MA35D0 edge device WEB server example on Linux

NuMicro® 64/32位系列微控制器範例程式碼介紹

文件資訊

應用簡述	本範例程式說明如何於 MA35D0 系列開發平臺透過網頁介面進行遠端監控和控制，利用 Node.js 和 HTML5 實現對系統指標和跨多個裝置管理的即時更新。
BSP 版本	Linux-5.10.x
開發平台	NuMaker-IoT-MA35D0 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

此範例代碼的目的是提供在系統內使用 Node.js 和 HTML5 建構遠端控制介面的方式。此範例代碼能夠透過網頁瀏覽器連線，提供系統狀態的即時更新，如裝置 IP 位址、UID、CPU 負載、記憶體容量，以及 IO 狀態和控制。另外，它還具備監控重要事件和進行遠端 LED 控制的能力。此範例的特色在於支援多台 MA35D0 EVB 裝置之間的資訊共享，增強協同作業和管理。

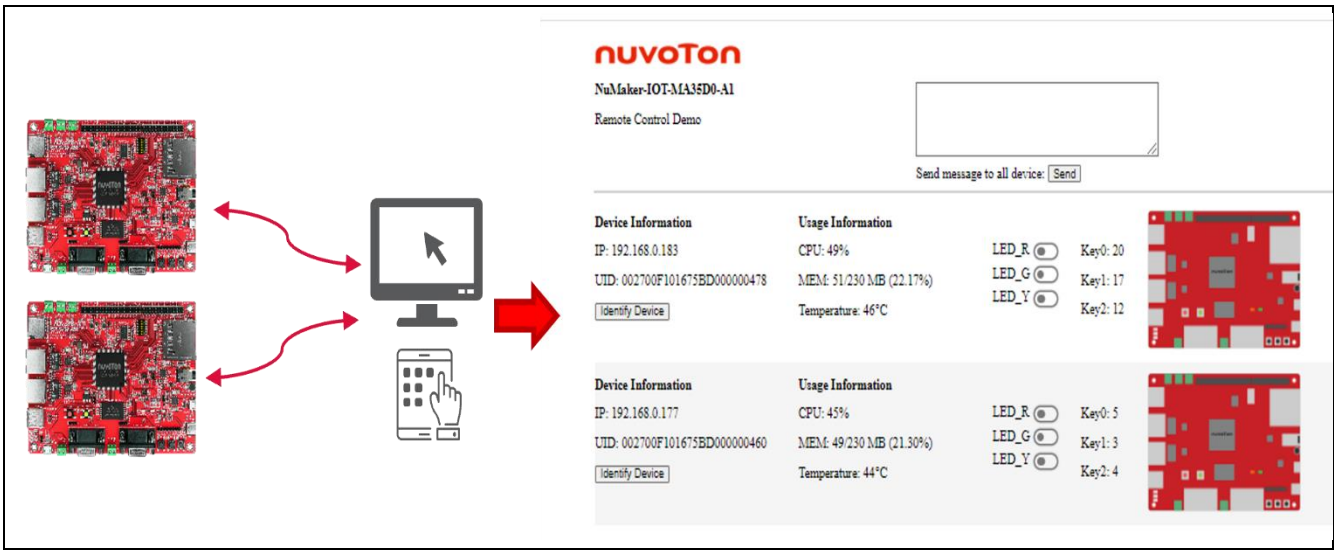

Figure 1-1 物聯網裝置遠程監控與控制概覽

1.1 軟體架構

- **init.sh** 初始化示範環境設定並啟動 Web 伺服器(server.js)，並呼叫 set_serverIP.sh 為客戶端 JavaScript 檔案分配 IP。
- **set_serverIP.sh** 設定 public 目錄中的 HTML layout 和客戶端 JavaScript 檔案。
- **server.js** 伺服器端代碼。
- **sys_monitor.sh** 負責獲取 CPU、記憶體和溫度等資訊。
- **LEDControl.sh** 控制 LED 的開啟/關閉狀態。
- **identify_device.sh** 裝置已收到對應的命令後閃爍 LED 6 次。
- **get_dev_info.sh** 取得裝置的 IP 和 UID 資訊。

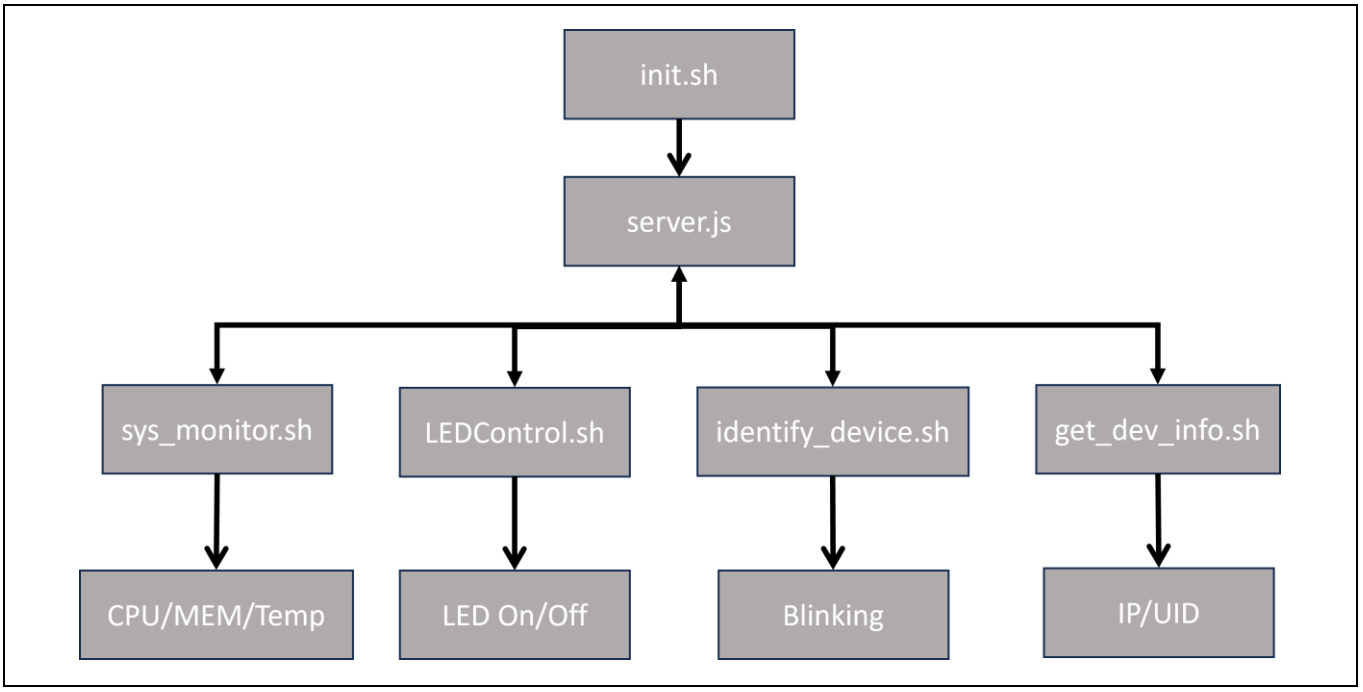

Figure 1-2 軟體框架

1.2 環境設定

MA35D0環境架設請參考 Nuvoton MA35D0 Evaluation Board Quick Start.

Buildroot

在 Buildroot menuconfig 啟用 MA35 支援所需的套件，請按照以下步驟操作：

1. 執行 make menuconfig 命令來開啟 Buildroot 設定選單。
2. 透過在套件選擇選單中選擇對應選項並按下空格鍵來啟用每個套件。可以使用'/'鍵按名稱搜尋套件。
    1) 將 BR2_ROOTFS_OVERLAY="…" 修改為 BR2_ROOTFS_OVERLAY="*board/nuvoton/ma35d0/overlay*"。
    2) 啟用 BR2_PACKAGE_NODEJS 以包含 Node.js 套件。
    3) 啟用 BR2_PACKAGE_ NODEJS_NPM 以包含 Node.js 套件。
    4) 啟用 BR2_PACKAGE_ NODEJS _MODULES_ADDITIONAL= “express ws” 以包含” express” 與 “ws” Nodes.js 套件。

3. 選定需要的套件後，退出 menuconfig 並儲存設定。

4. 在 Buildroot 執行 make 命令編譯韌體. 完成後使用 NuWriter 將韌體下載至 MA35D0 開發板。

Linux Device Tree

禁用 **gpio_keys_test** 驅動程式，直接讀取暫存器來獲取 IO 狀態。

ma35d0-iot-256m.dts

```
gpio_keys_test {
        compatible = "gpio-keys";
        status = "disabled"
   …
}
```

1.3 為範例代碼建立 Overlay 資料夾

在路徑*: **${Buildroot TOPDIR}/board/nuvoton/ma35d0/*** 建立一個新資料夾，命名為"overlay"，將範例代碼資料夾放置在此目錄下，在編譯之前，Buildroot 會將 overlay 資料夾的內容複製到 *output/target/*路徑下。

完成 1.2 和 1.3 步驟後，請進入 ***${Buildroot}/*** 目錄並運行 make 命令來編譯韌體。

編譯完成後，使用 **NuWriter** 將映射檔下載至 MA35D0 開發板。

2. 代碼介紹

2.1 系統初始化 “inittab”

inittab 文件有以下功能：

- **系統初始**：設定系統初始的過程，這包括掛載 proc 文件系統，重新掛載根文件系統為可讀寫模式，創建 */dev/pts* 和 */dev/shm* 目錄，掛載所有在 */etc/fstab* 文件中指定的文件系統，開啟交換分區（swap）等。
- **創建符號鏈接**：創建了指向當前處理進程文件描述符的 */dev/fd* 和標準輸入/輸出/錯誤的符號鏈接。
- **設定主機名**：使用 */etc/hostname* 文件中的內容來設定系統的主機名。
- **執行自定義初始化腳本**：執行 */opt/init.sh* 腳本，用於特定硬件或應用程序的自定義初始化指令。
- **執行 rc 腳本**：執行 */etc/init.d/rcS* 腳本，這是系統提供的標準運行級別初始化腳本。
- **配置串口控制台**：在 ttyS0 串口上啟動 getty 程序，以便可以登錄。

2.2 init.sh

開機過程中設定 GPIO 引腳、網路配置，並啟動 Node.js 應用程式：

- **設定 GPIO**：這個腳本定義了兩個函數 export_gpio 和 set_direction，分別用於啟用特定的 GPIO 並設定其方向。
- **連接網路**：腳本會執行 udhcpc 命令，透過 DHCP 協議，閒置網路介面 eth0 從 DHCP 伺服器取得一個 IP 地址。如果這個命令失敗，腳本會輸出一個錯誤訊息並終止執行。
- **導出 Node.js 路模組**：通過 export NODE_PATH=*/usr/lib/node_modules* 指令，這行命令設置了 Node.js 模組的路徑環境變數。
- **運行伺服器應用程式**：通過 node server.js & 命令在後臺啟動一個 Node.js 伺服器應用程式。

2.3 set_serverIP.sh

將目前 eth0 網路介面的 IP 位址更新到 JavaScript 檔案(/opt/public/index.js)。

- **取得 IP 地址**：利用 ifconfig 命令獲取 eth0 網路介面的 IP 位址，並儲存到 eth0_IP 變數中。
- **檢查 IP 地址是否取得成功**：如果沒有成功取得 IP 位址（即 eth0_IP 變數內容為空），則輸出錯誤訊息並結束程式執行。
- **檢查 /opt/public/index.js 是否存在**：如果檔不存在，則輸出錯誤訊息並結束程式執行。
- **創建暫存檔案**：使用 mktemp 命令創建一個暫存檔案，並把路徑儲存到 TMP_FILE 變數中。
- **輸入新的第一行**：將含有 eth0 IP 位址的新第一行寫入暫存檔案
- **釋出 index.js 的原始內容**：使用 tail 命令將 */opt/public/index.js* 檔的第二行開始的所有內容追加到暫存檔案。
- **用更改後的文件覆蓋原文件**：使用 mv 命令將暫存檔案重命名為 */opt/public/index.js*，達到替換原始 index.js 檔的目的。

2.4 server.js

設置伺服器應用，提供與客戶端建立 WebSocket 連線，使用 shell 腳本完成與客戶端之間的通訊，包含監控和控制系統的 IO、系統狀態、LED 以及裝置識別。

- **引入模組**：引入所需的 Express、WebSocket 以及 child process 模組，以便創建網頁伺服器並能夠執行 Shell 腳本。
- **構建 Express 應用**：設置一個基於 Express 的應用，用於處理 HTTP 請求和提供靜態檔案服務。
- **監聽端口**：在 3000 端口啟動 HTTP 伺服器，以便客戶端可以通過網頁訪問服務。
- **設置服務器 IP**：執行 Shell 腳本 set_serverIP.sh 來設定伺服器的 IP 地址，這樣客戶端就可以連接到正確的伺服器。
- **獲取裝置資訊**：執行 get_dev_info.sh 腳本來獲取裝置的 IP 地址和 UID，並儲存於 devInfo 物件中，這些資訊用於識別連接的裝置。

- **啟動 WebSocket 服務器：**在 121 端口建立一個 WebSocket 服務器，用於與客戶端建立實時通信。
- **處理客戶端連接和斷開：**當客戶端連接或斷開時，會有相應的處理函數，用於追蹤連接的客戶端並在有必要時更新客戶端的狀態。
- **定時發送 IO 和系統狀態：**使用定時器定期檢查和發送裝置的 IO 狀態與系統狀態給所有連接的客戶端，確保客戶端了解最新的裝置狀態。
- **處理 WebSocket 消息：**定義了處理客戶端發來的 WebSocket 消息的函數，根據收到的指令執行相應的裝置控制或反饋操作。

2.5 sys_monitor.sh

監控 GPIO 引腳、LED，以及 CPU、RAM 等各種系統資源的狀態：

- **read_gpio_value**: 讀取指定 GPIO 引腳的值。
- **io_status:** 輸出連接到 GPIO 的按鍵狀態，以及連接到三個 LED 的亮度值。
- **sys_status:** 輸出系統的 CPU 使用率、RAM 使用情況和溫度值。
- **cpu_usage:** 使用 top 命令取得 CPU 的使用情況。
- **ram_usage:** 使用 free 命令取得 RAM 的使用情況。

2.6 LEDControl.sh

控制名稱為 LED0、LED1 或 LED2 的三個 LED 燈。

- 檢查用戶是否輸入一個合法的 LED 名稱，也就是 "LED0"、"LED1" 或 "LED2" 中的一個。如果沒有，腳本就會輸出錯誤訊息 "Invalid LED"。
- 如果輸入的 LED 名稱是合法的，腳本就會驗證第二個輸入參數，也就是 LED 的目標狀態。如果目標狀態是 "On"，腳本就會在設定在對應的 */sys/class*。

2.7 identify_device.sh

讓三個 LED 燈（LED0、LED1 和 LED2）閃爍效果。

2.8 get_dev_info.sh

取得裝置的 eth0 網絡介面的 IP 地址和唯一識別碼（UID）：

- **獲取 eth0 的 IP 地址**：使用 ifconfig 命令取得 eth0 的 IP 地址，並將其儲存在 eth0_IP 變數。
- **讀取記憶體位址以形成 UID**：使用 devmem 命令讀取特定的三個記憶體位址，然後把這些位址串接起來形成 UID。

2.9 index.html

網頁版面設計，這個網頁主要包括以下部分：

- **設備信息**：這部分將顯示設備的 IP 地址和唯一識別碼（UID）。閱讀器可以將這些信息用於辨認連接的設備。
- **設備使用信息**：這部分將顯示設備的 CPU 使用率、記憶體使用狀況和溫度。用戶可以用這些信息來了解設備的運行狀況。
- **LED 燈和按鍵狀態**：這部分允許用戶遠程控制並查看設備上的 LED 燈狀態，包含紅色（LED0）、綠色（LED1）及黃色（LED2）的燈。「On」代表打開，「Off」代表關閉。網頁上也將顯示 Key0，Key1 和 Key2 的當前狀態。
- **發送訊息給所有設備**：使用者可以通過在文本區域中輸入消息，然後點擊'發送'按鈕來將訊息發送給所有設備。

2.10 index.js

在本地網絡上透過WebSocket連接發現和互動多個設備。它允許用戶控制LED燈，識別設備，並向設備的控制台發送消息。用戶界面會根據從設備收到的狀態資訊動態更新，主要功能如下：

- **建立 WebSocket 連線**：程式碼會掃描特定的 IP 範圍，並嘗試對每個 IP 地址建立 Web Socket 連線。若成功建立連線，程式碼便會建立對應的設備物件並存放於 deviceProfile 陣列中。
- **處理 WebSocket 訊息**：對每個建立的 WebSocket 連線都註冊了事件處理器來處理開啟、錯誤、關閉和接收訊息的情況。如果 WebSocket 從服務器收到訊息，程式碼會進行解析並更新對應設備的狀態和使用者介面。

- **動態生成設備狀態介面**：透過 duplicateElement 函數，為每個發現的設備複製並插入一個新的設備狀態區塊到網頁上，並且該設備狀態區塊會顯示該設備的 IP、UID、按鍵次數和溫度等信息。
- **設備識別和 LED 控制**：identifyDevice 與 controlLED 函數分別用於發送識別設備的請求和控制 LED 燈的開或關。
- **發送信息至所有設備**：sendMessageToAllDevices 函數會向所有已連接的設備發送特定的消息。

3. 軟體與硬體需求

3.1 軟體需求

- BSP 版本
  - Linux-5.10.x

3.2 硬體需求

- 電路元件
  - NuMaker-IoT-MA35D0 V1.0

4. 目錄資訊

EC_MA35D0_Edge_Device_WEB_Server_Example_On_Linux_V1.00	
SampleCode	
etc	
inittab	Script to handle boot-up process
opt	
get_dev_info.sh	Procedure to obtain IP and UID
identify_device.sh	Logic to handle LED blinking
init.sh	Custom script for handling boot procedures
LEDControl.sh	Script for managing LEDs
server.js	Server-side application
set_serverIP.sh	Update IP in the index.js file
sys_monitor.sh	Procedure to obtain CPU/MEM usage and I/O status
Images	Images for the web interface
public	Design of the web page layout
index.html	The client-side application
index.js	

Figure 4-1 目錄資訊

5. 範例程式執行

初始設置和 LED 閃燈行為:

將 MA35D0 的乙太網路端口(RMII0)連接到路由器，並將電腦端連接到控制台，並使用 TYPE C 供電。開機後，預設系統將成功運行該示例，且 LED 將同步閃爍 6 次。

通過控制台和瀏覽器訪問 MA35D0 裝置資訊:

如果演示運行成功，則會顯示 IP 地址和相關資訊:
使用相同網域的 PC 或手機，輸入${MA35D0 IP}:3000 進入瀏覽器。

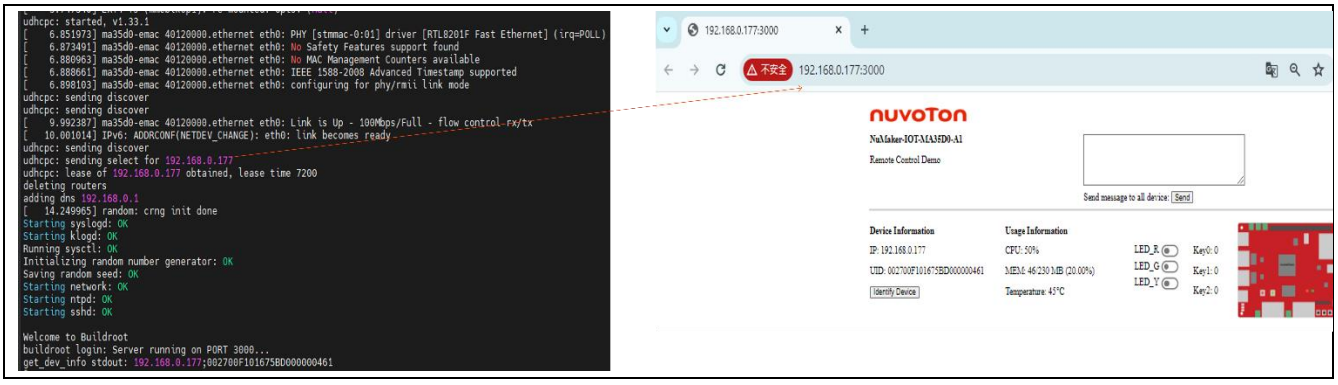

Figure 5-1 訪問 MA35D0 裝備資訊

廣播:

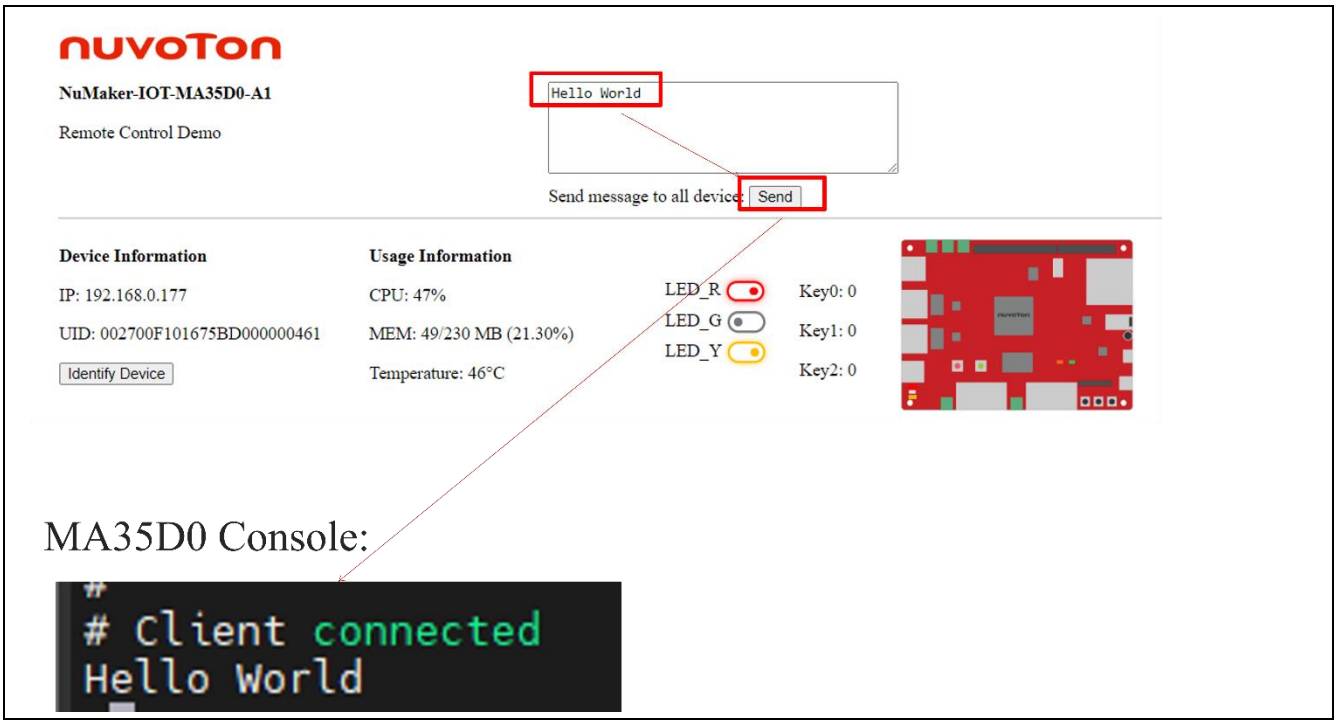

Figure 5-2 廣播

控制與按鍵狀態:

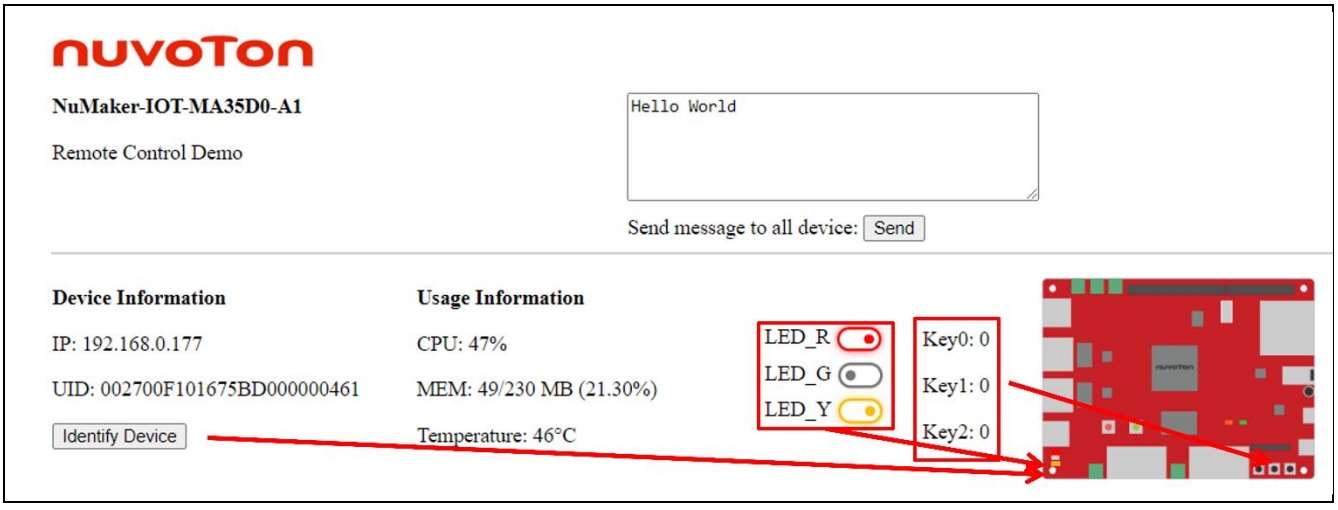

Figure 5-3 控制與按鍵狀態

多個 MA35D0 EVB 共享信息:

開啟另一個 EVB 並連接到控制台，在*/opt/init.sh* 修改初始化腳本以更改 MAC 地址。

init.sh 代碼片段

```
#Set new MAC address
ifconfig eth0 down
ifconfig eth0 hw ether 22:33:44:55:66:77
```

重新啟動 EVB 並檢查新的 IP 地址。

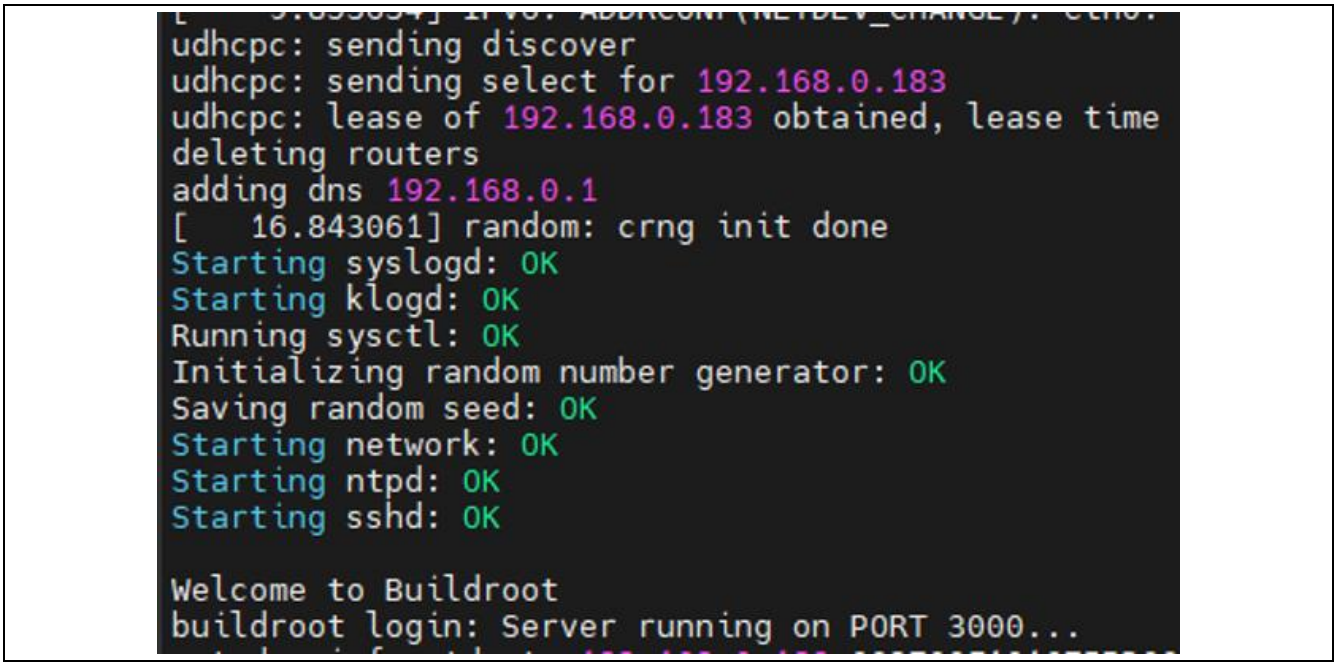

Figure 5-4 檢查新 IP 地址

輸入 ${MA35D0 IP}:3000 到瀏覽器，以進入每個 EVB 的網頁伺服器。

通過瀏覽器，可以在兩個 EVB 的伺服器上觀察每個 MA35D0 的狀態。

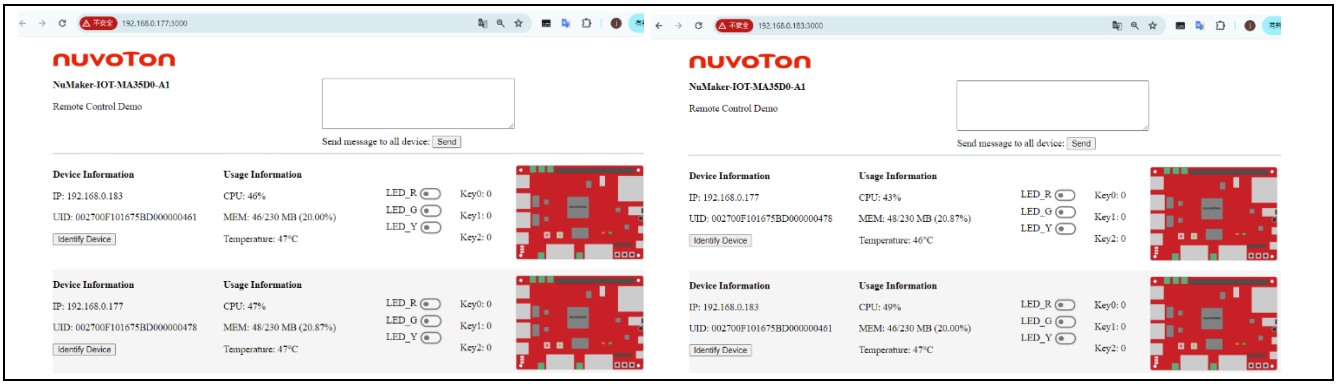

Figure 5-5 多個 EVB 狀態資訊

6. 修訂紀錄

Date	Revision	Description
2024.05.13	1.00	初始版本。

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.