NUC980 简易 OOB 模拟

NuMicro® 32 位系列微控制器范例程序代码介绍

文件信息

应用简述	此范例示范如何使用 NUC980 建立一个简单的远程监控和控制应用程序，具备类似带外（OOB, Out of Band）功能。
BSP 版本	NUC980 buildroot_2024
开发平台	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

本范例将说明如何运用 Nuvoton NuMaker NUC980 系列开发板，结合以 Buildroot 建构的嵌入式 Linux 操作系统，实现物联网（IoT）应用。范例程序以 C 语言及 Paho MQTT 函数库撰写，使开发板能透过以太网络与外部 MQTT broker 进行通讯，回报系统状态与按键状态，并可远程控制板载 LED。

1.1 原理

当在 NuMaker NUC980 板上执行本范例时，会透过 MQTT 协议连接到 test.mosquitto.org（免费的 MQTT broker）。NUC980 会订阅以下主题：

- board/status：显示 NuMaker 板是否在在线或脱机。
- board/leds/1：显示板载 LED（于 NuMaker-IIoT-NUC980 系列为 LED3，于 NuMaker-RTU-NUC980/Chili 为 LED1）的状态。
- board/buttons/1：反映板上 K1 按键的按下或释放状态。
- board/temperature/1：回报 MAX31875 传感器的温度数据。

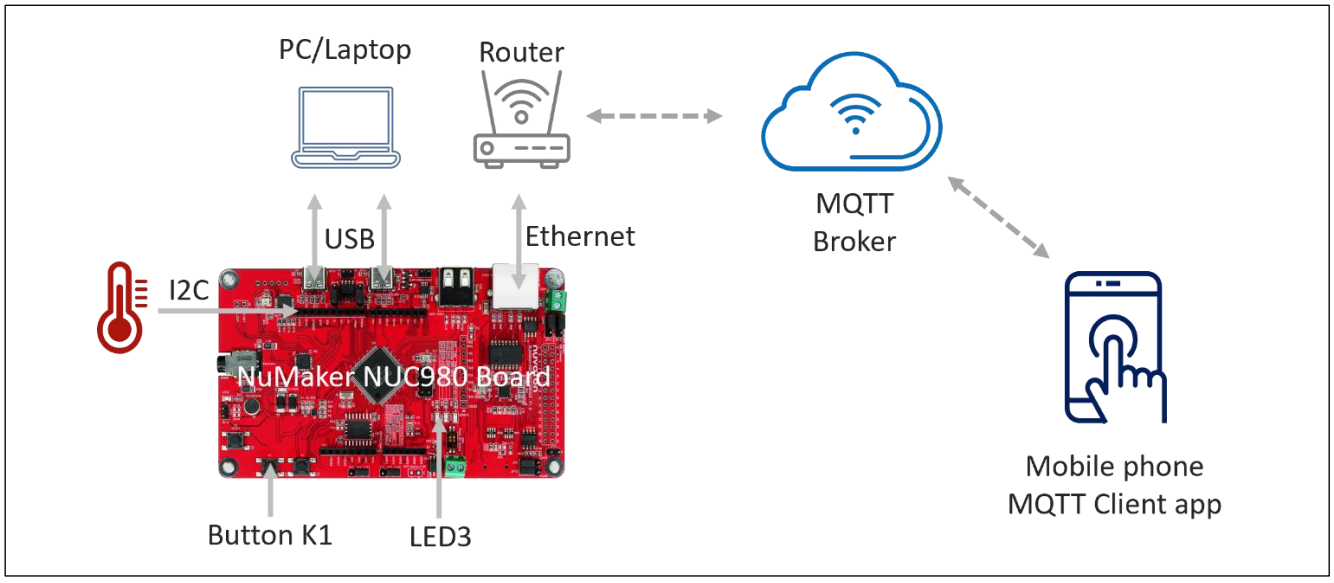


图 1-1 本范例的架构图

1.2 先决条件

请先详阅「软件与硬件需求」章节，准备所有所需组件。接着，依照「范例程序执行」章节的步骤，完成范例程序的建置、烧录至开发板，并设定 MQTT 应用程序。

范例 ZIP 文件中已包含预建工程，详细说明请参考「目录信息」章节。

1.3 执行结果

完成将固件烧录至 NuMaker NUC980 开发板并设定好 MQTT 应用程序后，即可依照下列步骤执行本范例。

- 重新启动 NuMaker NUC980 板，使用「root」账号登入。
- 启用以太网络并由 DHCP 服务器取得 IP 地址。

```
# udhcpc
```

- 执行 rcdemo。

```
# ./rcdemo
```

- NuMaker NUC980 板会连接至 MQTT broker，订阅主题并发布自身状态。

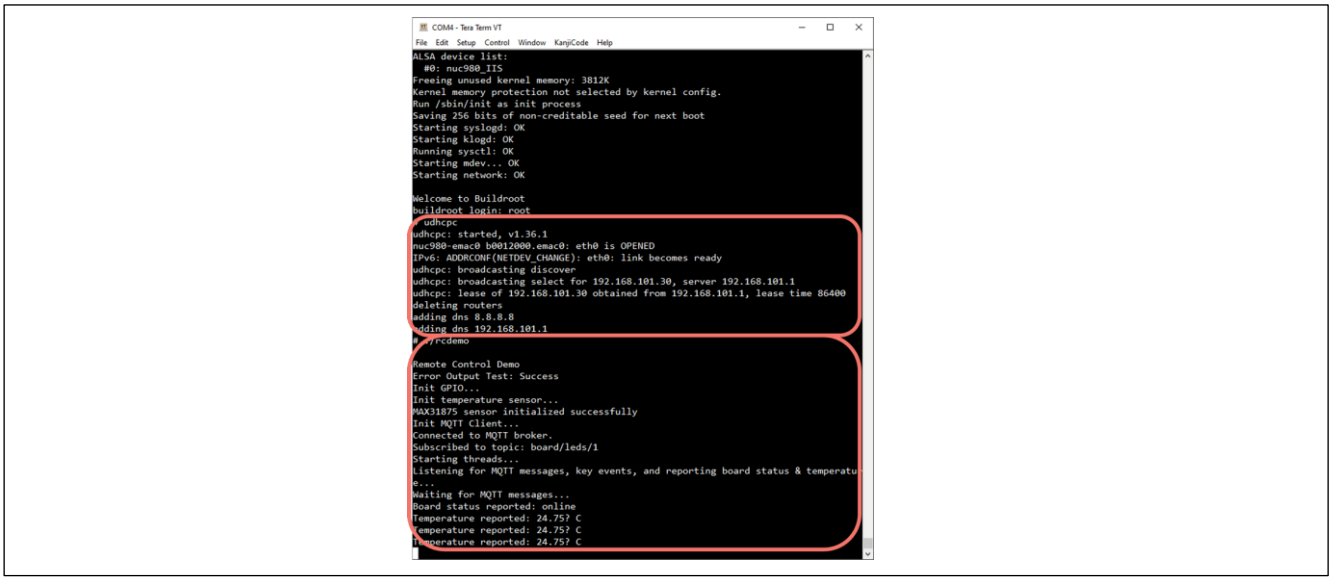


图 1-2 NUC980 终端机显示信息

- 在 Android 手机上开启 MQTT 应用程序。
- 确认应用程序已连接至 MQTT broker，云端图示应为橙色，若否请点击重新联机。
- 当 NuMaker 板成功联机并执行范例时，**Board Status** 显示为 **online**。
- 尝试下列步骤进行操作：
  - 按下并持续压住 NuMaker NUC980 开发板上的 **K1** 按键，应用程序中的 **Button status** 会显示为 **pressed**，放开后则显示为 **released**。
  - 在应用程序中切换 **LED 1** 开关，即可远程控制开发板上的 **LED**。
  - 应用程序会显示温度传感器的历史纪录。

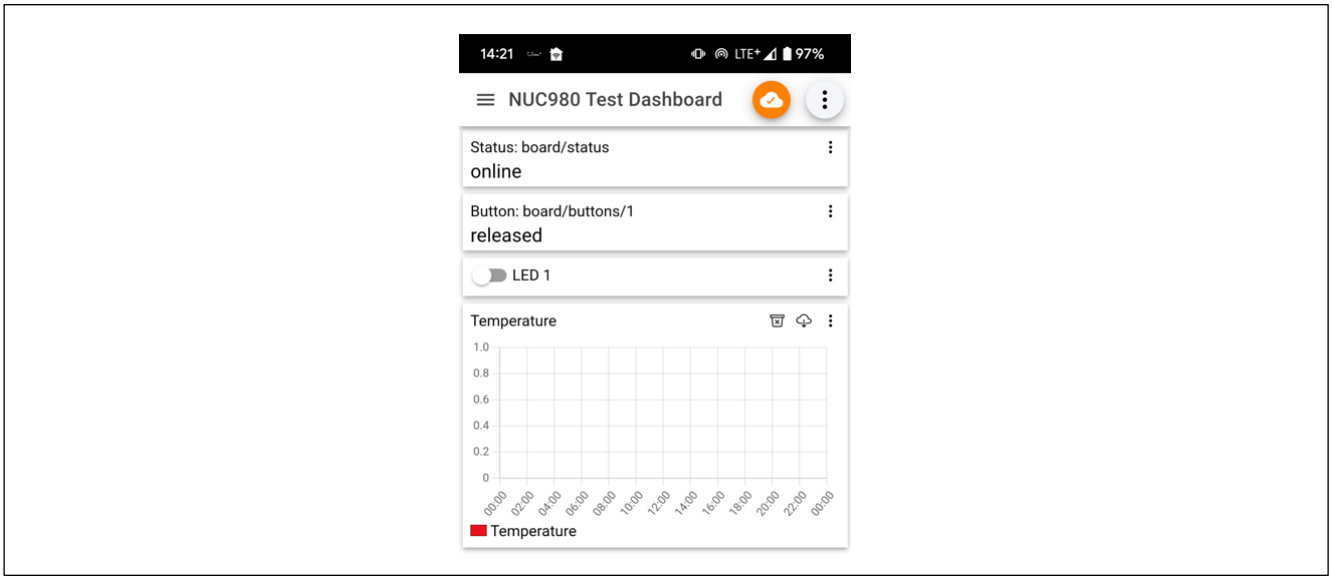


图 1-3 MQTT 应用程序画面

2. 程序代码介绍

本范例的所有程序代码皆集中于 rcdemo.c 文档。NUMAKER_BOARD 用于指定 LED、按键 I/O 及 I2C 口所对应的开发板型号。

```
#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 的网址与网口号设定如下：

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

范例程序会建立多个线程分别处理按键（key_monitor_thread）、板子状态（board_status_thread）、温度（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 函数库可协助开发者轻松建立 MQTT 客户端并订阅相关主题。

```
    // 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. 软件与硬件需求

3.1 软件需求

- 执行 Ubuntu 的 PC

  - 可为实体机或虚拟机（VM）。

- Buildroot

  - 下载网址：https://github.com/OpenNuvoton/buildroot_2024。

- PC 终端机软件

  - 如 Putty、Realterm、TeraTerm 等。

- MQTT 应用程序

  - IoT MQTT Panel (可于 Google Play 下载)。

- 或其他 MQTT 客户端应用程序。

3.2 硬件需求

- 以下任一片 NuMaker NUC980 开发板

  - NuMaker-IIoT-NUC980 系列，包含：
    - NuMaker-IIoT-NUC980
    - NuMaker-IIoT-NUC980G1
    - NuMaker-IIoT-NUC980G2
  - NuMaker-RTU-NUC980 (另稱 Chili)

- 一条 LAN 网络线

- 两条 USB 传输线

  - MicroUSB 适用于 NuMaker-IIoT-NUC980 或 Chili。

  - Type-C 适用于 NuMaker-IIoT-NUC980G1/G2。

- 一台具有 LAN 埠的 Wi-Fi 路由器

- Thermo 6 Click 扩充板 (内含 MAX31875 温度传感器)

请依连接方式完成范例环境的建置。

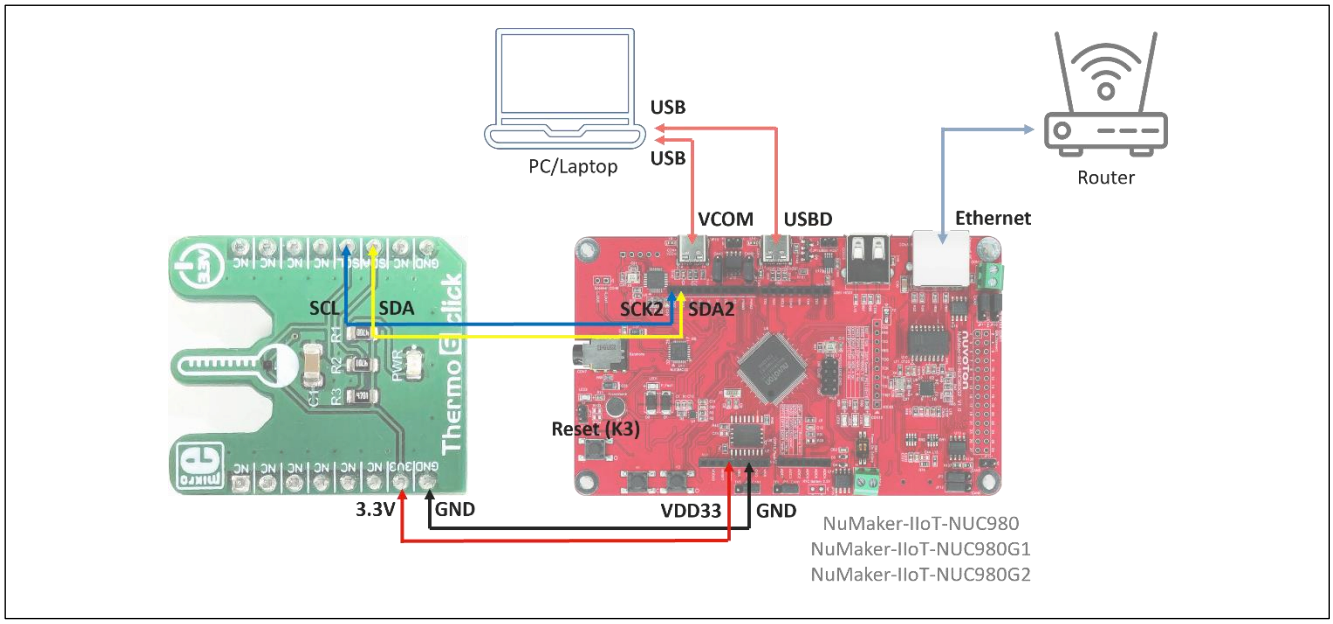


图 3-1 NuMaker-IIoT-NUC980 系列开发板连接图

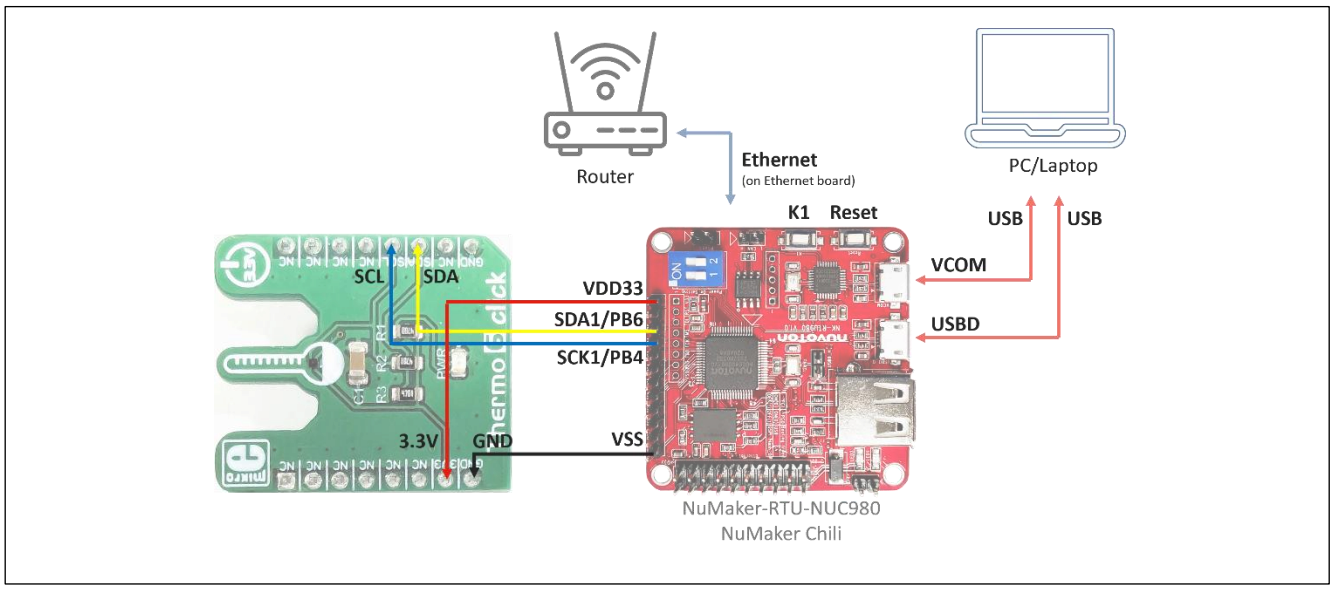


图 3-2 NuMaker-RTU-NUC980 (Chili) 系列开发板连接图

4. 目录信息

目录结构如下图所示。

EC_NUC980_Simple_OOB_Simulation_V1.00	
numaker-nuc980-iot	NuMaker-IIoT-NU980 系列开发板的 U-boot、system image 和 device tree
numaker-nuc980-rtu(chili)	NuMaker-RTU-NU980 (Chili) 开发板的 U-boot、system image 和 device tree
env.txt	环境配置文件
rcdemo.c	范例原始码

图 4-1 目录信息

5. 范例程序执行

请依照下列步骤，使用 NUC980 Buildroot 建立并设定开发环境，并完成范例程序的建置与部署。

5.1 固件建置

- 在 Ubuntu 里安装 Buildroot

  - 下载 NUC980 用的 buildroot_2024。

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

  - 为 NuMaker NUC980 开发板设定 Buildroot. 针对 NuMaker-IIoT-NUC980/G1。

```
$ make nuvoton_nuc980_iot_defconfig
```

    针对 NuMaker-IIoT-NUC980G2。

```
$ make nuvoton_nuc980_iot_g2_defconfig
```

    针对 NuMaker-RTU-NUC980 (Chili) 。

```
$ make nuvoton_nuc980_chili_defconfig
```

  - 设定 Buildroot 加入相关的套件。

```
$ make menuconfig
```

  - 勾选 Paho MQTT 函数库。

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

  - 然后按 <Save> 储存, 再按 <Exit> 离开设定画面。

  - 设定 Linux kernel 支持 NUC980 按键事件。

```
$ make linux-menuconfig
```

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

- 然后按 <Save> 储存, 再按 <Exit> 离开 Linux kernel 设定。

- 建构系统映像文件。

```
$ make
```

- 仅在 **NuMaker-RTU-NUC980 (Chili)** 需修改 device tree 以启用 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>;
    };
```

- 范例 ZIP 文件已包含 rcdemo.c 原始码，请直接复制至 Buildroot 目录。

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

- 设定 cross-compiler 环境。

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

- 编译范例程序 rcdemo.c（针对 **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
```

- 编译范例程序 rcdemo.c（针对 **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
```

- 将执行档复制到 root 的 home directory。

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

- 重新建置映像档以纳入 rcdemo 执行档。

```
$ cd buildroot_2024
$ make
```

- 建置完成后，请从 Buildroot 子目录中取得下列文档：

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**	●	●	●

- 接着用纯文本编辑器建立 env.txt 文档，内容如下。env.txt 亦包含于范例 ZIP 文件中。

```
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 烧录固件至开发板

- 下载 **NuWriter** 工具以烧录固件到开发板

  - 下载网址 https://github.com/OpenNuvoton/NUC980_NuWriter/。

  - NuWriter 说明文件于「.\」目录下。

  - NuWriter Windows 驱动程序于「.\NuWriter\Driver」目录，若未安装请先安装。

  - NuWriter 工具于「.\NuWriter\NuWriter\Release」目录，无需安装。

- 开启开发板终端机 (Console)

  - 在 PC 或笔电上使用任一终端机软件（如 TeraTerm、Putty）开启 NUC980 VCOM，装置名称为「USB Serial Device」

  - 设定 Baud rate 为 115200, non-parity, 8-bit, one stop bit.

- **NuMaker-IIoT-NUC980 series** 系列开发板烧录固件：

- 将 **SW1** 的 **No.1** 和 **No.2** 皆拨至 **ON** 以从 USB 开机。
- 按下 **Reset** (**K3**) 键由 USB 开机。
- 使用 **NuWriter** 依下表烧录下列文档。
- DDR 参数：依据开发板上 NUC980 型号选取正确的 **NUC980DK61Y.ini / NUC980DK63YC.ini / NUC980DK71YC.ini**。
- Type 选取 **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)** 开发板烧录固件：
  - 将 **SW1** 的 **No.1** 和 **No.2** 皆拨至 **ON** 以从 USB 开机。
  - 按下 **Reset** 键由 USB 开机。
  - 使用 **NuWriter** 依下表烧录下列文档。
  - DDR 参数：依据开发板上 NUC980 型号选取正确的 **NUC980DR61YC.ini / NUC980DR63YC.ini**。
  - Type 选取 **SPI NOR**。

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

- 将 NuMaker NUC980 开发板设为 SPI Flash 开机：
  - 将 **SW1** 的 **No.1** 和 **No.2** 皆拨至 **OFF**。
  - 按下 **Reset** 键由 SPI Flash 开机。
- 使用「root」账号登入。

5.3 设定 MQTT 应用程序

- 在 Android 手机上启动 IoT MQTT Panel 应用程序。
- 新增联机并输入 MQTT broker 信息。
  - **Connection name**：自定义名称，如 **NUC980-MQTT-Test**
  - **Client ID**：留空，会自动产生
  - Broker Web/IP address：填入 test.mosquitto.org
  - Port：1883
  - Network protocol：TCP
  - 点选「+」新增仪表板，命名如 **NUC980 Test Dashboard**
  - 然后点选 **CREATE**

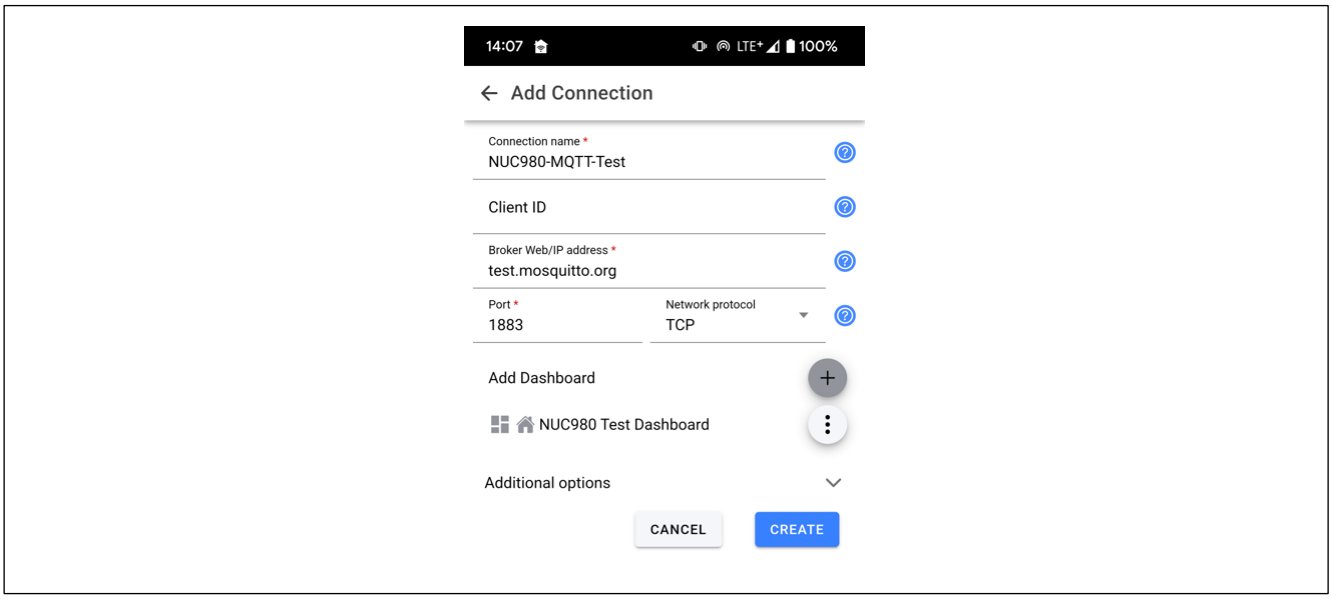


图 5-1 MQTT 应用程序建立联机

- 回到主画面，点选联机名称（如 NUC980-MQTT-Test）新增面板。
- 新增四个面板。

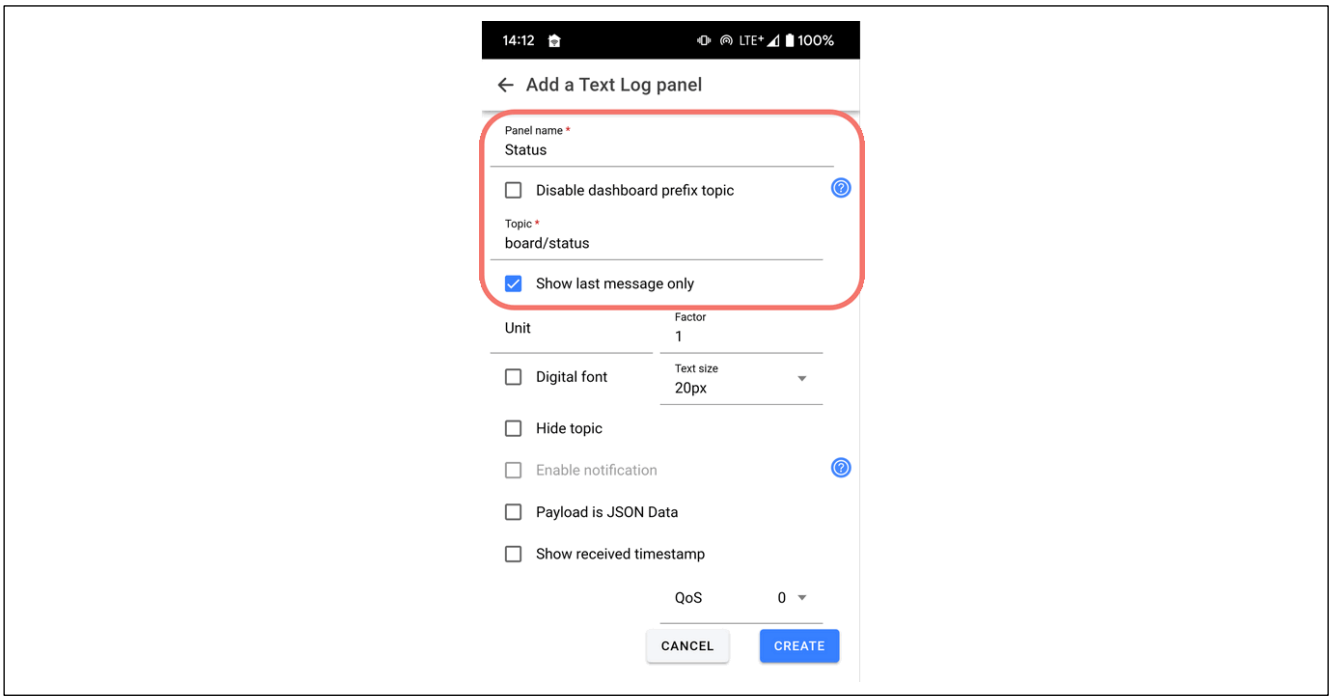


图 5-2 MQTT 应用程序建立第一个面板

- 第一个面板

  - ♦ 选择 Text Log
  - ♦ Panel name：Status
  - ♦ Topic：board/status
  - ♦ 勾选 Show last message only
  - ♦ 点选 CREATE

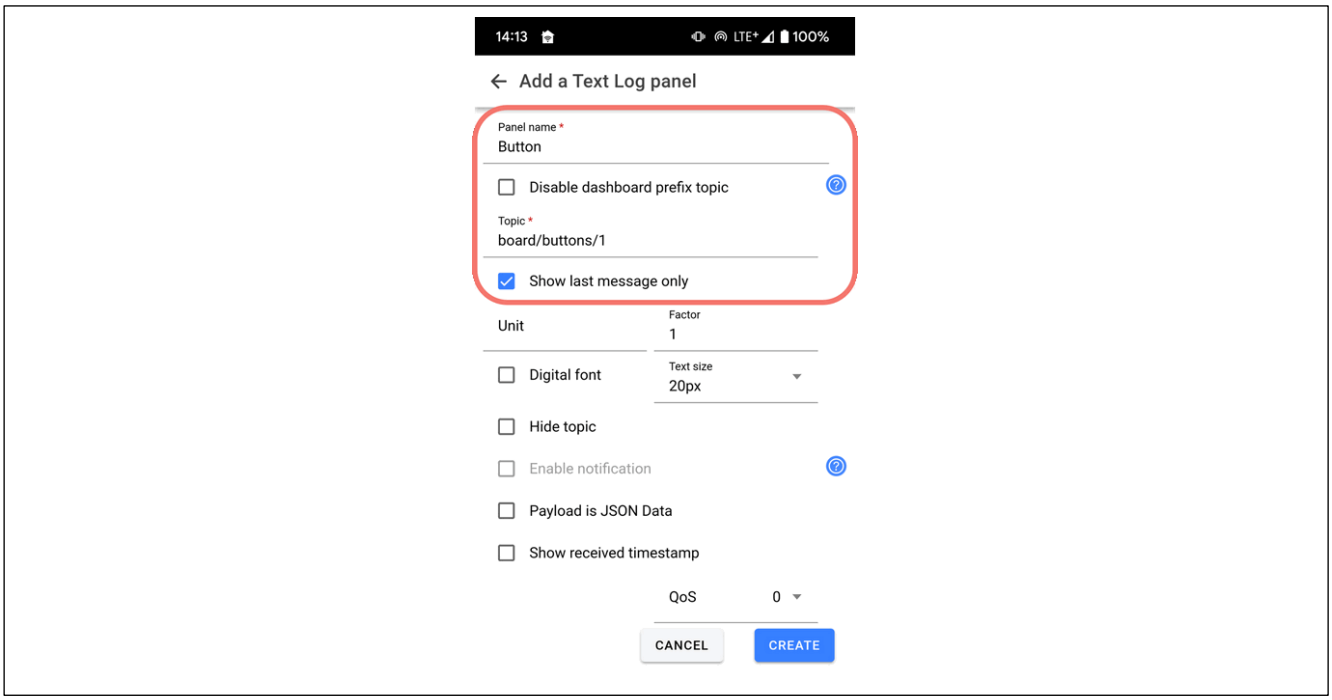


图 5-3 MQTT 应用程序建立第二个面板

- 第二个面板

  - ♦ 选择 Text Log
  - ♦ Panel name：Button
  - ♦ Topic：board/buttons/1
  - ♦ 勾选 Show last message only
  - ♦ 点选 CREATE

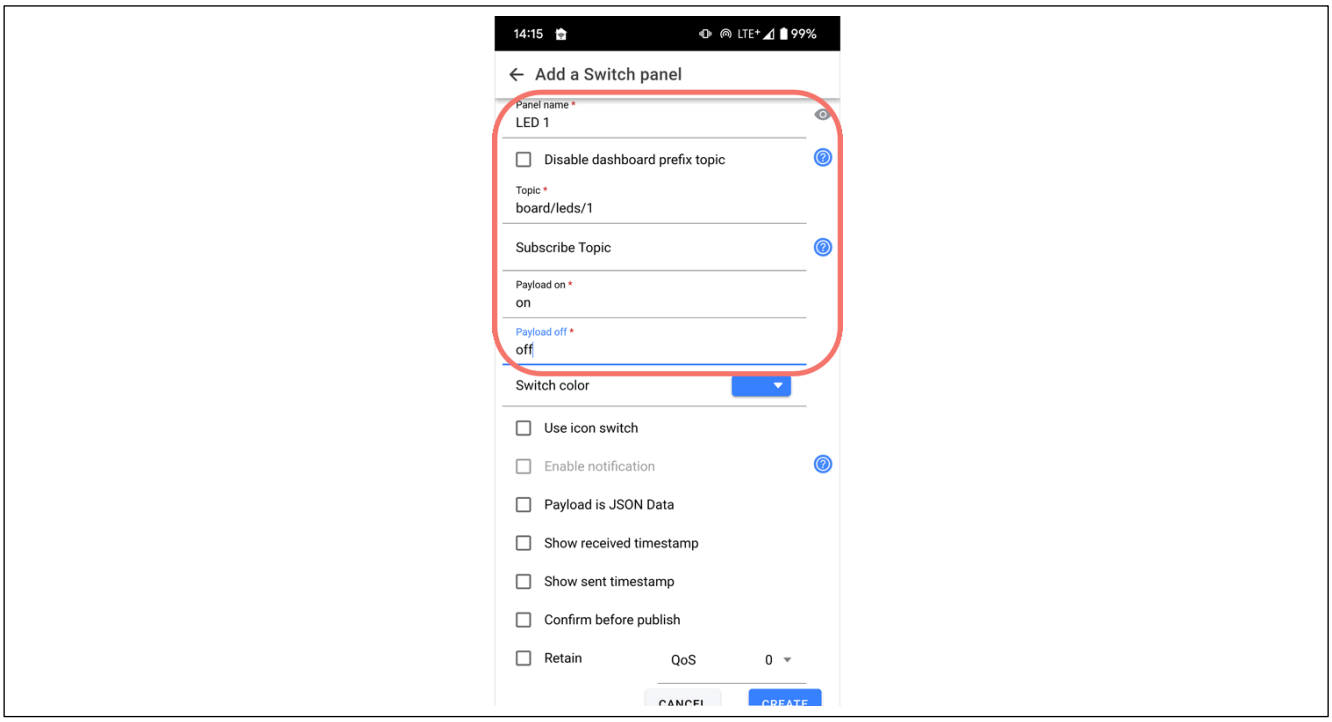


图 5-4 MQTT 应用程序建立第三个面板

- 第三个面板

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

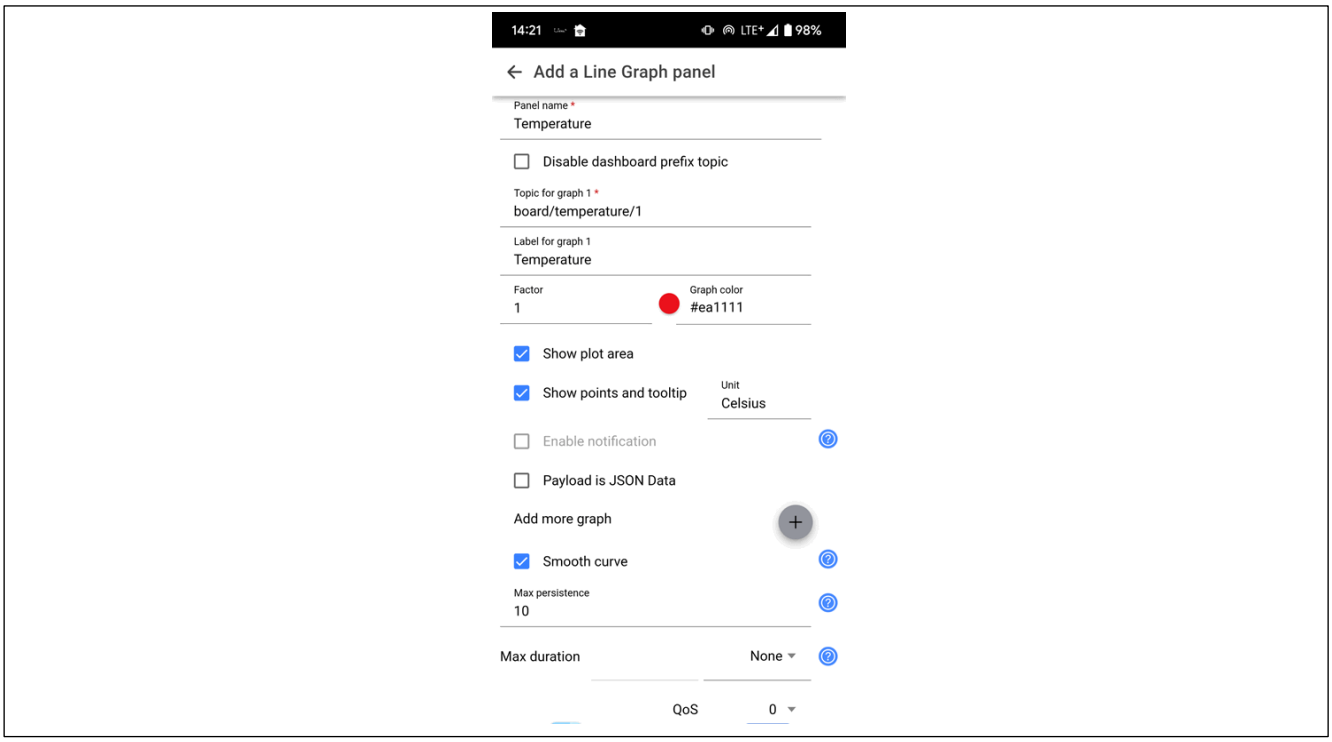


图 5-5 MQTT 应用程序建立第四个面板

- 第四个面板

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

6. 结尾

本范例说明如何利用 Nuvoton NuMaker NUC980 系列开发板于嵌入式 Linux 环境下，结合 MQTT 协议与行动 IoT 控制面板应用程序，实现远程监控与控制功能。

利用 C 语言及 Paho MQTT 函数库，完整示范如何远程控制开发板上的周边（如 LED 与按键），并实时监控环境数据（如温度）。

展现硬件、软件与云端服务的无缝整合，提供 IoT 远程控制应用的实用解决方案，适合作为教学与实务应用之参考范例。

7. 修订纪录

Date	Revision	Description
2025.06.21	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.