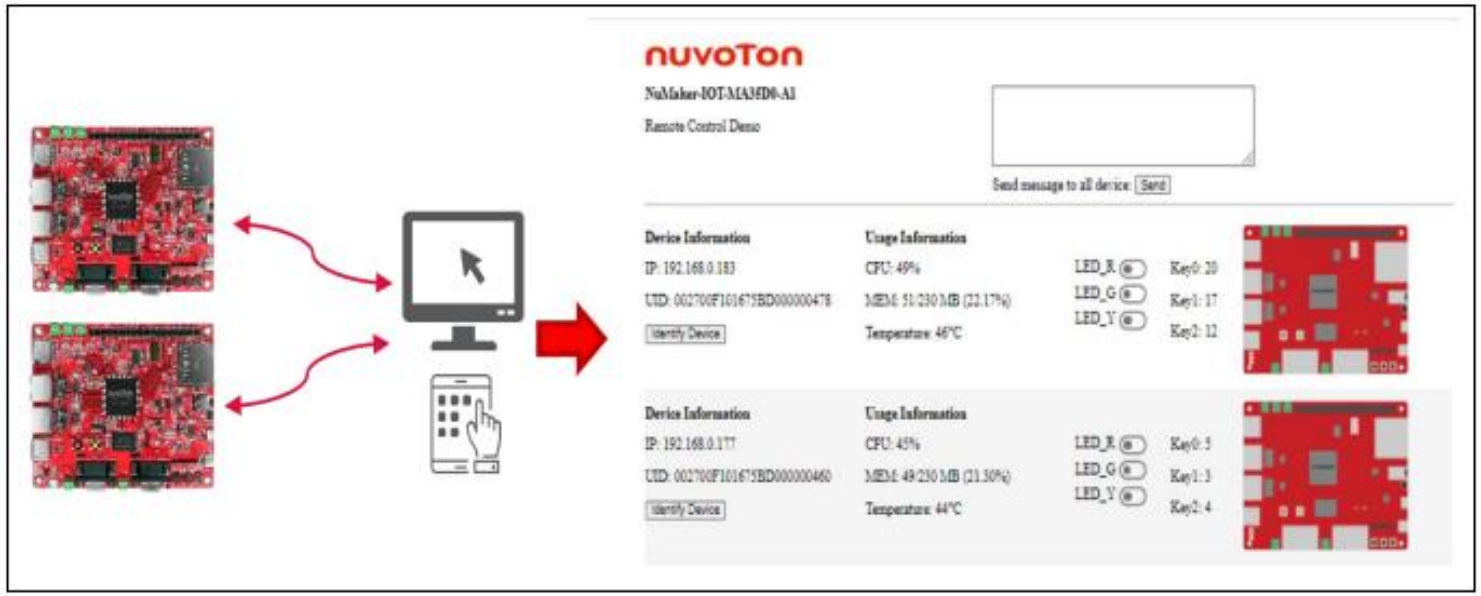
IoT Application Practice

2023.03.15

IoT Application Scenario Overview

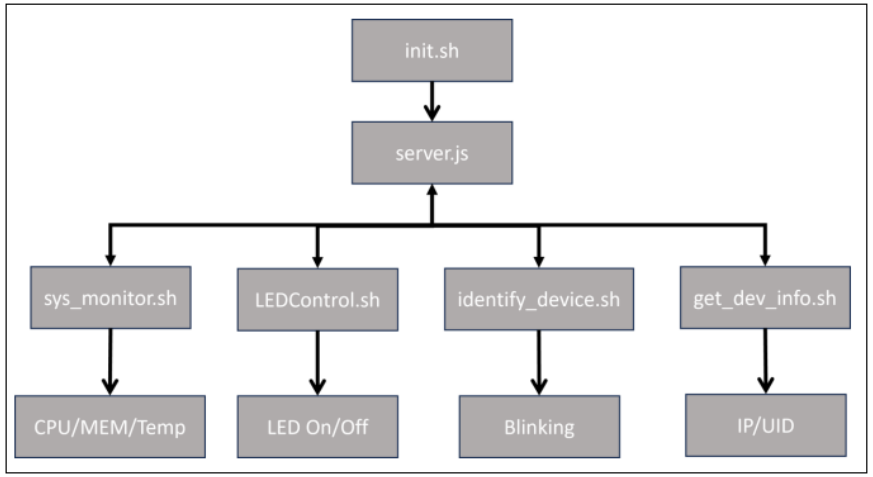
Example Code Features

- **System Status Updates**:
  - Device's IP address
  - UID
  - CPU load
  - Memory capacity
  - IO status and controls
- **Remote Monitoring & Control**:
  - Monitor key events
  - Remote LED controls
- **Inter-Device Information Sharing**:
  - Supports collaborative operations
  - Multiple MA35D0 EVB units



Software Architecture

- **init.sh** initializes the demo environment settings and starts the web server (server.js), and calls set_serverIP.sh to allocate IP to the client JavaScript file.
- **set_serverIP.sh** affects the HTML layout and client JavaScript files in the public directory.
- **server.js** server JavaScript file.
- **sys_monitor.sh** script is responsible for obtaining information such as CPU, memory, and temperature.
- **LEDControl.sh** controls the LED's on/off state.
- **dentify_device.sh** flashes the LED 6 times to indicate the device has received the corresponding command.
- **get_dev_info.sh** retrieves the device's IP and UID information.



Hands-On Time

- Original Buildroot
- Enable packages
- Modify device tree
- Add example code
- Compile firmware
- Download and Run

Environment Settings - Buildroot

Open the Buildroot configuration menu by running the command

“**make menuconfig**”

Search for each package by selecting the corresponding option in the package selection menu. Use the **‘/’** key to search for packages by name.

- **Modify BR2_ROOTFS_OVERLAY=””** to **BR2_ROOTFS_OVERLAY=” board/nuvoton/ma35d0/overlay”**
- Enable **BR2_PACKAGE_NODEJS** to include the Node.js package.
- Enable **BR2_PACKAGE_NODEJS_NPM** to include the Node.js package manager.
- Enable **BR2_PACKAGE_MODULES_NODEJS_ADDITIONAL= “express ws”** to include “express” and “ws” Node.js packages
- Enable **BR_PACKAGE_MODULES_RTL8188EU** to install Wi-Fi dongle firmware into rootfs**.**

Environment Settings - Buildroot

BR2_ROOTFS_OVERLAY

```
Symbol: BR2_ROOTFS_OVERLAY [=board/nuvoton/ma35d0/overlay]
Type  : string
Prompt: Root filesystem overlay directories
  Location:
(1) -> System configuration
  Defined at system/Config.in:525
```

Environment Settings - Buildroot

BR2_PACKAGE_NODEJS

```
Symbol: BR2_PACKAGE_NODEJS [=y]
Type  : bool
Prompt: nodejs
  Location:
    -> Target packages
(1)   -> Interpreter languages and scripting
  Defined at package/nodejs/Config.in:16
  Depends on: BR2_TOOLCHAIN_HAS_THREADS_NPTL [=y] && BR2_INSTALL_LIBSTDCPP [=y] && BR2_PACKAGE_NODEJS_ARCH_SUPPORTS [=y] &&
  Selects: BR2_PACKAGE_C_ARES [=y] && BR2_PACKAGE_LIBUV [=y] && BR2_PACKAGE_ZLIB [=y] && BR2_PACKAGE_NGHTTP2 [=y]
  Selected by [n]:
  - BR2_BR2_PACKAGE_NODEJS_0_10_X [=n] && !BR2_SKIP_LEGACY [=n]
  - BR2_BR2_PACKAGE_NODEJS_0_12_X [=n] && !BR2_SKIP_LEGACY [=n]
  - BR2_BR2_PACKAGE_NODEJS_4_X [=n] && !BR2_SKIP_LEGACY [=n]
```

Environment Settings - Buildroot

BR2_PACKAGE_NODEJS_NPM

```
Symbol: BR2_PACKAGE_NODEJS_NPM [=y]
Type  : bool
Prompt: NPM for the target
  Location:
    -> Target packages
      -> Interpreter languages and scripting
(1)     -> nodejs (BR2_PACKAGE_NODEJS [=y])
  Defined at package/nodejs/Config.in:41
  Depends on: BR2_PACKAGE_NODEJS [=y]
  Selects: BR2_PACKAGE_OPENSSL [=y]
```

Environment Settings - Buildroot

BR2_PACKAGE_NODEJS_MODULES_ADDITIONAL

```
Symbol: BR2_PACKAGE_NODEJS_MODULES_ADDITIONAL [=express ws]
Type  : string
Prompt: Additional modules
  Location:
    -> Target packages
      -> Interpreter languages and scripting
(4)     -> nodejs (BR2_PACKAGE_NODEJS [=y])
  Defined at package/nodejs/Config.in:53
  Depends on: BR2_PACKAGE_NODEJS [=y]
```

Environment Settings - Buildroot

BR_PACKAGE_MODULES_RTL8188EU

```
Symbol: BR2_PACKAGE_RTL8188EU [=y]
Type  : bool
Prompt: rtl8188eu
  Location:
    -> Target packages
(1)   -> Hardware handling
  Defined at package/rtl8188eu/Config.in:1
  Depends on: BR2_LINUX_KERNEL [=y]
```

Environment Settings – Linux Device Tree

- Disable the **gpio_keys_test** driver, read the register directly to get the IO status

ma35d0-iot-256m.dts

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

Copy application to File System

- Create a new folder named "**overlay**" in the path: **${Buildroot TOPDIR}/board/nuvoton/ma35d0**
- Place the example code sub-folder(“**opt**” and “**etc**”) in this directory, Buildroot will copy the contents of the overlay folder to **${Buildroot}/output/target/** before compiling.

Directory Information

EC_MA35D0_Edge_Device_WEB_Server_Example_On_Linux_V1.00	
SampleCode	
etc	
inittab	Script to handle boot-up process
opt	
Images	Images for the web interface
public	
index.html	Design of the web page layout
index.js	The client-side application
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 IO status

Code Details

nuvoTon MA35D0 Series

MA35D0 edge device WEB server example on Linux

Example Code Introduction for 64/32-bit NuMicro® Family

Document Information

Application	The MA35D0 series sample code shows remote monitoring and control via a web interface, using Node.js and HTML5 for real-time updates on system metrics and device management across multiple units.
BSP Version	Linux-5.10.x
Hardware	NuMaker-IoT-MA35D0

Joy of innovation

NUVOTON

谢谢

謝謝

Děkuji

Bedankt

Thank you

Kiitos

Merci

Danke

Grazie

ありがとう

감사합니다

Dziękujemy

Obrigado

Спасибо

Gracias

Teşekkür ederim

Cảm ơn