NUC980 簡易 OOB 模擬

NuMicro® 32 位元系列微控制器範例程式碼介紹

文件資訊

應用簡述	此範例示範如何使用 NUC980 建立一個簡單的遠端監控和控制應用程序，具備類似頻外（OOB, Out of Band）功能。
BSP 版本	NUC980 buildroot_2024
開發平台	NuMaker-RTU-NUC980 (Chili) 或 NuMaker-IIoT-NUC980 系列

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

本範例將說明如何運用 Nuvoton NuMaker NUC980 系列開發板，結合以 Buildroot 建構的嵌入式 Linux 作業系統，實作物聯網（IoT）應用。範例程式以 C 語言及 Paho MQTT 函式庫撰寫，使開發板能透過乙太網路與外部 MQTT broker 進行通訊，回報系統狀態與按鍵狀態，並可遠端控制板載 LED。

1.1 原理

當在 NuMaker NUC980 板上執行本範例時，會透過 MQTT 協定連接到 test.mosquitto.org（免費的 MQTT broker）。NUC980 會訂閱以下主題：

- board/status：顯示 NuMaker 板是否在線上或離線。
- board/leds/1：顯示板載 LED（於 NuMaker-IIoT-NUC980 系列為 LED3，於 NuMaker-RTU-NUC980/Chili 為 LED1）的狀態。
- board/buttons/1：反映板上 K1 按鍵的按下或釋放狀態。
- board/temperature/1：回報 MAX31875 感測器的溫度資料。

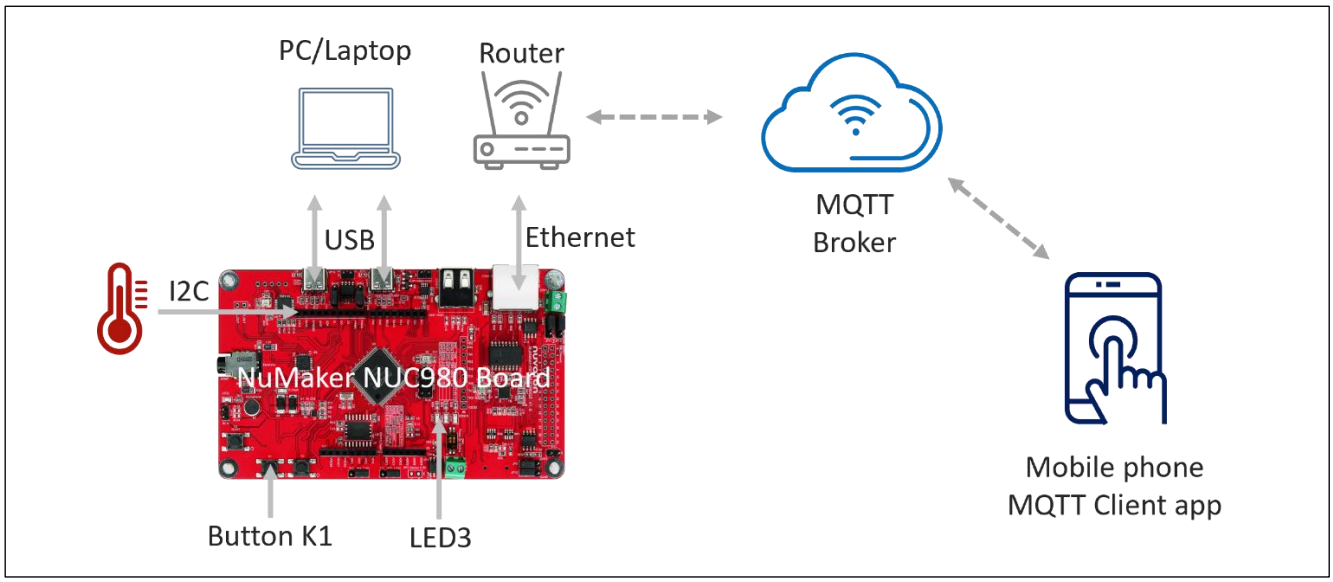


圖 1-1 本範例的架構圖

1.2 先決條件

請先詳閱「軟體與硬體需求」章節，準備所有所需元件。接著，依照「範例程式執行」章節的步驟，完成範例程式的建置和燒錄至開發板，並設定 MQTT 應用程式。

範例 ZIP 檔中已包含預建檔案，詳細說明請參考「目錄資訊」章節。

1.3 執行結果

完成將韌體燒錄至 NuMaker NUC980 開發板並設定好 MQTT 應用程式後，即可依照下列步驟執行本範例。

- 重新啟動 NuMaker NUC980 板，使用「root」帳號登入。
- 啟用乙太網路並由 DHCP 伺服器取得 IP 位址

```
# udhcpc
```

- 執行 rcdemo

```
# ./rcdemo
```

- NuMaker NUC980 板會連接至 MQTT broker，訂閱主題並發佈自身狀態。

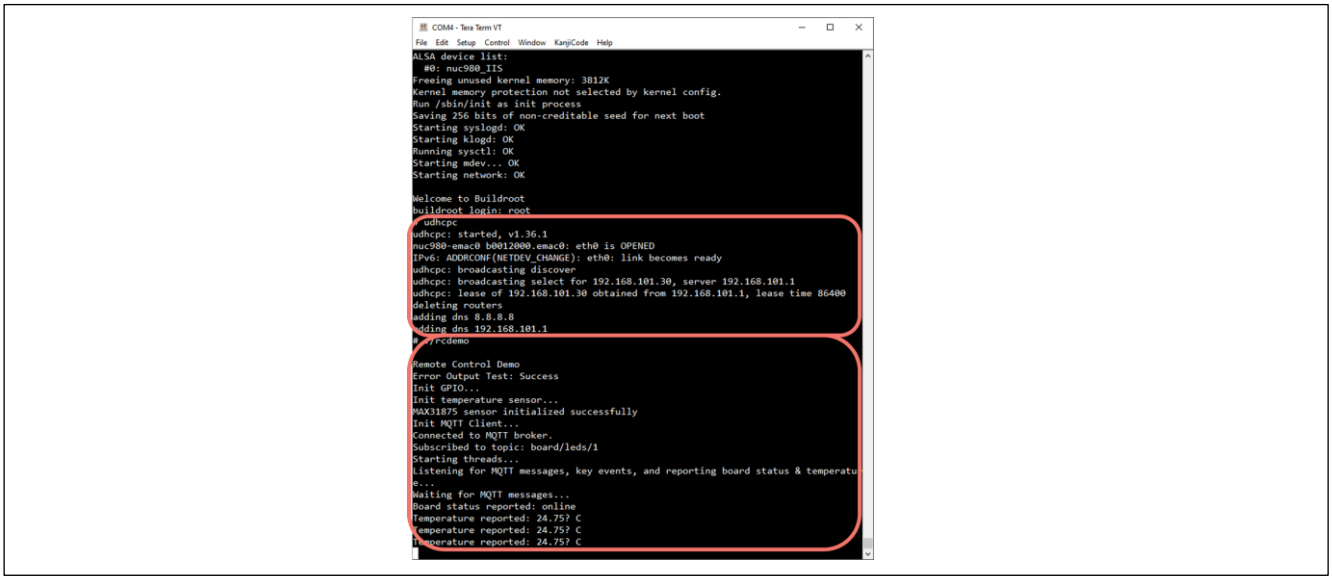


圖 1-2 NUC980 終端機顯示訊息

- 在 Android 手機上開啟 MQTT 應用程式。
- 確認應用程式已連接至 MQTT broker，雲端圖示應為橙色，若否請點擊重新連線。
- 當 NuMaker 板成功連線並執行範例時，**Board Status** 顯示為 **online**。
- 嘗試下列步驟進行操作：

  - 按下並持續壓住 NuMaker NUC980 開發板上的 **K1** 按鍵，應用程式中的 **Button status** 會顯示為 **pressed**，放開後則顯示為 **released**。
  - 在應用程式中切換 **LED 1** 開關，即可遠端控制開發板上的 **LED**。
  - 應用程式會顯示溫度感測器的歷史紀錄。

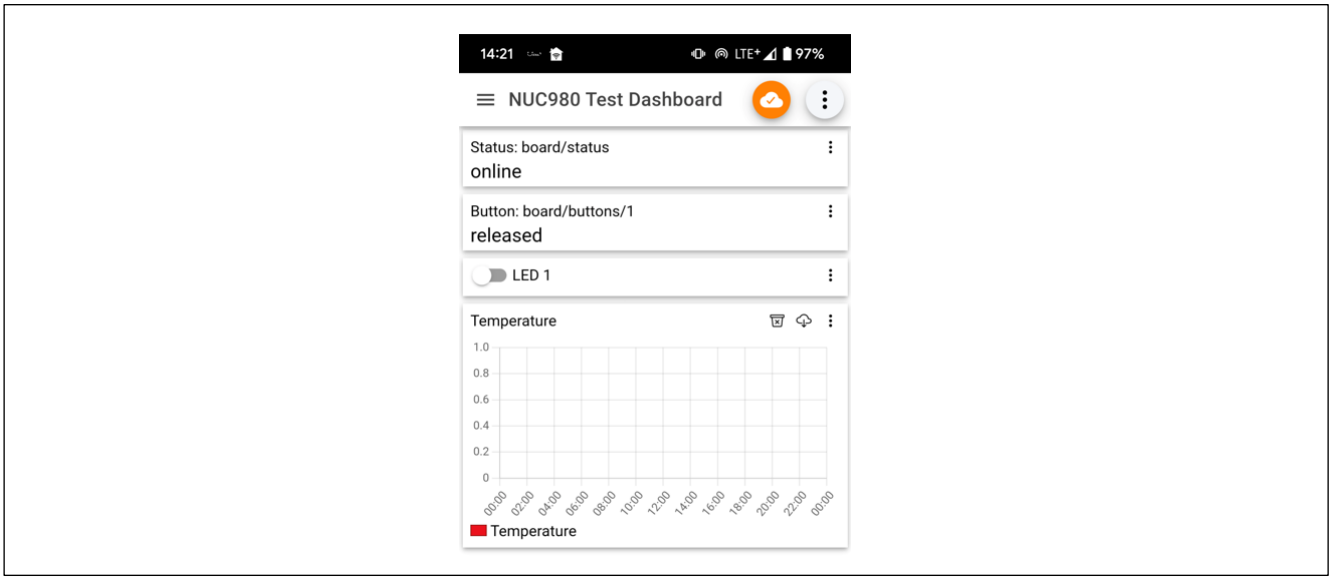


圖 1-3 MQTT 應用程式顯示畫面

2. 程式碼介紹

本範例的所有程式碼皆集中於 rcdemo.c 檔案。NUMAKER_BOARD 巨集用於指定 LED、按鍵 I/O 及 I²C 埠所對應的開發板型號。

```
#define NUC980_IOT      1      // For NuMaker-IIoT-NUC980/G1/G2 board
#define NUC980_CHILI    2      // For NuMaker-RTU-NUC980 (Chili) board

#ifndef NUMAKER_BOARD
#define NUMAKER_BOARD NUC980_IOT    // Default to IoT series board if not defined.
#endif
```

MQTT broker 的網址與埠號設定如下：

```
#define ADDRESS     "tcp://test.mosquitto.org:1883"
#define CLIENTID    "NUC980Client"
#define QOS         1
#define TIMEOUT     10000L
```

範例程式會建立多個執行緒分別處理按鍵（key_monitor_thread）、板子狀態（board_status_thread）、溫度（temperature_thread）及 LED（mqtt_client_thread）。

```
    // Start threads
    printf("Starting threads...\n");
    pthread_create(&key_thread, NULL, key_monitor_thread, NULL);
    pthread_create(&status_thread, NULL, board_status_thread, NULL);
    pthread_create(&temp_thread, NULL, temperature_thread, NULL);
    pthread_create(&mqtt_thread, NULL, mqtt_client_thread, NULL);
```

Paho MQTT 函式庫可協助開發者輕鬆建立 MQTT 用戶端並訂閱相關主題。

```
    // Initialize MQTT client
    printf("Init MQTT Client...\n");
    MQTTClient_create(&client, ADDRESS, CLIENTID, MQTTCLIENT_PERSISTENCE_NONE, NULL);
    conn_opts.keepAliveInterval = 20;
    conn_opts.cleansession = 1;

    if ((rc = MQTTClient_connect(client, &conn_opts)) != MQTTCLIENT_SUCCESS)
    {
        printf("Failed to connect to MQTT broker, return code %d\n", rc);
        cleanup_gpio();
        exit(EXIT_FAILURE);
    }
    printf("Connected to MQTT broker.\n");
```

```
if ((rc = MQTTClient_subscribe(client, TOPIC_LED, QOS)) != MQTTCLIENT_SUCCESS)
{
    printf("Failed to subscribe topic %s, return code %d\n", TOPIC_LED, rc);
}
else
{
    printf("Subscribed to topic: %s\n", TOPIC_LED);
}
```

3. 軟體與硬體需求

3.1 軟體需求

- 執行 Ubuntu 的 PC

  - 可為實體機或虛擬機（VM）。

- Buildroot

  - 下載網址：https://github.com/OpenNuvoton/buildroot_2024 。

- PC 終端機軟體

  - 如 Putty、Realterm、TeraTerm 等。

- MQTT 應用程式

  - IoT MQTT Panel (可於 Google Play 下載)。

  - 或其他 MQTT 客戶端應用程式。

3.2 硬體需求

- 以下任一片 NuMaker NUC980 開發板

  - NuMaker-IIoT-NUC980 系列，包含：

    - NuMaker-IIoT-NUC980
    - NuMaker-IIoT-NUC980G1
    - NuMaker-IIoT-NUC980G2

  - NuMaker-RTU-NUC980 (另稱 Chili)

- 一條 LAN 網路線

- 兩條 USB 傳輸線

  - MicroUSB 適用於 NuMaker-IIoT-NUC980 或 Chili。

  - Type-C 適用於 NuMaker-IIoT-NUC980G1/G2。

- 一台具有 LAN 埠的 Wi-Fi 路由器

- Thermo 6 Click 擴充板 (內含 MAX31875 溫度感測器)

請依連接方式完成範例環境的建置。

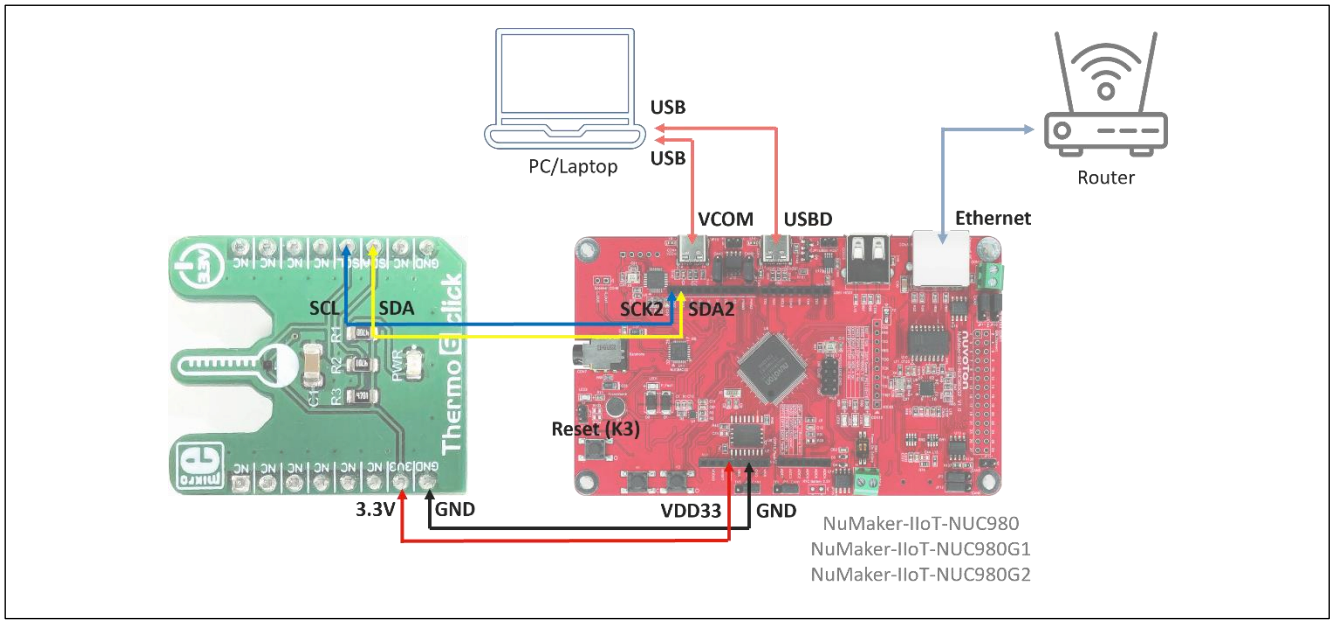


圖 3-1 NuMaker-IIoT-NUC980 系列開發板連接圖

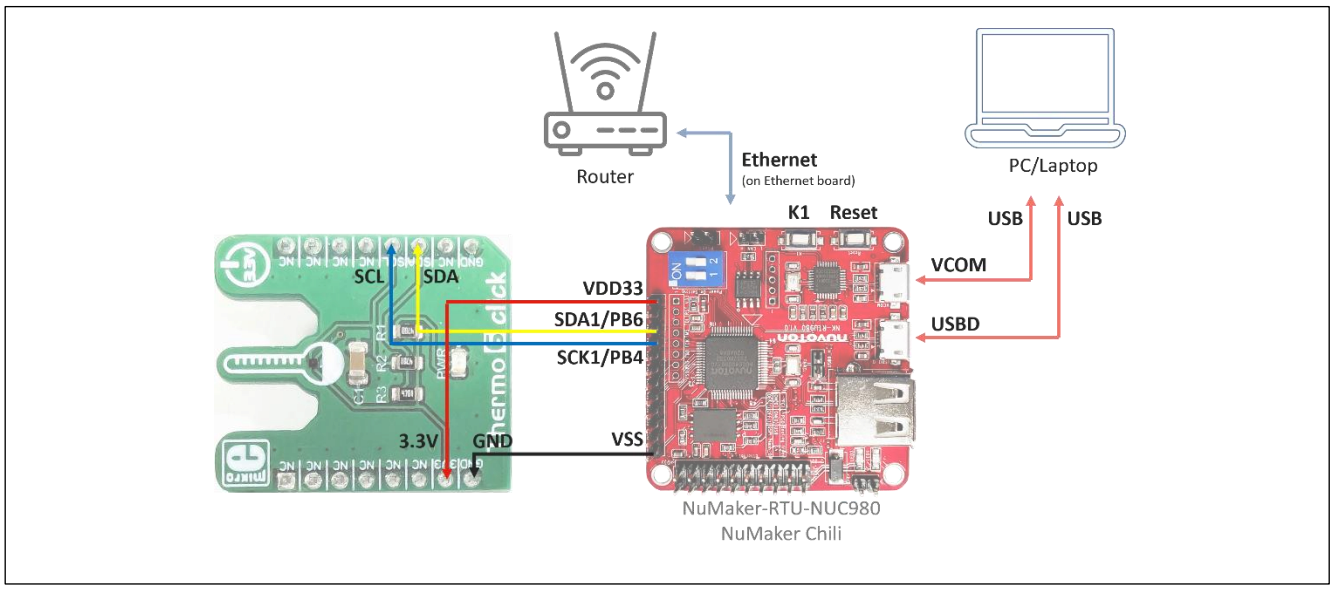


圖 3-2 NuMaker-RTU-NUC980 (Chili) 開發板連接圖

4. 目錄資訊

目錄結構如下圖所示。

EC_NUC980_Simple_OOB_Simulation_V1.00	
numaker-nuc980-iot	NuMaker-IIoT-NU980 系列開發板的 U-boot、system image 和 device tree
numaker-nuc980-rtu(chili)	NuMaker-RTU-NU980 (Chili) 開發板的 U-boot、system image 和 device tree
env.txt	環境設定檔
rcdemo.c	範例原始碼

圖 4-1 目錄資訊

5. 範例程式執行

請依照下列步驟，使用 NUC980 Buildroot 建立並設定開發環境，並完成範例程式的建置與部署。

5.1 韌體建置

- 在 Ubuntu 裡安裝 Buildroot

  - 下載 NUC980 用的 buildroot_2024。

```
$ git clone https://github.com/OpenNuvoton/buildroot_2024
$ cd buildroot_2024
```

  - 為 NuMaker NUC980 開發板設定 Buildroot. 針對 NuMaker-IIoT-NUC980/G1。

```
$ make nuvoton_nuc980_iot_defconfig
```

    針對 NuMaker-IIoT-NUC980G2。

```
$ make nuvoton_nuc980_iot_g2_defconfig
```

    針對 NuMaker-RTU-NUC980 (Chili) 。

```
$ make nuvoton_nuc980_chili_defconfig
```

  - 設定 Buildroot 加入相關的套件。

```
$ make menuconfig
```

  - 勾選 Paho MQTT 函式庫。

```
Target packages ➔ Libraries ➔ Networking ➔ [*] paho-mqtt-c
```

  - 然後按 <Save> 儲存, 再按 <Exit> 離開設定畫面。

  - 設定 Linux kernel 支援 NUC980 按鍵事件。

```
$ make linux-menuconfig
```

```
Device Drivers ➔ Input device support ➔ <*> Event interface
Device Drivers ➔ Input device support ➔ [*] keyboards ➔ [*] Nuvoton keys
support
```

- 然後按 <Save> 儲存, 再按 <Exit> 離開 Linux kernel 設定。

- 建構系統映像檔。

```
$ make
```

- 僅在 **NuMaker-RTU-NUC980 (Chili)** 需修改 device tree 以啟用 J2 上的 I²C。

```
$ vi output/build/linux-custom/arch/arm/boot/dts/nuc980-chili.dts
```

```
    i2c1: i2c1@b0081000 {
            status = “okay”;
            pinctrl-0 = <&pinctrl_i2c1_PB>;
    };
```

- 範例 ZIP 檔已包含 rcdemo.c 原始碼，請直接複製至 Buildroot 目錄。

```
$ cd ..
$ cp your-directory-of-EC_NUC980_Simple_OOB_simulation_V1.00/rcdemo.c .
```

- 設定 cross-compiler 環境。

```
$ source buildroot_2024/output/host/environment-setup
```

- 編譯範例程式 rcdemo.c（針對 **NuMaker-IIoT-NUC980/G1/G2**）。

```
$ $CC rcdemo.c -o rcdemo -Ibuildroot_2024/output/staging/usr/include \
  -Lbuildroot_2024/output/staging/usr/lib -Wl,-Bstatic -lpaho-mqtt3c -Wl,-Bdynamic
```

- 編譯範例程式 rcdemo.c（針對 **NuMaker-RTU-NUC980 (Chili)**）。

```
$ $CC rcdemo.c -o rcdemo -Ibuildroot_2024/output/staging/usr/include \
  -DNUMAKER_BOARD=2 -Lbuildroot_2024/output/staging/usr/lib -Wl,-Bstatic \
  -lpaho-mqtt3c -Wl,-Bdynamic
```

- 將執行檔複製到 root 的 home directory。

```
$ cp rcdemo buildroot_2024/output/target/root
```

- 重新建置映像檔以納入 rcdemo 執行檔。

```
$ cd buildroot_2024
$ make
```

- 建置完成後，請從 Buildroot 子目錄中取得下列檔案：

Files	IIoT-NUC980 IIoT-NUC980G1	IIoT-NUC980G2	RTU-NUC980 (Chili)
./output/images/**u-boot.bin**	●	●	●
./output/build/uboot-custom/spl/**u-boot-spl.bin**	●	●	
./output/images/**nuc980-iot-v1.0.dtb**	●		
./output/images/**nuc980-iot-g2-v1.0.dtb**		●	
./output/images/**nuc980-chili.dtb**			●
./output/images/**uImage**	●	●	●

- 接著用純文字編輯器建立 env.txt 檔案，內容如下。env.txt 亦包含於範例 ZIP 檔中。

```
baudrate=115200
bootdelay=1
stderr=serial
stdin=serial
stdout=serial
setspi=sf probe 0 30000000
loadkernel=sf read 0x7fc0 0x200000 0xC00000
loaddtb=sf read 0x1800000 0xE00000 0x10000
bootcmd=run setspi;run loadkernel;run loaddtb;bootm 0x7fc0 - 0x1800000
```

5.2 燒錄韌體至開發板

- 下載 **NuWriter** 工具以燒錄韌體到開發板

  - 下載網址 https://github.com/OpenNuvoton/NUC980_NuWriter/

  - NuWriter 說明文件於「.\」目錄下。

  - NuWriter Windows 驅動程式於「.\NuWriter\Driver」目錄，若未安裝請先安裝。

  - NuWriter 工具於「.\NuWriter\NuWriter\Release」目錄，無需安裝。

- 開啟開發板終端機 (Console)

  - 在 PC 或筆電上使用任一終端機軟體（如 TeraTerm、Putty）開啟 NUC980 VCOM，裝置名稱為「USB Serial Device」。

  - 設定 Baud rate 為 115200, non-parity, 8-bit, one stop bit。

- **NuMaker-IIoT-NUC980 series** 系列開發板燒錄韌體

  - 將 **SW1** 的 **No.1** 和 **No.2** 皆撥至 **ON** 以從 USB 開機。

  - 按下 **Reset** (**K3**) 鍵由 USB 開機。

  - 使用 **NuWriter** 依下表燒錄下列檔案。

  - DDR 參數: 依據開發板上 NUC980 型號選取正確的 **NUC980DK61Y.ini / NUC980DK63YC.ini / NUC980DK71YC.ini**。

  - Type 選取 **SPI NAND**。

Files	Type	Address
u-boot-spl.bin	Loader	0x200
u-boot.bin	Data	0x100000
uImage	Data	0x200000
nuc980-iot-v1.0.dtb (NUC980/G1) nuc980-iot-g2-v1.0.dtb (NUC980G2)	Data	0xE00000
env.txt	Environment	0x80000

- **NuMaker-RTU-NUC980 (Chili)** 開發板燒錄韌體

  - 將 **SW1** 的 **No.1** 和 **No.2** 皆撥至 **ON** 以從 USB 開機。

  - 按下 **Reset** 鍵由 USB 開機。

  - 使用 **NuWriter** 依下表燒錄下列檔案。

  - DDR 參數: 依據開發板上 NUC980 型號選取正確的 **NUC980DR61YC.ini / NUC980DR63YC.ini**。

  - Type 選取 **SPI NOR**。

Files	Type	Address
u-boot.bin	Loader	0xE00000
uImage	Data	0x200000
nuc980-chilli.dtb	Data	0xE00000
env.txt	Environment	0x80000

- 將 NuMaker NUC980 開發板設為 SPI Flash 開機:

  - 將 **SW1** 的 **No.1** 和 **No.2** 皆撥至 **OFF**。

  - 按下 **Reset** 鍵由 SPI Flash 開機。

- 使用「root」帳號登入。

5.3 設定 MQTT 應用程式

- 在 Android 手機上啟動 IoT MQTT Panel 應用程式。
- 新增連線並輸入 MQTT broker 資訊：

  - **Connection name**：自訂名稱，如 **NUC980-MQTT-Test**
  - **Client ID**：留空，會自動產生。
  - Broker Web/IP address：填入 test.mosquitto.org
  - Port：1883
  - Network protocol：TCP
  - 點選「+」新增儀表板，命名如 **NUC980 Test Dashboard**
  - 然後點選 **CREATE**

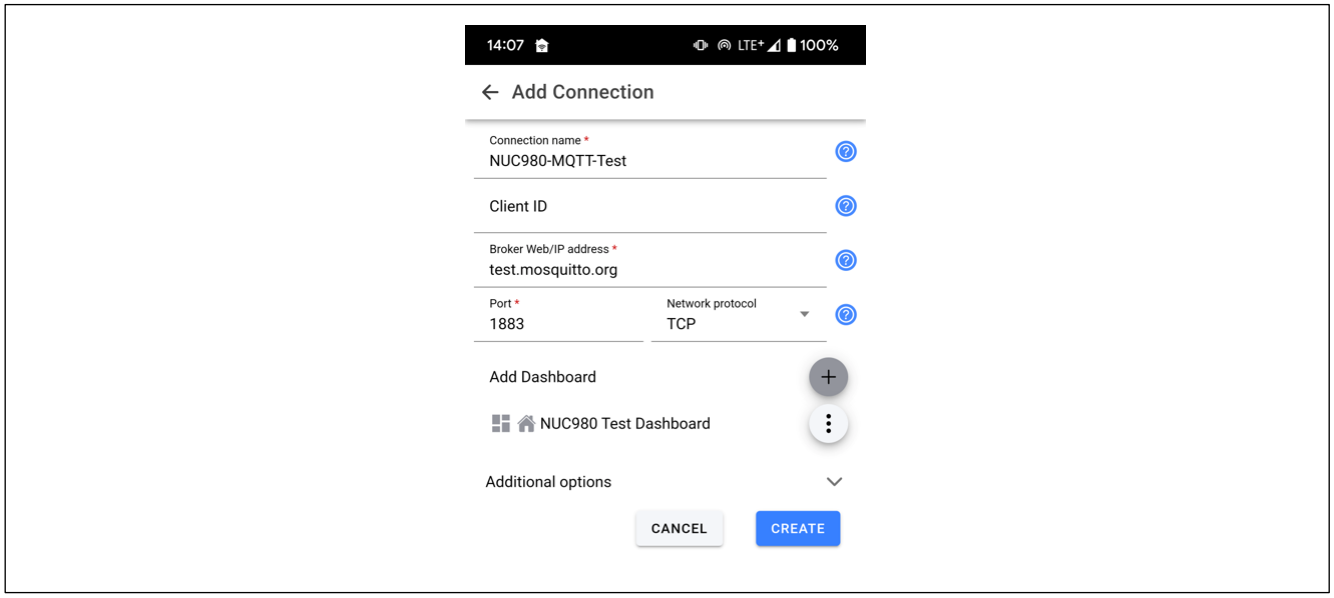


圖 5-1 MQTT 應用程式建立連線

- 回到主畫面，點選連線名稱（如 NUC980-MQTT-Test）新增面板。
- 新增四個面板。

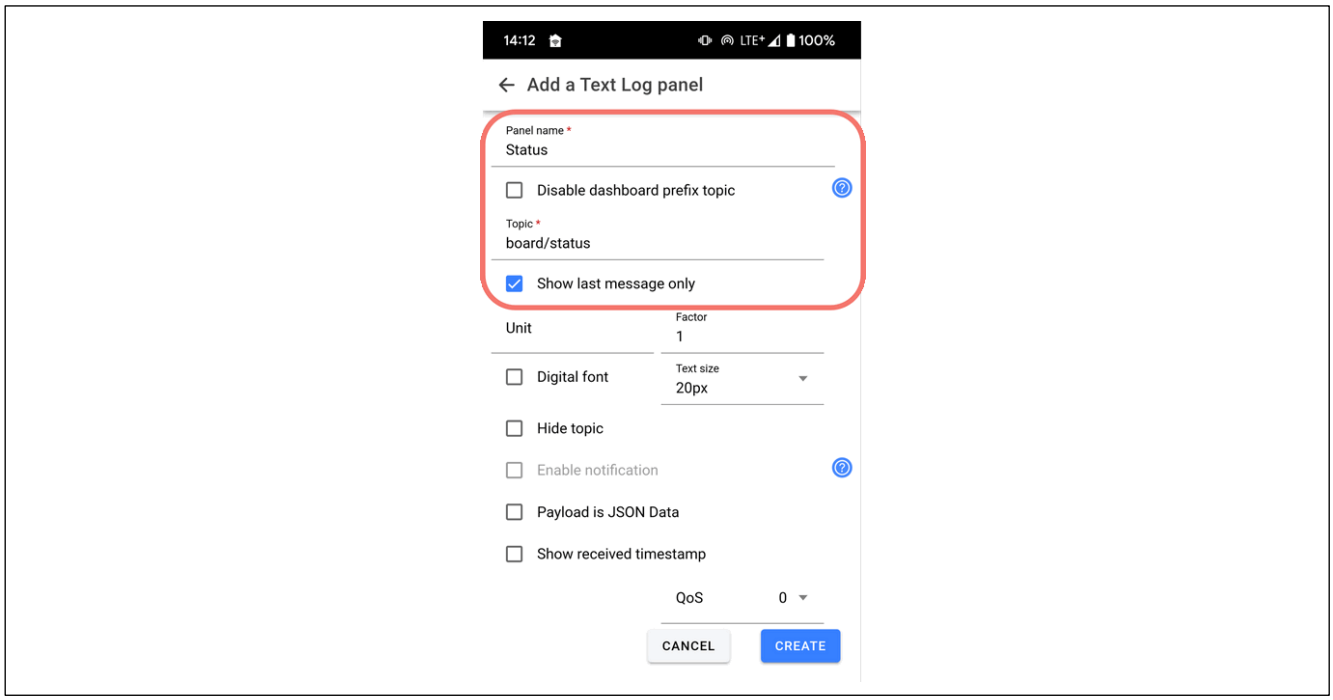


圖 5-2 MQTT 應用程式建立第一個面板

- 第一個面板

  - ♦ 選擇 Text Log
  - ♦ Panel name：Status
  - ♦ Topic：board/status
  - ♦ 勾選 Show last message only
  - ♦ 點選 CREATE

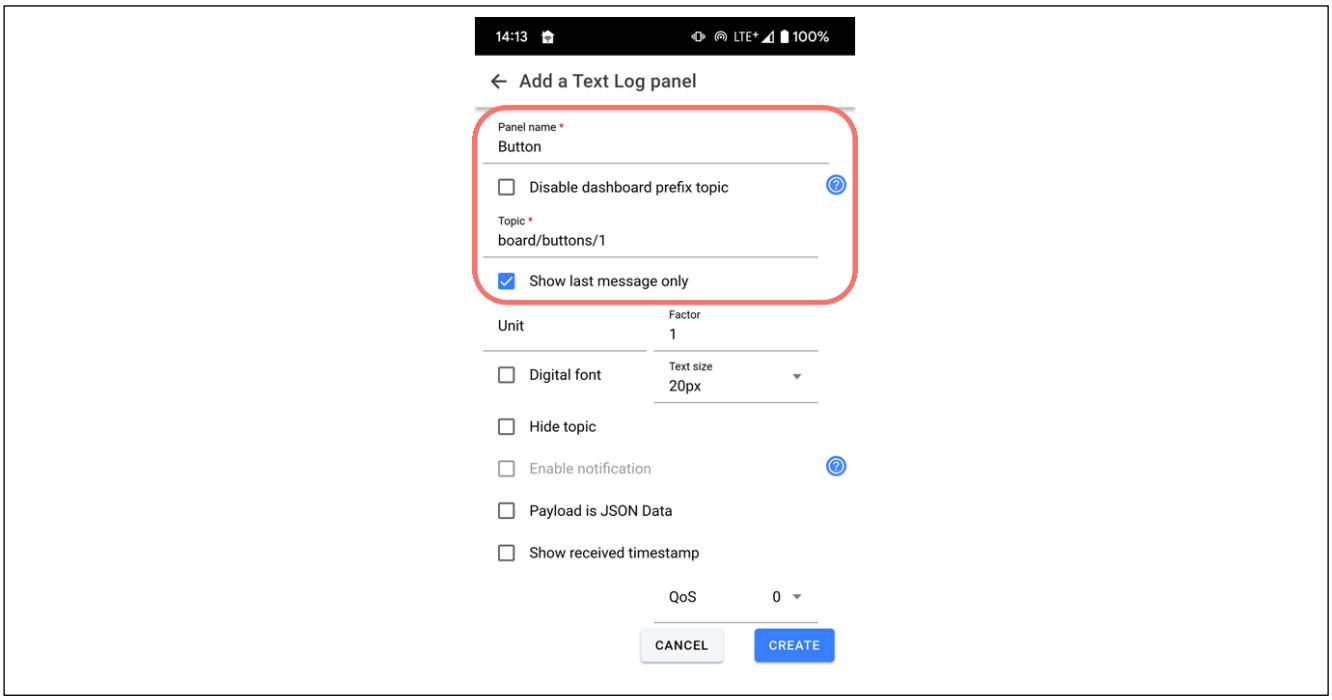


圖 5-3 MQTT 應用程式建立第二個面板

- 第二個面板

  - ♦ 選擇 Text Log
  - ♦ Panel name：Button
  - ♦ Topic：board/buttons/1
  - ♦ 勾選 Show last message only
  - ♦ 點選 CREATE

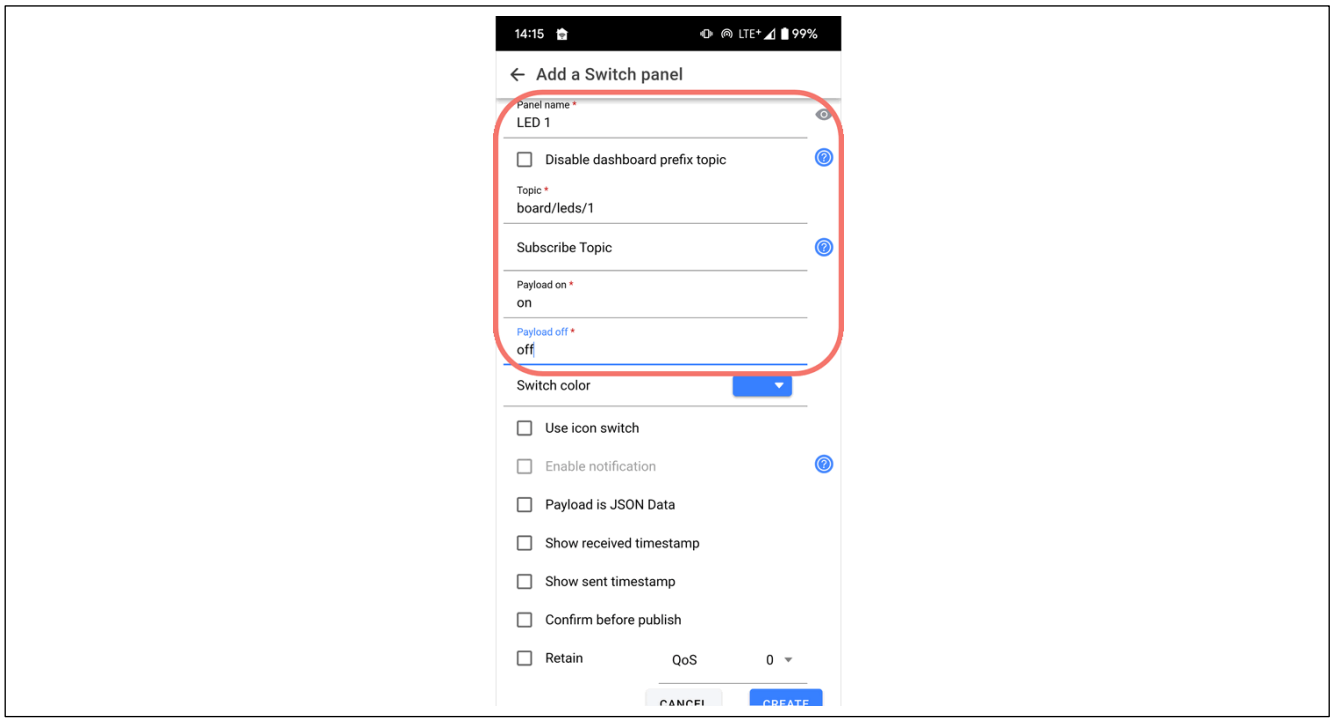


圖 5-4 MQTT 應用程式建立第三個面板

- 第三個面板

  - ♦ 選擇 Switch
  - ♦ Panel name：LED 1
  - ♦ Topic：board/leds/1
  - ♦ Payload on：on
  - ♦ Payload off：off
  - ♦ 點選 CREATE

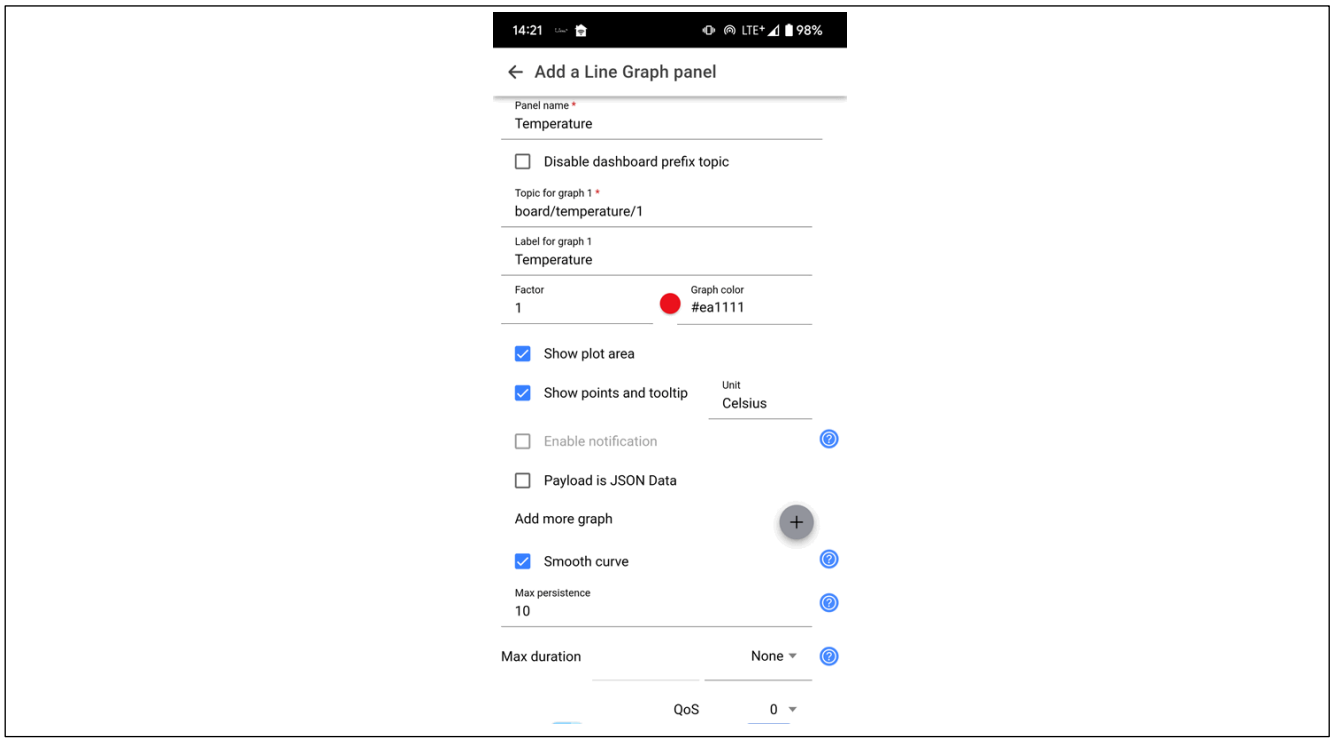


圖 5-5 MQTT 應用程式建立第四個面板

- 第四個面板

- ♦ 選擇 Line Graph
- ♦ Panel name：Temperature
- ♦ Topic：board/temperature/1
- ♦ Label for graph 1：Temperature
- ♦ Factor：1
- ♦ Graph color：#ea1111
- ♦ 勾選 Show plot area，Show points and tooltip 和 Smooth curve
- ♦ Unit：Celsius
- ♦ Max persistence：10
- ♦ 點選 CREATE

6. 結尾

本範例說明如何利用 Nuvoton NuMaker NUC980 系列開發板於嵌入式 Linux 環境下，結合 MQTT 協定與行動 IoT 控制面板應用程式，實現遠端監控與控制功能。

利用 C 語言及 Paho MQTT 函式庫實作，完整示範如何遠端控制開發板上的周邊（如 LED 與按鍵），並即時監控環境資料（如溫度）。

展現硬體、軟體與雲端服務的無縫整合，提供 IoT 遠端控制應用的實用解決方案，適合作為教學與實務應用之參考範例。

7. 修訂紀錄

Date	Revision	Description
2025.06.21	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.