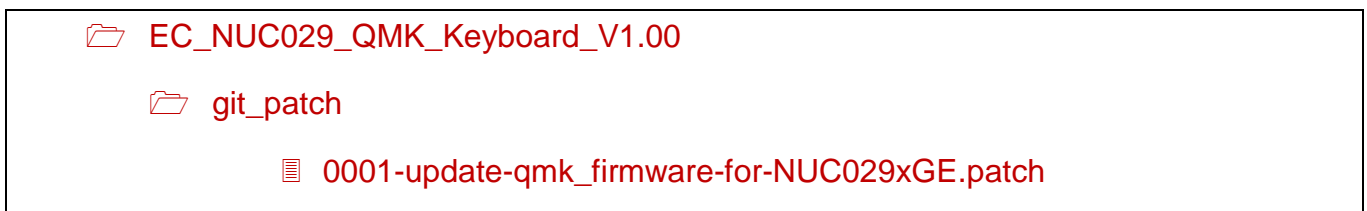


圖 5-3 本範例代碼 qmk\_firmware git 補丁檔案位置

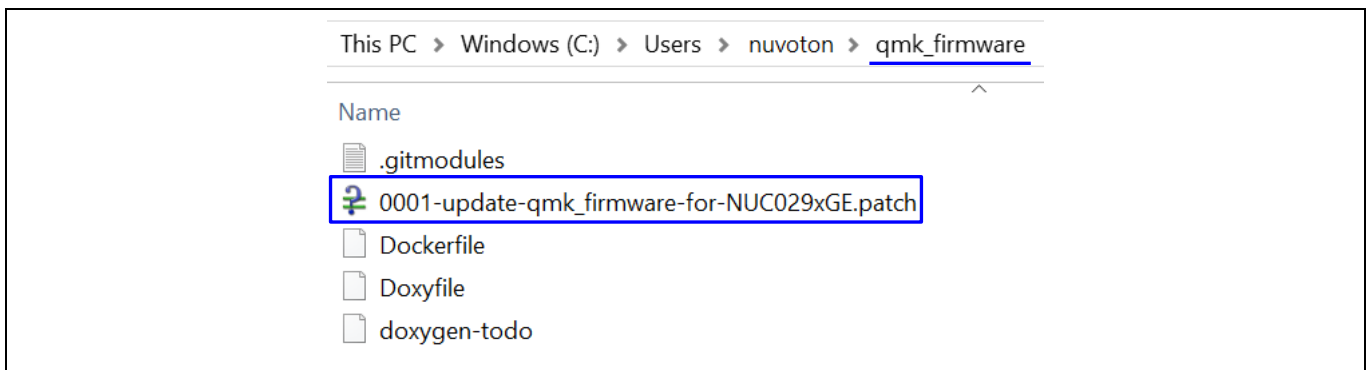


圖 5-4 將 qmk\_firmware git 補丁放在 qmk\_firmware 目錄底下

EC\_NUC029\_QMK\_Keyboard\_V1.00

git\_patch

0002-update-chibios\_contrib-for-NUC029xGE.patch

圖 5-5 本範例代碼 chibios-contrib git 補丁檔案位置

This PC > Windows (C:) > Users > nuvoton > qmk\_firmware > lib > chibios-contrib >

Name

.gitmodules

0002-update-chibios\_contrib-for-NUC029xGE.patch

AUTHORS.txt

lgm.yml

README.md

圖 5-6 將 chibios-contrib git 補丁放在 qmk\_firmware\lib\chibios-contrib 目錄底下

- 在 qmk\_firmware 目錄底下，按右鍵，選擇 [Git Bash Here]，開啟 Git。

This PC > Windows (C:) > Users > nuvoton > qmk\_firmware >

Name

.git

.github

.vscode

builddefs

data

docs

drivers

keyboards

layouts

lib

platforms

quantum

View

Sort by

Group by

Refresh

Customize this folder...

Paste

Paste shortcut

Undo Delete Ctrl+Z

Git GUI Here

Git Bash Here

圖 5-7 開啟 Git

- 在開啟的 MINGW64 視窗中執行 `git apply patch` 命令，加入 NUC029xGE 的相關檔案到 qmk\_firmware 程式。

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

6. 切換工作目錄到 `qmk_firmware/lib/chibios-contrib`。

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

7. 執行 `git apply patch` 命令，加入 NUC029xGE 的相關檔案到 `chibios-contrib` 程式。

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

```

MINGW64;C:/Users/nuvoton/qmk_firmware/lib/chibios-contrib
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
  
```

圖 5-8 加入 NUC029xGE 的相關檔案到 `qmk_firmware` 和 `chibios-contrib` 程式

### 5.1.2 編譯鍵盤程式

加入 NUC029xGE 的相關檔案之後開始編譯程式，在 QMK MSYS 執行 `qmk compile` 命令，編譯完成後產生的韌體檔案會新增在 `qmk_firmware` 目錄底下。

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

```

QMK MSYS
[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]
  
```

圖 5-9 編譯鍵盤程式

```

QMK MSYS
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)

```

圖 5-10 編譯鍵盤程式完成

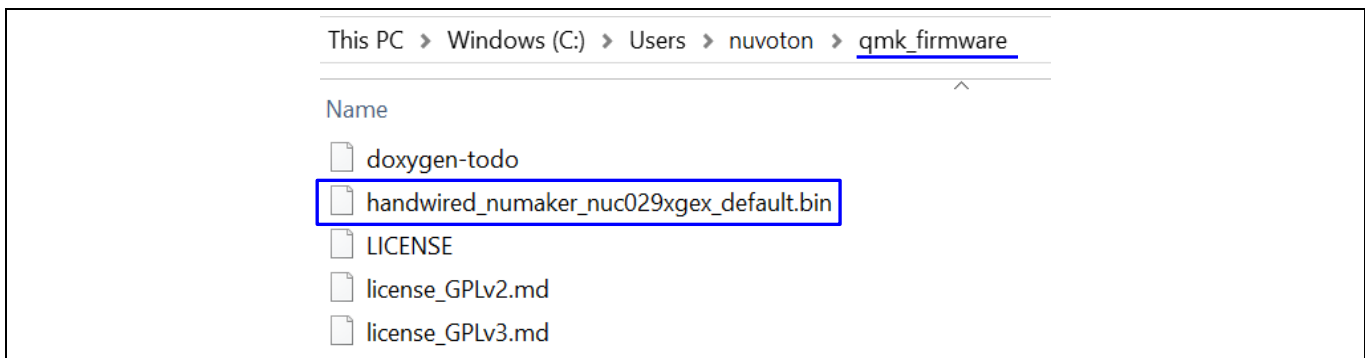


圖 5-11 鍵盤程式韌體檔案

## 5.2 Bootloader 程式

### 5.2.1 下載 Bootloader 程式

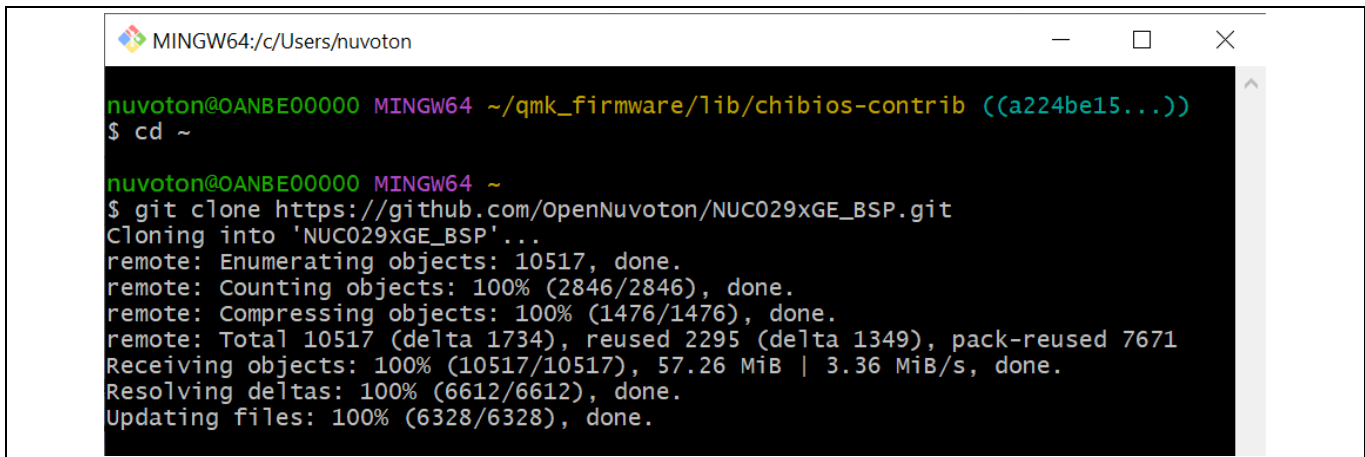
本範例代碼的 bootloader 修改自 NUC029xGE BSP 的 ISP\_DFU 範例程式，先從 GitHub 下載 NUC029xGE BSP 程式，切換至版本 v3.00.004，然後再透過 git apply patch 的方式加入修改的檔案。

1. 在 MINGW64 視窗中，切換回工作目錄。

```
cd ~
```

2. 執行 git clone 命令，從 GitHub 下載 NUC029xGE BSP 程式。

```
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.
  
```

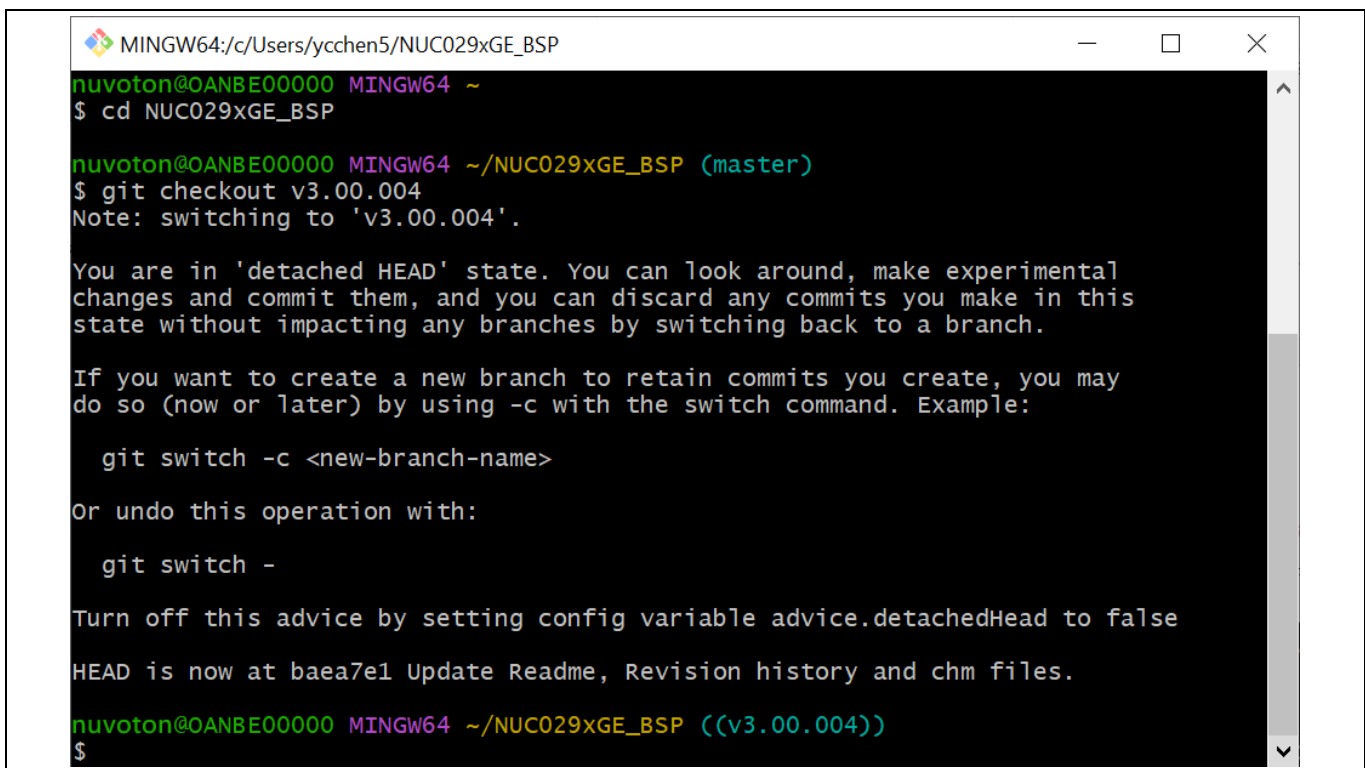
圖 5-12 下載 NUC029xGE\_BSP 程式

3. 切換到 NUC029xGE\_BSP 目錄。

```
cd NUC029xGE_BSP
```

4. 切換到 NUC029xGE\_BSP 版本 v3.00.004。

```
git checkout v3.00.004
```



```

MINGW64:/c/Users/ycchen5/NUC029xGE_BSP
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))
$
  
```

圖 5-13 切換至 NUC029xGE\_BSP 程式版本 v3.00.004

- 將本範例代碼 ISP\_DFU git 補丁放在剛才下載的程式 NUC029xGE\_BSP 目錄底下。

 EC\_NUC029\_QMK\_Keyboard\_V1.00  
 git\_patch  
 0003-update-ISP\_DFU-for-NUC029xGE.patch

圖 5-14 本範例代碼 ISP\_DFU git 補丁檔案位置

This PC > Windows (C:) > Users > nuvoton > NUC029xGE\_BSP


Name
.git
Document
Library
SampleCode
 0003-update-ISP_DFU-for-NUC029xGE.patch
README.md
Readme.pdf

圖 5-15 將 ISP\_DFU git 補丁放在 NUC029xGE\_BSP 目錄底下

- 在 MINGW64 視窗中，執行 `git apply patch` 命令，加入修改的 ISP\_DFU 範例程式檔案到 NUC029xGE\_BSP 程式。

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

MINGW64:/c/Users/ycchen5/NUC029xGE\_BSP

```

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

圖 5-16 加入 ISP\_DFU 範例程式的修改檔案到 NUC029xGE\_BSP 程式



## 5.2.2 編譯 Bootloader 程式

1. 雙擊開啟 NUC029xGE BSP 的 ISP\_DFU 範例程式 Keil 專案，檔案路徑如圖 5-17 所示。

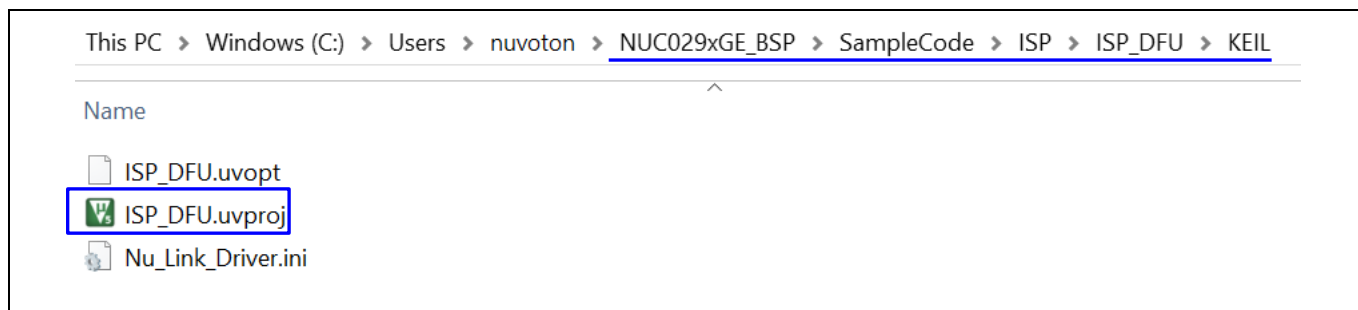


圖 5-17 ISP\_DFU 範例程式 Keil 專案路徑

2. 在開啟的 Keil 專案中，按下畫面上方的 **Rebuild** [Rebuild] 按鈕編譯程式，編譯完成後產生 ISP\_DFU 範例程式韌體檔案 ISP\_DFU.bin 於 obj 目錄下。

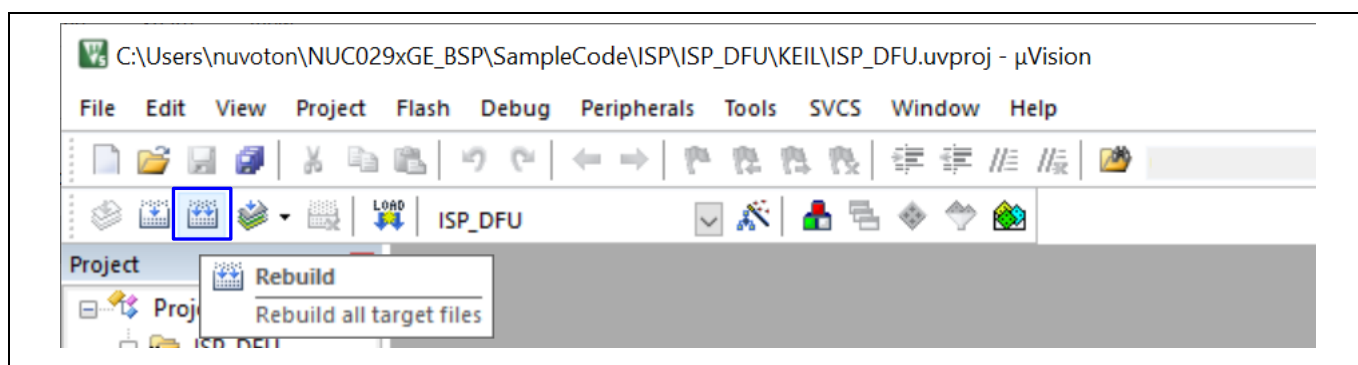


圖 5-18 編譯 ISP\_DFU 範例程式

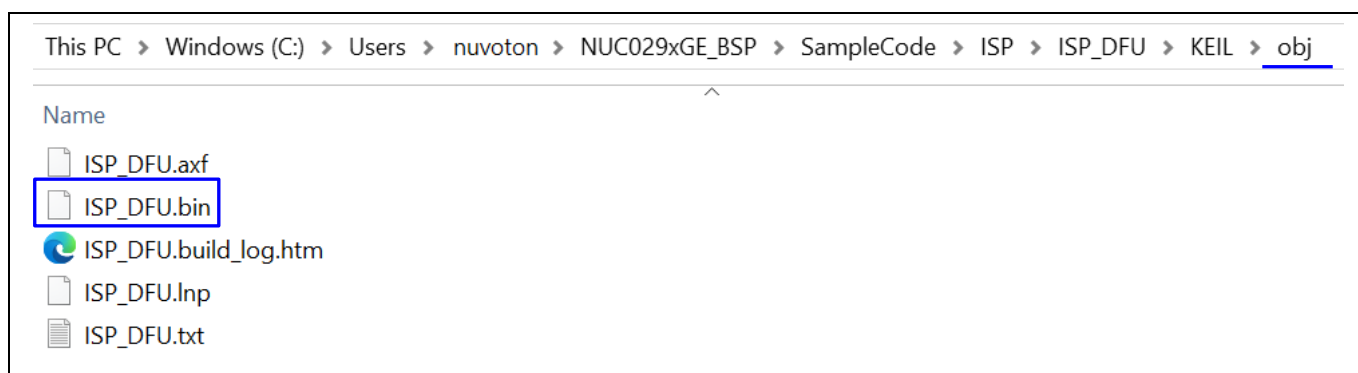


圖 5-19 ISP\_DFU 範例程式韌體檔案

### 5.3 燒錄程式

接下來我們要透過 Nuvoton NuMicro ICP Programming Tool 將編譯好的鍵盤和 bootloader 韌體燒錄到 NuMaker-NUC029SGE 板子上。

1. 開啟 Nuvoton NuMicro ICP Programming Tool，在[MCU 工具系列]下拉選單中選擇 [NUC029 系列]，按[繼續>>]按鈕繼續。



圖 5-20 Nuvoton NuMicro ICP Programming Tool

2. 將 NuMaker-NUC029SGE 板子上 Nu-Link2-Me 端的 USB 連接至電腦，在[連線狀態檢測]欄位按[連接]按鈕，連接 NuMaker-NUC029SGE 板子。

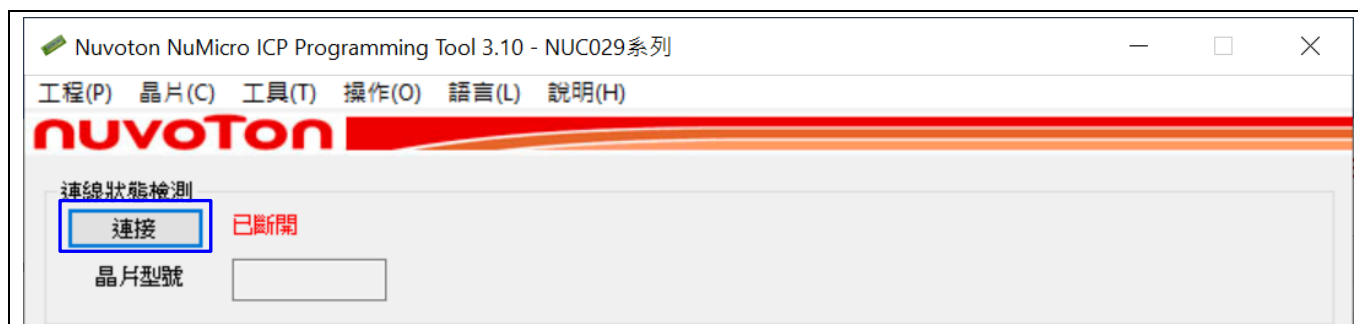


圖 5-21 連接 NuMaker-NUC029SGE 板子

- 連線成功後，在[載入文件]欄位選擇[LDROM]的韌體檔案為 bootloader 程式，[APROM]的韌體檔案為鍵盤程式。

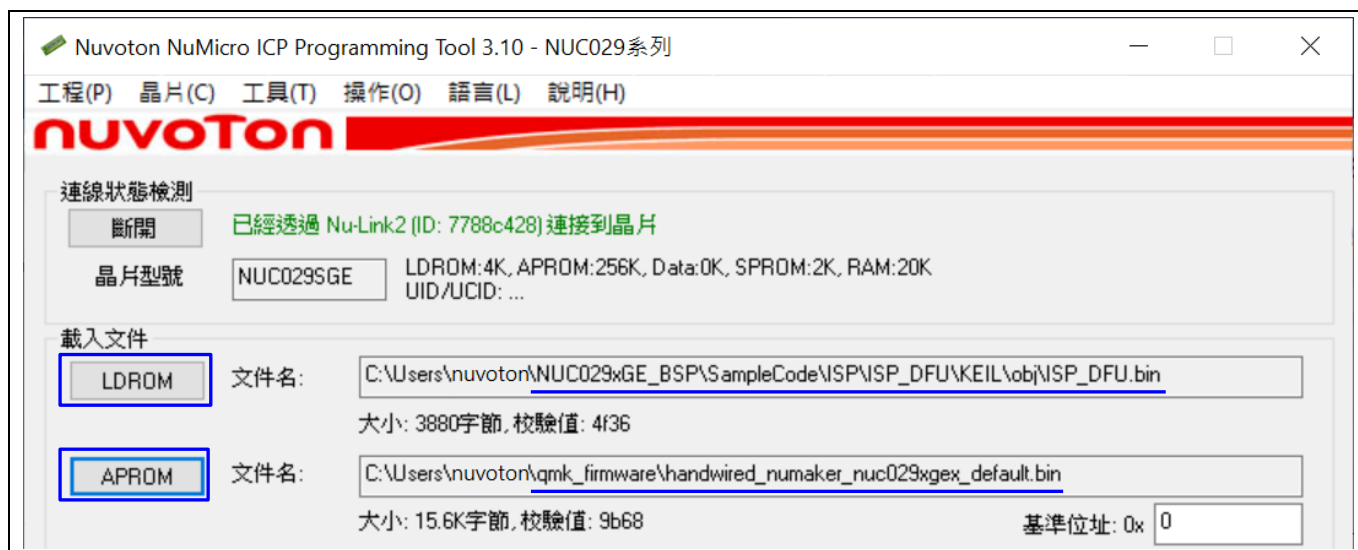


圖 5-22 選擇 LDROM 和 APROM 程式韌體檔案

- 在最底下的[編程]欄位，選擇要燒錄的區域包括[LDROM]和[APROM]，然後按[開始]按鈕，開始編程，最後顯示編程完成訊息。

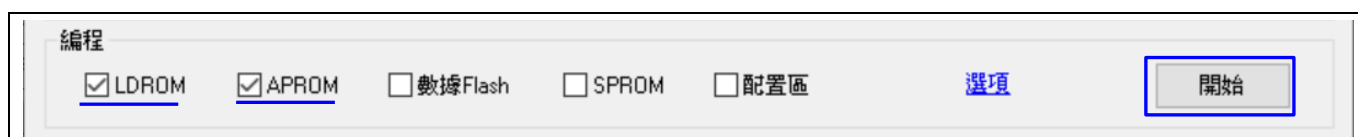


圖 5-23 開始編程

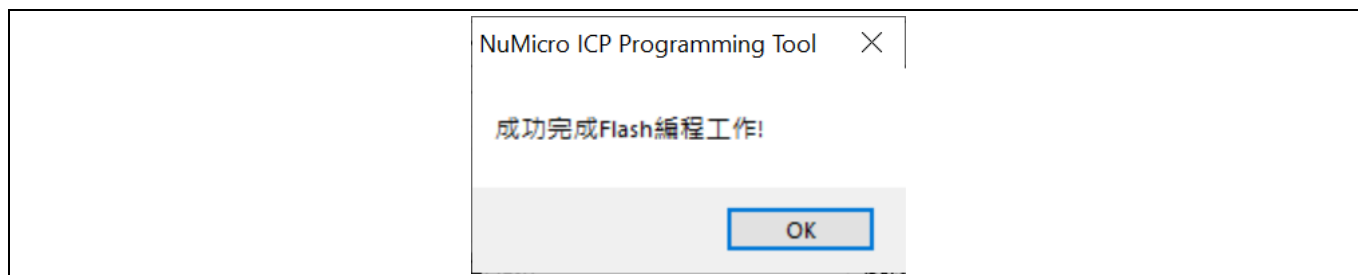


圖 5-24 編程完成

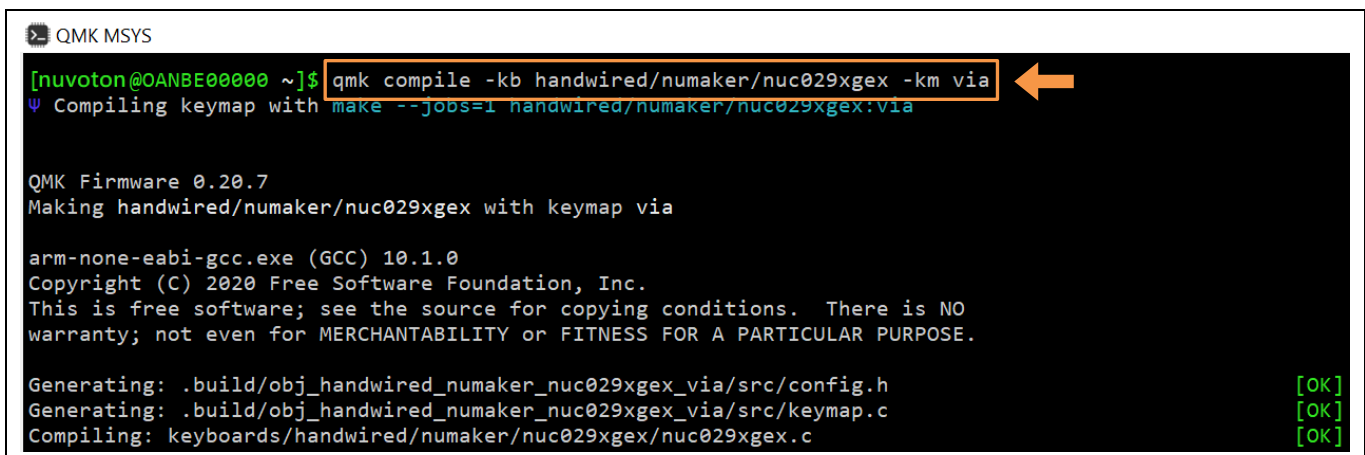
編程完成後，將 NuMaker-NUC029SGE 端的 USB 連接至電腦，會在電腦上識別為一個 USB HID 鍵盤設備，按下 NuTFT\_LCM 上的按鈕就可以模擬鍵盤上的按鍵。

## 5.4 韌體更新

QMK 提供 QMK Toolbox 軟體作為更新韌體的工具，以下將示範透過 QMK Toolbox (Nuvoton Edition) 更新鍵盤韌體為支援 VIA 功能的 QMK 鍵盤程式。

1. 開啟 QMK MSYS，執行 `qmk compile -kb handwired/numaker/nuc029xgex -km via` 命令，編譯支援 VIA 功能的 QMK 鍵盤程式，編譯完成後的韌體檔案會新增在 `qmk_firmware` 目錄底下。

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



```

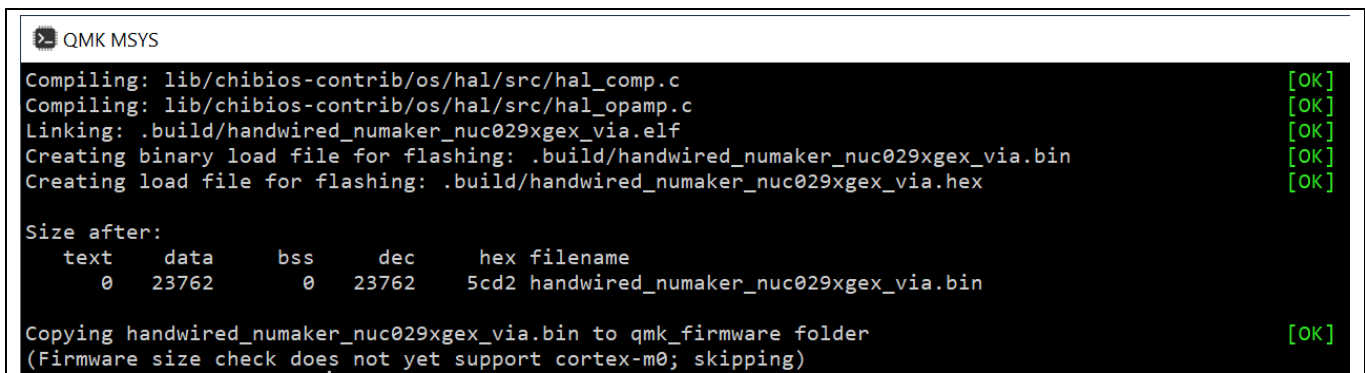
QMK MSYS
[nuvoton@OANBE00000 ~]$ qmk compile -kb handwired/numaker/nuc029xgex -km via
Ψ Compiling keymap with make --jobs=1 handwired/numaker/nuc029xgex:via

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]
  
```

圖 5-25 編譯支援 VIA 功能的鍵盤韌體



```

QMK MSYS
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)
  
```

圖 5-26 編譯支援 VIA 功能鍵盤韌體完成

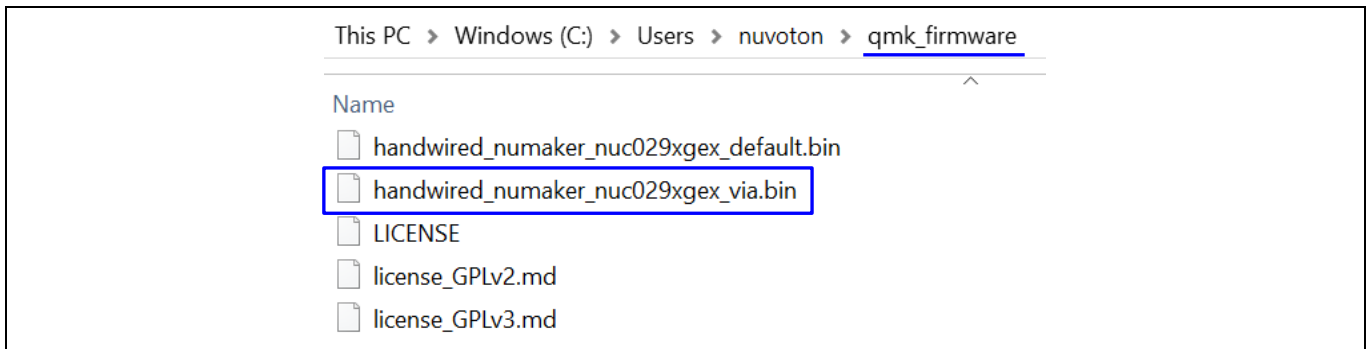


圖 5-27 支援 VIA 功能鍵盤韌體檔案

2. 將板子切換到 **bootloader** 模式：連接 NuMaker-NUC029SGE 端的 USB 至電腦，按住板子上的 SW1 按鈕，然後再按一下 RESET 按鈕。
3. 進入 bootloader 會被識別為一個 Nu\_DFU 裝置，識別到 Nu\_DFU 裝置後即可放開 SW1 按鈕。如果無法識別到 Nu\_DFU 裝置，則需要安裝驅動，檔案路徑如圖 5-29 所示。

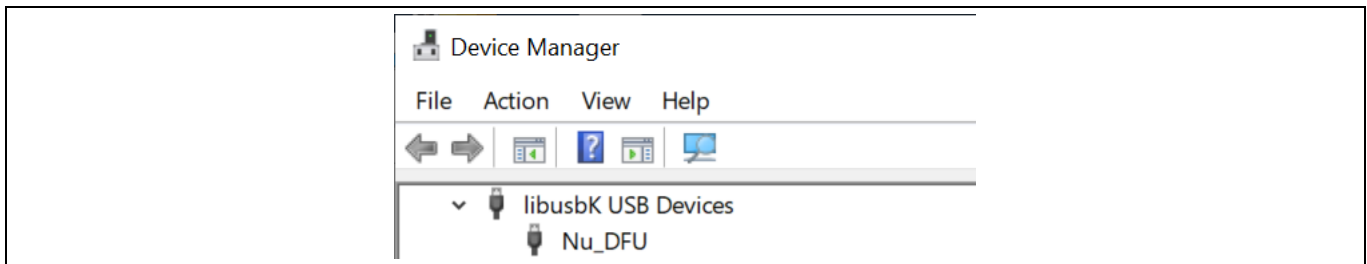


圖 5-28 Nu\_DFU 裝置

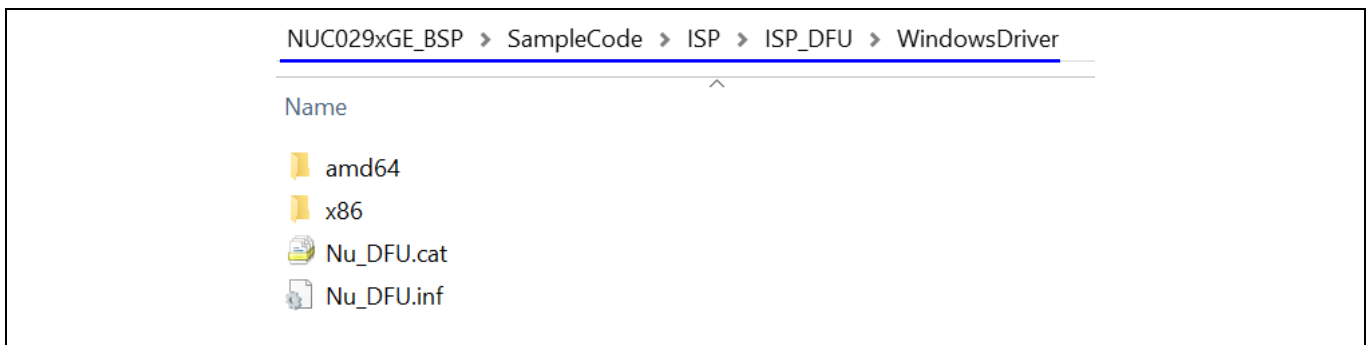


圖 5-29 Nu\_DFU 裝置驅動檔案路徑

4. 開啟本範例代碼裡的 **QMK Toolbox (Nuvoton Edition)**，檔案路徑如圖 5-30 所示。如果開啟時顯示如圖 5-31 訊息詢問是否要安裝裝置驅動，注意此時詢問安裝的驅動適用於其他支援 QMK 的裝置，但不包括我們現在要用的 Nu\_DFU 裝置，Nu\_DFU 裝置的驅動程式仍需要手動安裝，使用者在此可以自行決定是否要安裝其他裝置驅動。

EC\_NUC029\_QMK\_Keyboard\_V1.00

qmk\_tool

qmk\_toolbox - nuvoton\_edition.exe

圖 5-30 QMK Toolbox (Nuvoton Edition) 檔案路徑

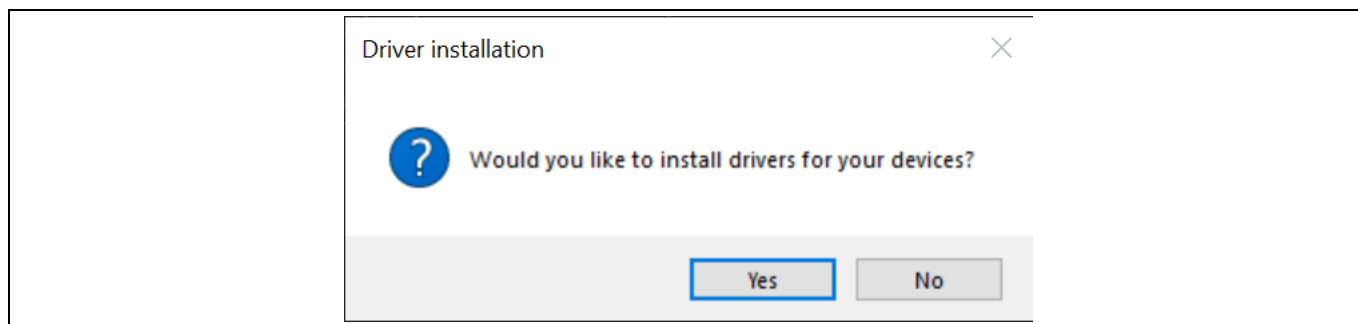


圖 5-31 詢問是否安裝裝置驅動(但不包括 Nu\_DFU 裝置)

5. QMK Toolbox (Nuvoton Edition)顯示偵測到 Nuvoton DFU Device 訊息。
6. 按[Open]按鈕選擇要更新的韌體檔案。
7. 按[Flash]按鈕開始進行韌體更新，最後顯示韌體更新完成訊息。

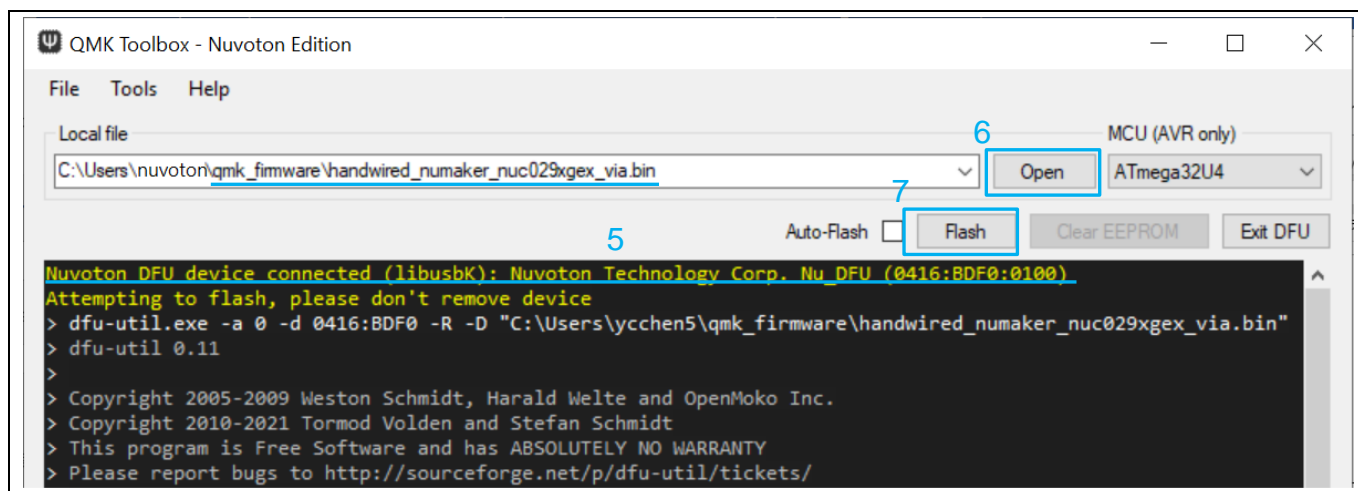


圖 5-32 QMK Toolbox (Nuvoton Edition)進行韌體更新

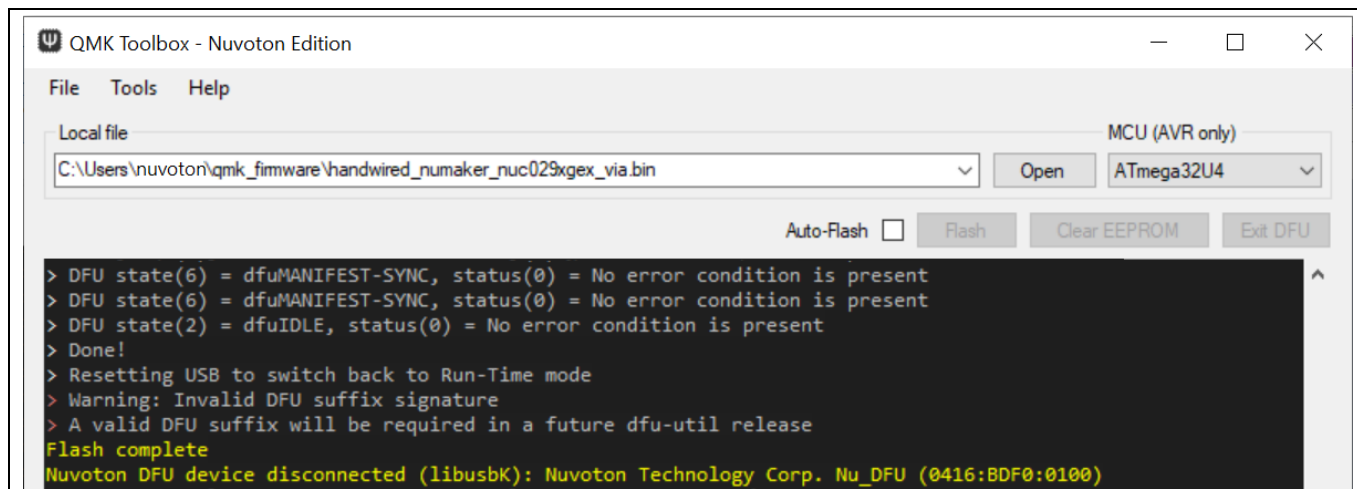


圖 5-33 QMK Toolbox (Nuvoton Edition) 進行韌體更新完成

## 5.5 VIA

[VIA](#) 是一款可以線上及時修改按鍵定義的工具，而不需要再重新編譯和更新韌體，使用時需搭配支援 VIA 功能的 QMK 鍵盤程式，注意 VIA 功能在預設的 QMK 鍵盤程式未被開啟。

1. 開啟 [VIA 網頁 App](#)，在上方選單選擇[SETTINGS]頁面。

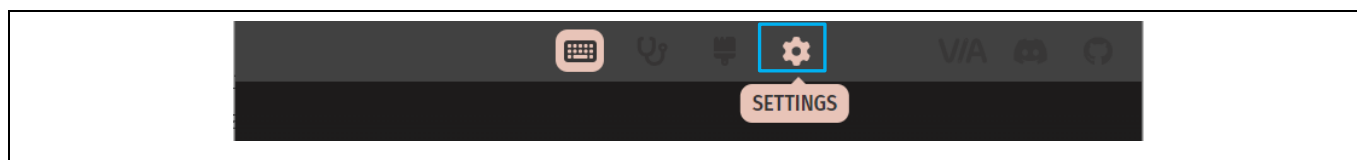


圖 5-34 VIA 網頁 App

2. 在 [SETTINGS]頁面如圖 5-35 所示，啟用[Show Design tab]功能，上方選單即會多出 [DESIGN]頁面。

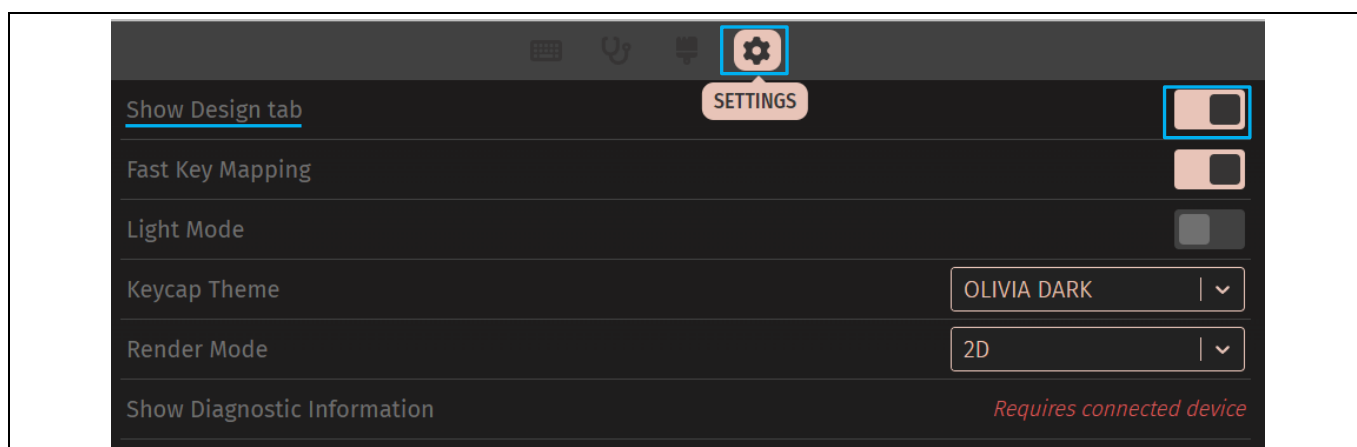


圖 5-35 VIA 工具 [SETTINGS]頁面

3. 在[DESIGN]頁面如圖 5-36 所示，關掉[Use V2 definitions (deprecated)]功能。
4. 按[Load Draft Definition]選項的[Load]按鈕，選擇本範例代碼中給 VIA 工具使用的 NUC029xGE JSON 檔案，檔案路徑如圖 5-37 所示。

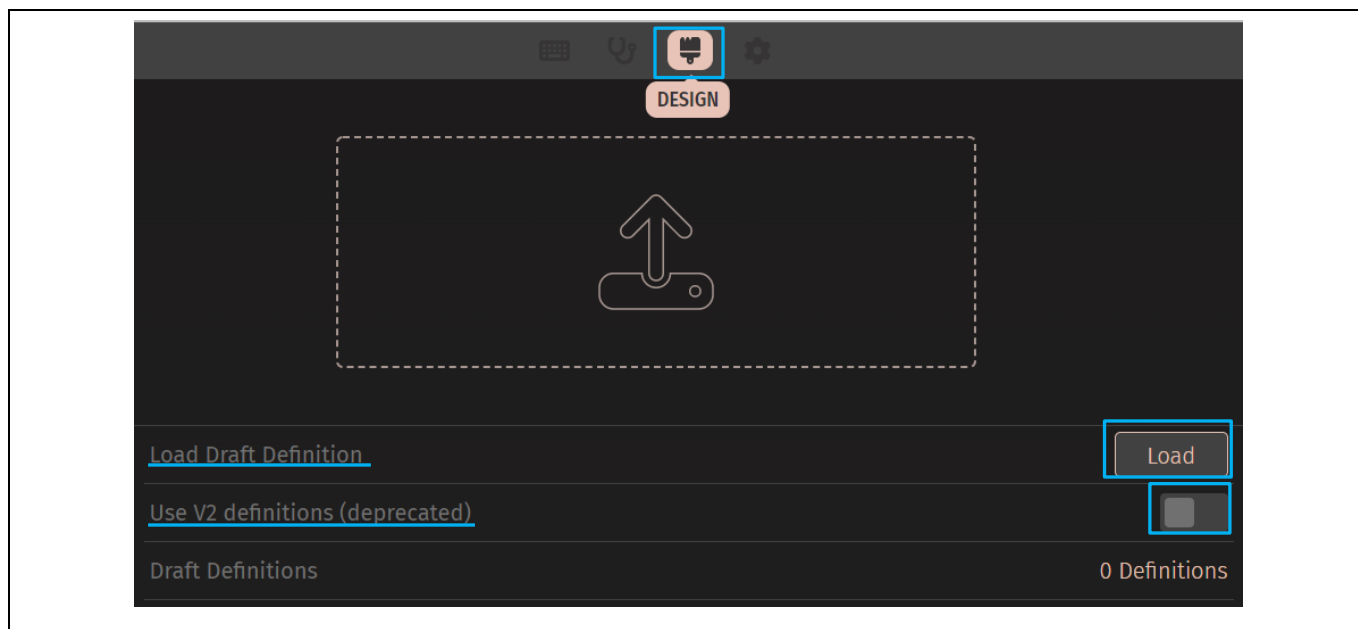


圖 5-36 VIA 工具 [DESIGN]頁面

```

└─ EC_NUC029_QMK_Keyboard_V1.00
    └─ via_json
        └─ NUC029xGE_QMK_VIA_V3.json
    
```

圖 5-37 給 VIA 工具使用的 NUC029xGE JSON 檔案路徑



- 加入 JSON 檔案後，在[DESIGN]頁面顯示 NUC029xGE 的設計有兩個按鍵。

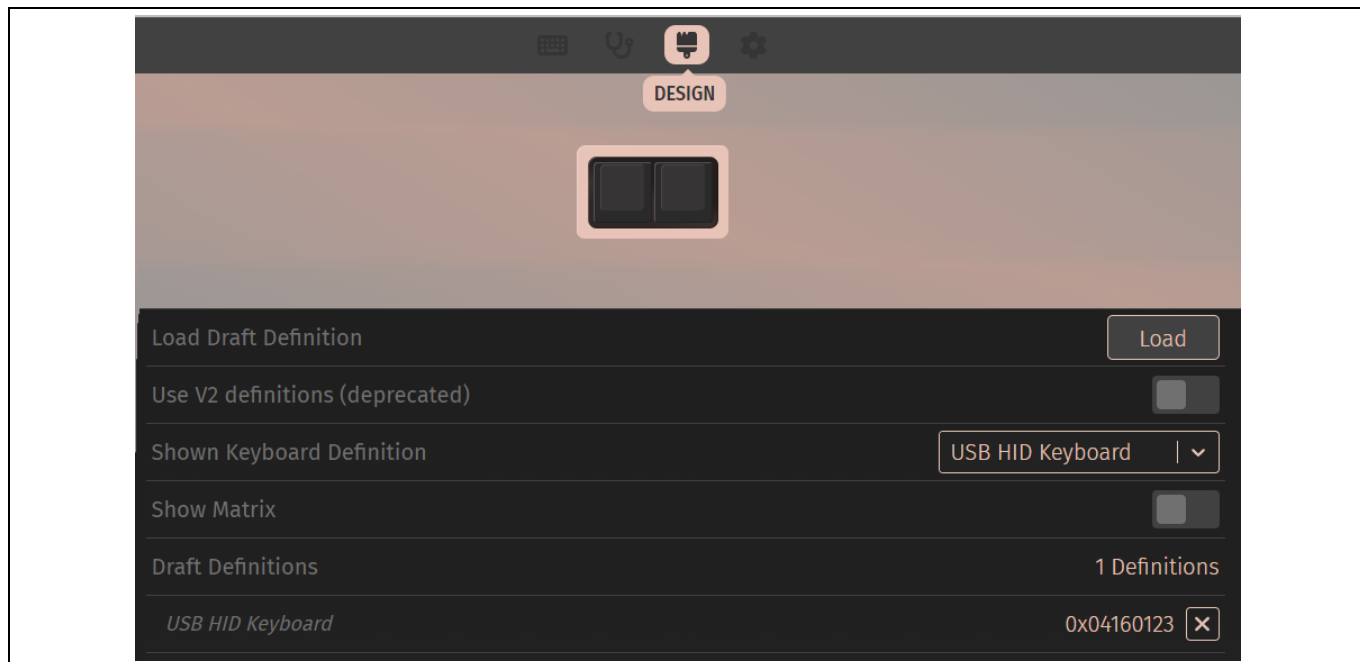


圖 5-38 NUC029xGE 的設計有兩個按鍵

- 如果有認到支援 VIA 工具的 HID 設備，會跳出要求連接該設備的視窗，選擇該設備後按 [Connect]連接。

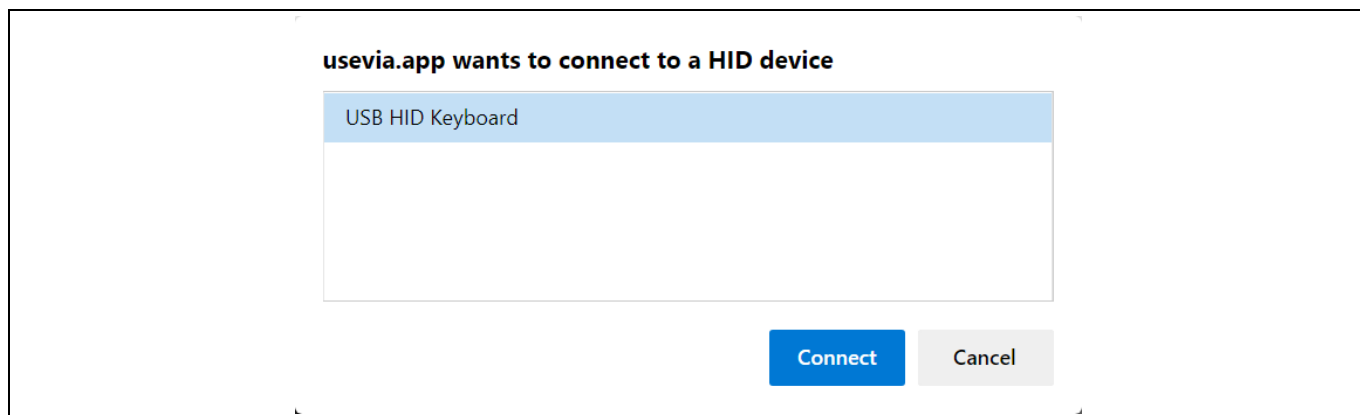


圖 5-39 連接 HID 設備

7. 如果沒有自動跳出連線視窗，也可以在[CONFIGURE]頁面上按[Authorize device +]按鈕嘗試手動連接。

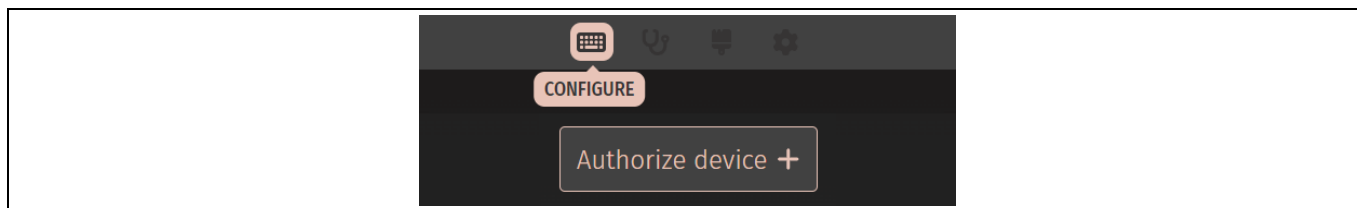


圖 5-40 按[Authorize device +]按鈕手動連接 HID 設備

8. 連接上 HID 設備之後，在[CONFIGURE]頁面顯示設計上的兩個按鍵，分別是 [ESC]和 [1]。



圖 5-41 按鍵[ESC]和[1]

9. 如果要修改按鍵定義，先點選要修改的按鍵例如按鍵[1]，當按鍵[1]被選中時會閃爍，接著再點選下方按鍵列表中想要更改的目標按鍵，例如按鍵[4]，如圖 5-42 所示。



圖 5-42 修改按鍵定義

10. 此時我們可以看到畫面顯示：第二顆按鍵已經從原本的按鍵[1]更改成按鍵[4]，如圖 5-43 所示。



圖 5-43 按鍵 [ESC]和[4]

11. VIA 工具也附有鍵盤測試工具，在上方選單中的[KEY TESTER]頁面就可以直接測試按鍵修改前和修改後的結果，如圖 5-44 所示，可以看到兩顆按鍵已經修改成[ESC]和[4]。



圖 5-44 VIA 鍵盤測試結果

## 6. 修訂紀錄

Date	Revision	Description
2023.05.23	1.00	初始發布。

### 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.