

DUSUN DSGW-021 L-Serial Smart Gateway User Guide

Hangzhou Roombanker Technology Co.,Ltd.

A DUSUN company

SDK Quick Start Guide

Product Name: L-Serial Smart Gateway
Model Name: DSGW-021

Revision History

Specification		Sect.	Update Description	By
Rev	Date			
1.0	2021-08-06		New version release	

Approvals

Organization	Name	Title	Date

Contents

- 1 1. Introduction
- 2 2. Gateway Information
 - 2.1 2.1 Basic information
 - 2.2 2.2 Gateway appearance
- 3 3. Target Setup
 - 3.1 3.1 Connecting a gateway – Power
 - 3.2 3.2 Connecting a gateway - LAN port
- 4 4. Compile the Environment to Build
 - 4.1 4.1 OpenWrt Version
 - 4.2 4.2 PC Compile the Environment to Build
 - 4.3 4.3 OpenWrt configuration and compilation
- 5 5. Network interfaces
- 6 6. Openwrt restore to factory setting
- 7 7. Add APP component in Openwrt
- 8 8. uboot build and upgrade
- 9 9. System firmware upgrade
- 10 10. Communication between zigbee module
- 11 11. Communication between Z-Wave module
- 12 12. Communication between BLE module
- 13 13. System firmware upgrade
- 14 14. Others
- 15 Documents / Resources
 - 15.1 References
- 16 Related Posts

1. Introduction

This Quick Start Guide explains the basics: how to connect and set up your target on the network; how to install the SDK; and how to build the firmware images.

The Linux Software Developer's Kit (SDK) is an embedded hardware and software suite that enables Linux developers to create applications on Dusun's DSGW-021 gateway.

Base on the MTK OpenWrt V19.07, and leveraging existing open source software, the SDK simplifies the process of adding custom applications. Device drivers, toolchain, Pre-defined configuration profiles, and sample applications are all included.

2. Gateway Information

2.1 Basic information

Processor: MTK7628A

Supply: DC-5V

ZigBee: EFR32MG1B232F256G

Z-Wave: ZGM130S037HGN

Bluetooth: EFR32BG21 or NRF52840

RAM: 64MB

Flash: 16MB

2.2 Gateway appearance



3. Target Setup

This section describes how to connect the gateway into your host computer and network.

3.1 Connecting a gateway – Power

3.1.1 Make sure that the power adapter is 5V/2A.

3.1.2 Select the appropriate power plug adaptor for your geographical location. Insert it into the slot on the Universal Power Supply; then plug the power supply into an outlet.

3.1.3 Connect the output plug of the power supply to the gateway

3.2 Connecting a gateway - LAN port

3.2.1 Connect one end of the network cable to the Ethernet port on the laptop or desktop

3.2.2 Connect the other end of network cable to the LAN port on the gateway.

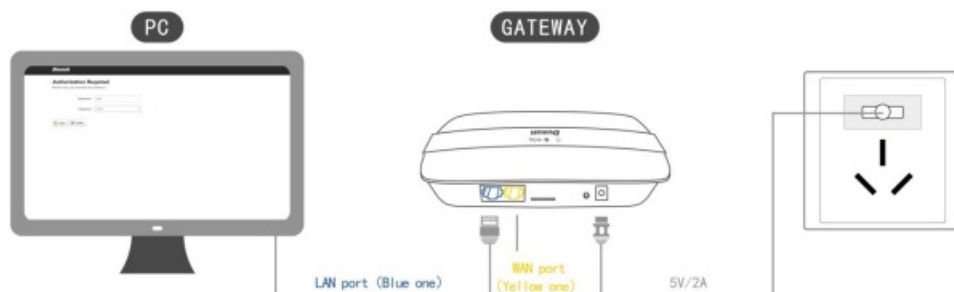


Figure3-1. Connecting a gateway via Ethernet port

Before you set up your development test bed, please connect the PCB serial port to your develop PC via USB-to-serial bridge.

Serial port setting:

Baud rate: **57600**

Bits: **8**

Stop Bits: **1**

Hardware flow control: **None**

4. Compile the Environment to Build

4.1 OpenWrt Version

Take OpenwWRT19.07 as an example

4.2 PC Compile the Environment to Build

Compilation environment: Linux

Openwrt's compilation tool is automatically generated by SDK built, no additional installation is required

4.3 OpenWrt configuration and compilation

Get the source code from Dusun FTP server or github and uncompress it under your work directory.

Github link: <https://github.com/openwrt/openwrt.git>

Checkout to correct branch after clone, take OpenWRT19.07 as an example,

git checkout openwrt-19.07

git branch -a

```
[msh@git openwrt]$ git branch -a
master
* openwrt-19.07
remotes/origin/HEAD -> origin/master
remotes/origin/lede-17.01
remotes/origin/master
remotes/origin/openwrt-18.06
remotes/origin/openwrt-19.07
remotes/origin/openwrt-21.02
```

4.3.1 DTS file

a) About DTS file for connecting to Dusun gateway's hardware, please download in Dusun FTP server.

b) Then move it to the right directory **mv DSGW021.dts target/linux/ramips/dts/**

c) Add the component in file **vi target/linux/ramips/image/mt76x8.mk**

```
define Device/DSGW021
    DTS := DSGW021
    IMAGE_SIZE := 16064k
    DEVICE_TITLE := DSGW021
    DEVICE_PACKAGES := kmod-usb2 kmod-usb-ohci
endef
TARGET_DEVICES += DSGW021
```

4.3.2 Configure compilation options

a) Configure compilation options

There is already a default configuration in the SDK, which satisfies the basic functions of routing. Customers could also customize the configuration according to their own needs.

Command:

cd <work_dir>/openwrt/

make menuconfig

Choose system (MediaTek Ralink MIPS)

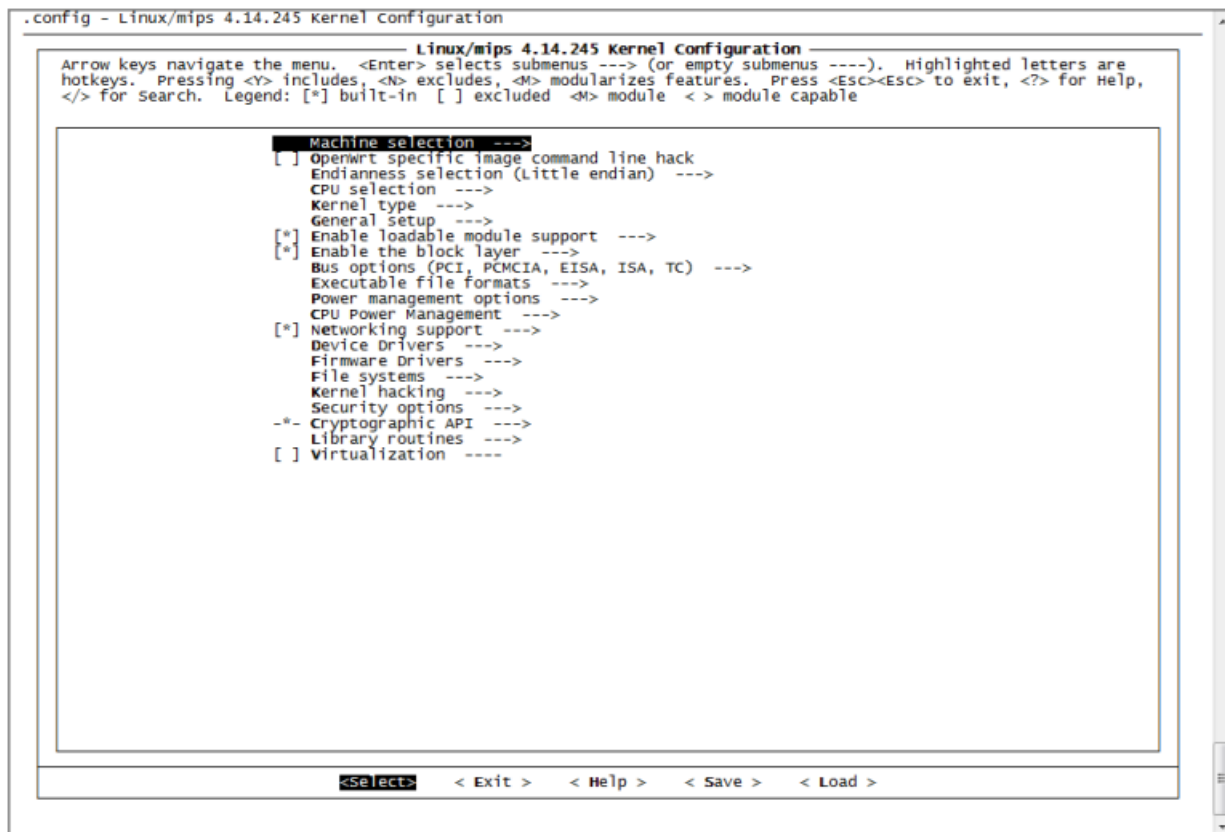
Select Subtarget MT76x8 based boards



Choose target profile DSGW-021



If user want to change kernel configuration make kernel_menuconfig



Exit and save the configuration.

b) Build the image

Just use command 'make V=99', the image built will take a long time.

The target image will be under ./bin/targets/ramips/mt7620

Image used for update is *openwrt-ramips-mt7620-dsi0134-squashfs-sysupgrade.bin*

5. Network interfaces

Login into the gateway device, using command 'ifconfig', you can get all network interfaces.

Interface eth1 connects to WAN port (Blue RJ45 port), eth0 is LAN port, it's connected to an internal switch on SoC, all LAN ports are connected to eth0.

Interface ra0 is 2.4G radio, rai0 is 5G radio.

6. Openwrt restore to factory setting

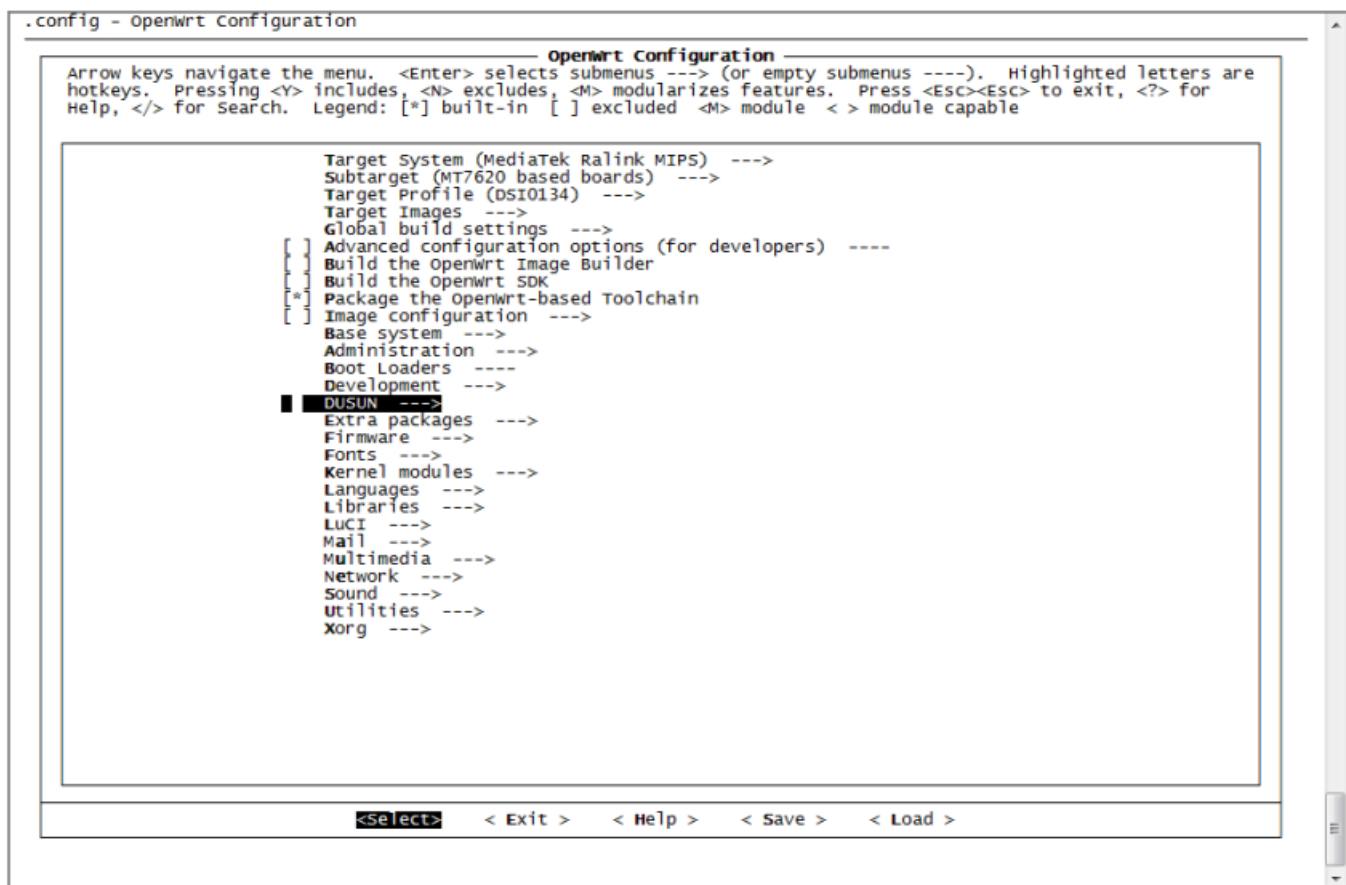
Command:

firstboot

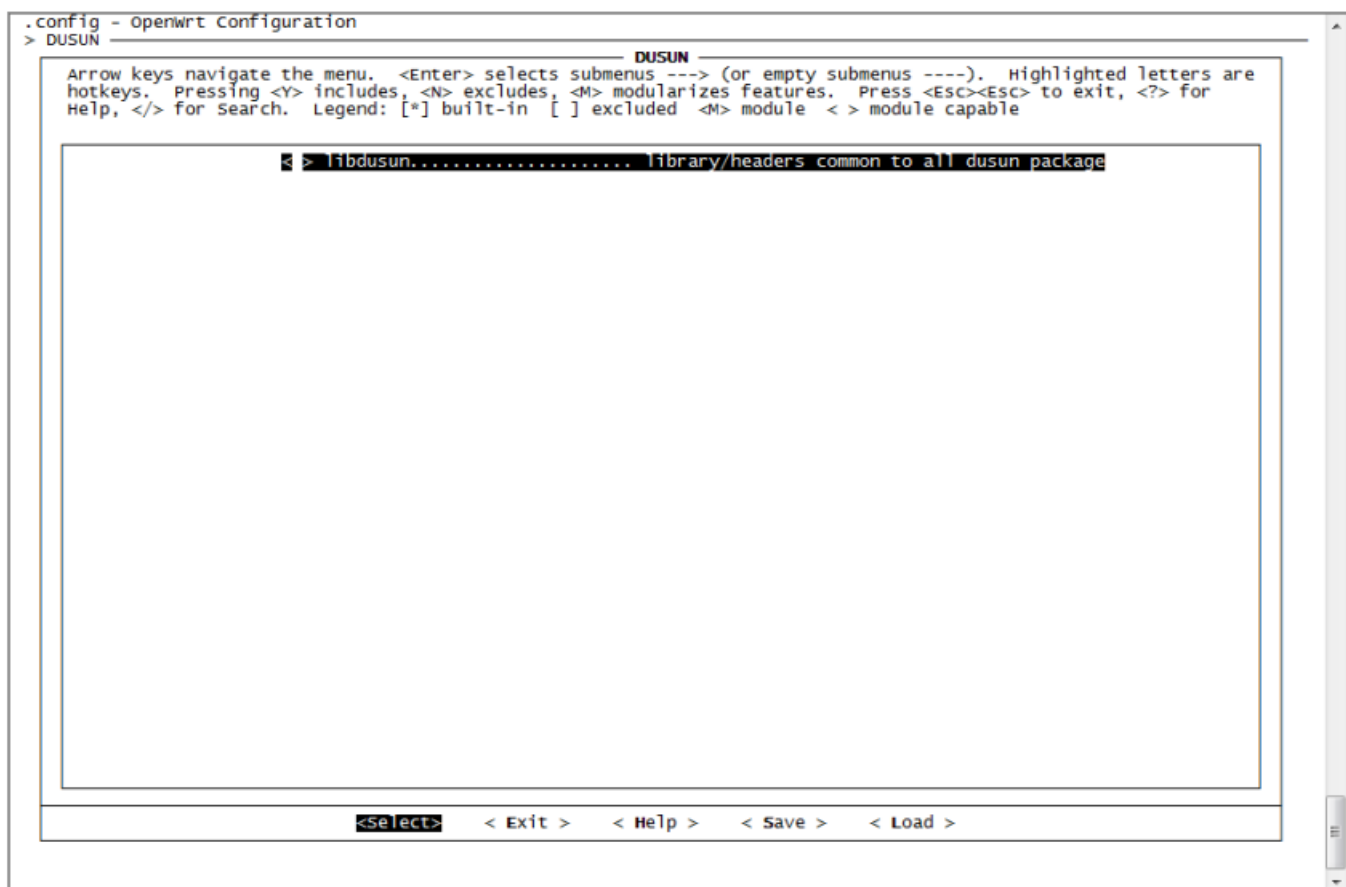
Press 'y' to confirm and then reboot system.

7. Add APP component in Openwrt

Take openwrt-sdk/package/dusun/lib_dusun as an example, If user want to add a new user lever application into openwrt SDK, just follow the example. After code is ready, you still need to add it to config file by "make menuconfig", and select Dusun



Select libdusun package,



Exit and save config, 'make V=99' again, you can get the image with libdusun

1. uboot configuration and build

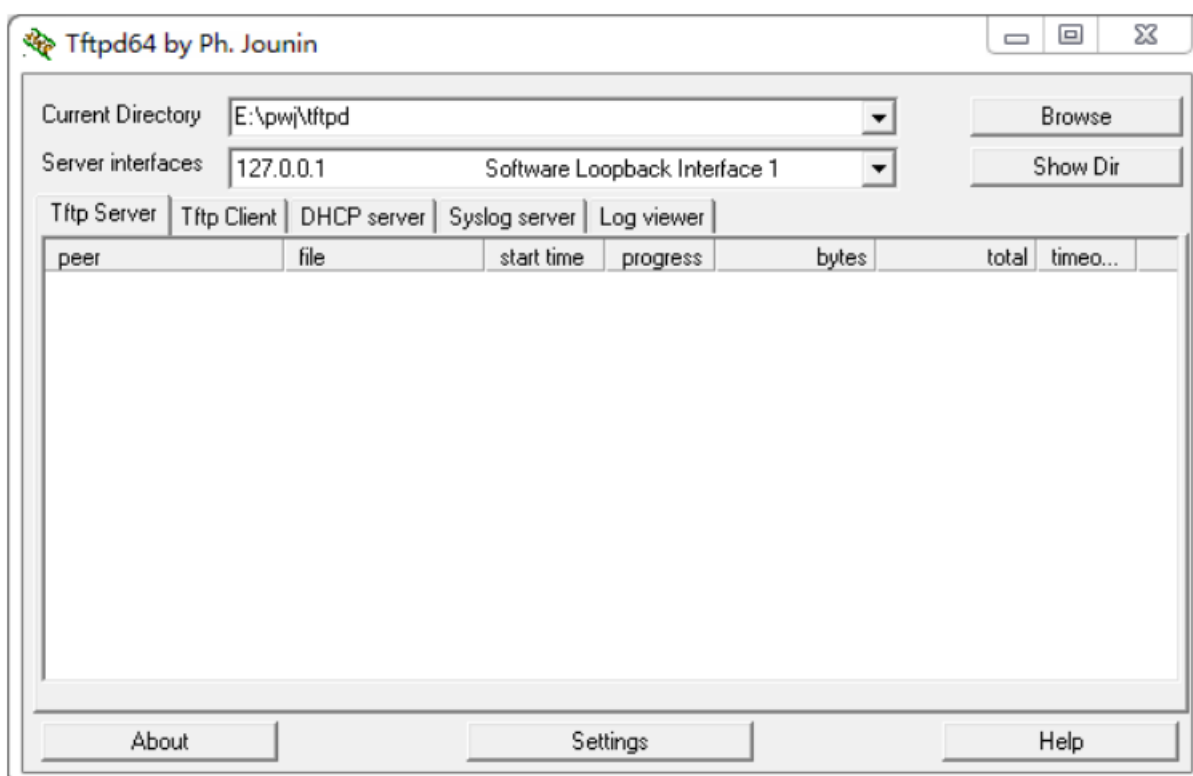


make

2. uboot upgrade

2. uboot upgrade

Setup tftp server on the development PC, and put uboot.img under tftp root directory.



Make sure connect the board to PC via serial port, reset board and press '9' when current uboot is booting up. Follow the steps in the console:

```
Date:Aug 20 2021 Time:09:10:53
=====
icache: sets:256, ways:4, linesz:32 ,total:32768
dcache: sets:256, ways:4, linesz:32 ,total:32768

#### The CPU freq = 880 MHZ ####
estimate memory size =128 Mbytes
#Reset_MT7530

Please choose the operation:
  1: Load system code to SDRAM via TFTP.
  2: Load system code then write to Flash via TFTP.
  3: Boot system code via Flash (default).
  4: Entr boot command line interface.
  7: Load Boot Loader code then write to Flash via Serial.
  9: Load Boot Loader code then write to Flash via TFTP.
default: 3
You chose 9
0

9: System Load Boot Loader then write to Flash via TFTP.
warning!! Erase Boot Loader in Flash then burn new one. Are you sure?(Y/N)
Please Input new ones /or Ctrl-C to discard
Input device IP (192.168.1.190) ==:192.168.1.190
Input server IP (192.168.1.3) ==:192.168.1.3
Input Uboot filename () ==:uboot.img
```

The uboot.img will be downloaded to the board and upgraded, then automatically reset to boot new uboot:

```
serial-com3-57600 x
You chose 9
0

9: System Load Boot Loader then write to Flash via TFTP.
warning!! Erase Boot Loader in Flash then burn new one. Are you sure?(Y/N)
Please Input new ones /or Ctrl-C to discard
Input device IP (192.168.1.190) ==:192.168.1.190
Input server IP (192.168.1.3) ==:192.168.1.3
Input Uboot filename () ==:uboot.img

NetLoop,call eth_halt !
NetLoop,call eth_init !
Trying Eth0 (10/100-M)

waiting for RX_DMA_BUSY status Start... done

ETH_STATE_ACTIVE!!
TFTP from server 192.168.1.3; our IP address is 192.168.1.190
Filename 'uboot.img'.

TIMEOUT_COUNT=10,Load address: 0x84000000
Loading: Got ARP REPLY, set server/gtwy eth addr (24:4b:fe:98:c0:5c)
Got it
#####
done
Bytes transferred = 130648 (1fe58 hex)
LoadAddr=84000000 NetBootFileXfersize= 0001fe58
..ranand_erase: start:0, len:20000
.(5192)offs=0 piece=0 piece_size=130648 rc=0
Done!

=====
MT7621 stage1 code 10:33:11 (ASIC)
CPU=50000000 HZ BUS=12500000 HZ
=====

Change MPLL source from XTAL to CR...
do MEMPLL setting..
MEMPLL Config : 0x31100000
3PLL mode + External loopback
=== XTAL-40Mhz === DDR-800Mhz ===
```

9. System firmware upgrade

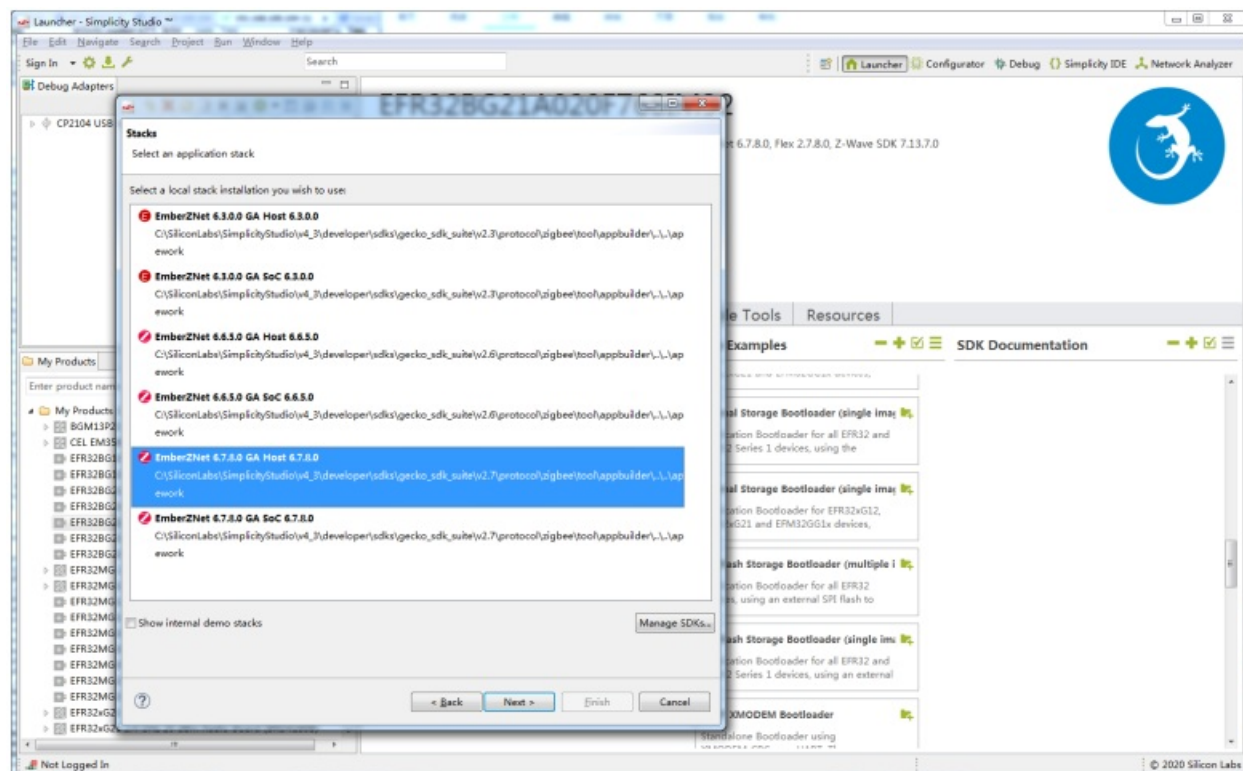
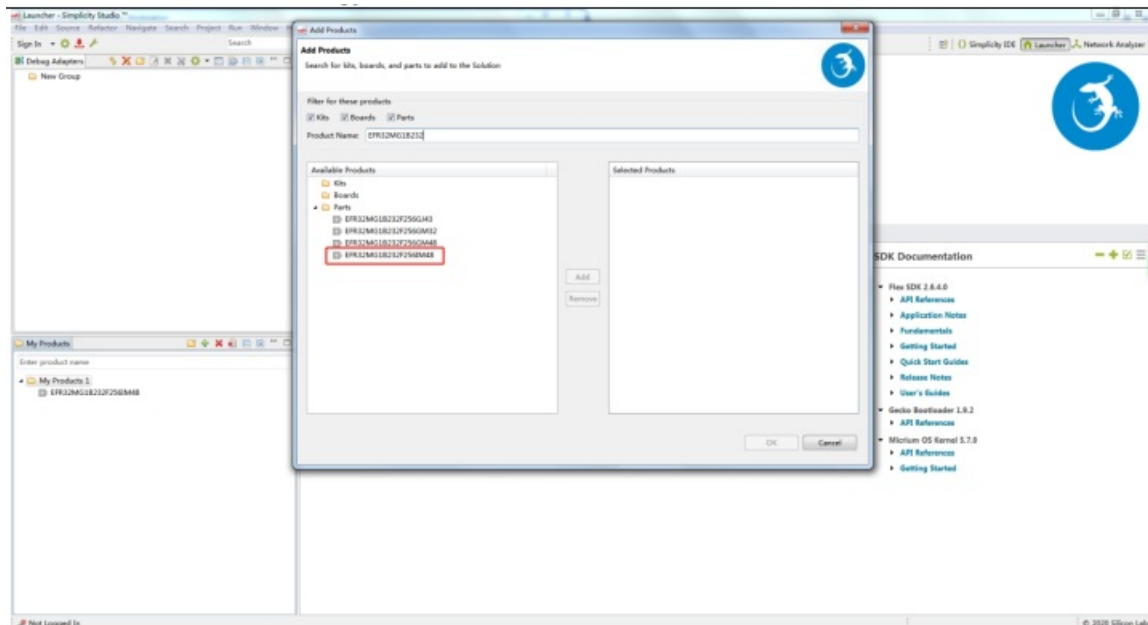
Put the image on any server which supports ssh/scp service.

```
scp user@serverip:~/openwrt-ramips-mt7621-mt7621-squashfs-sysupgrade.bin /tmp/ sysupgrade -v
```

/tmp/openwrt-ramips-mt7621-mt7621-squashfs-sysupgrade.bin

10. Communication between zigbee module

User can obtain the zigbee module's NCP program in simplicity studio, the module number is **EFR32MG1B232F256G**



For detailed information to flash image to the zigbee module, please refer to document **ZIGBEE MODULE FLASH FIRMWAREv1.0**

For SDK to develop program in gateway, please refer to document **API Reference for EmberZNet PC Host**

It can be found in the ss5's directory of

C:\SiliconLabs\SimplicityStudio\v4_3\developer\sdk\gecko_sdk_suite\v2.7\protocol\zigbee\documentation

Silicon Labs EmberZNet Documentation

Release Notes / Quick Start Guide

• [EmberZNet PRO Release Notes](#)

Lists compatibility requirements and sources for all software components in the development environment. Discusses the latest changes to the EmberZNet PRO stack (and associated utilities) including added/deleted/deprecated features/API, and lists bugs that have been fixed since the last release and any pending ones.

• [Getting Started with EmberZNet PRO](#) – QSG106

Provides basic information on configuring, building, and installing applications for the Mighty Gecko family and EM35x using the EmberZNet PRO stack and Simplicity Studio v4.

• [Using the Silicon Labs Dynamic Multiprotocol Demonstrations](#) – QSG155

Shows how to demonstrate Dynamic Multiprotocol functionality using a Bluetooth LE smartphone app with either Zigbee-Bluetooth or RAIL-Bluetooth demo applications.

Zigbee Fundamentals

• [Zigbee Fundamentals](#) – UG103-02

Describes the key features and characteristics of a Zigbee solution. It also includes a section on Zigbee 3.0.

API References

• [API Reference for the EmberZNet SOC Platform](#)

Lists SoC Platform APIs used to interface to the EmberZNet PRO stack, HAL, and status of the application-controlled network. These APIs concern network management, device and stack management, messaging, fragmentation, serial communication, token access, peripheral access, bootload utilities, and others. They are independent of the Application Framework and therefore can be used to develop applications that do not rely on the Zigbee Cluster Library.

• [API Reference for the EmberZNet STM32F103RET Host](#)

Lists STM32F103RET Host APIs used to interface to the EmberZNet PRO stack, HAL, and status of the application-controlled network. These APIs concern network management, device and stack configuration, message fragmentation, serial communication, peripheral access, and others. They are independent of the Application Framework and therefore can be used for developing applications that do not rely on the Zigbee Cluster Library.

• [API Reference for EmberZNet PC Host](#)

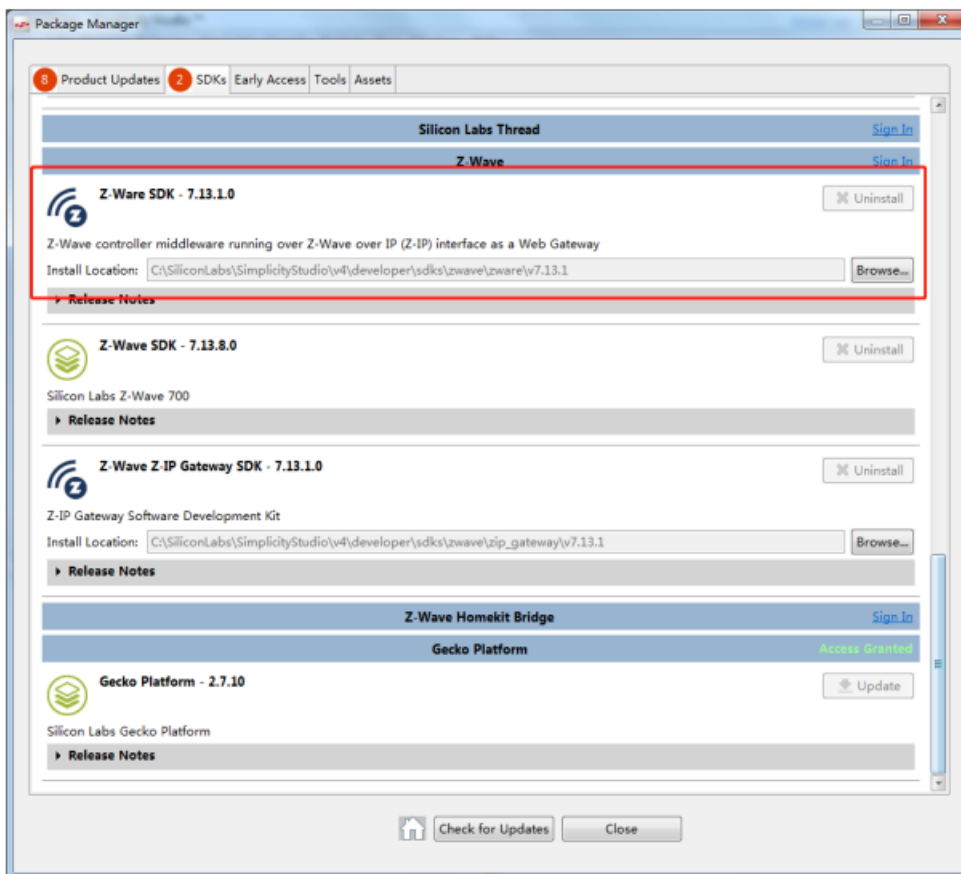
Lists PC Host APIs used to interface to the EmberZNet stack, HAL, and status of the application-controlled network. These APIs concern network management, device and stack configuration, message fragmentation, serial communication, ASH (Asynchronous Serial Host) utilities, and others. They are independent of the Application Framework and therefore can be used for developing applications that do not rely on the Zigbee Cluster Library.

• [Advanced Application Programming with the Stack and HAL APIs](#) – UG185

A companion to the EmberZNet API references, for developers whose applications require functionality not available through AppBuilder and the application framework, or who prefer working with an API. Includes an introduction to the stack API, a discussion of advanced design issues to consider when developing an application using the API, and provides an example application.

11. Communication between Z-Wave module

User can obtain the Z-Wave module's NCP program in simplicity studio, and acquire the Z-Wave sdk in siliconlabs for further development

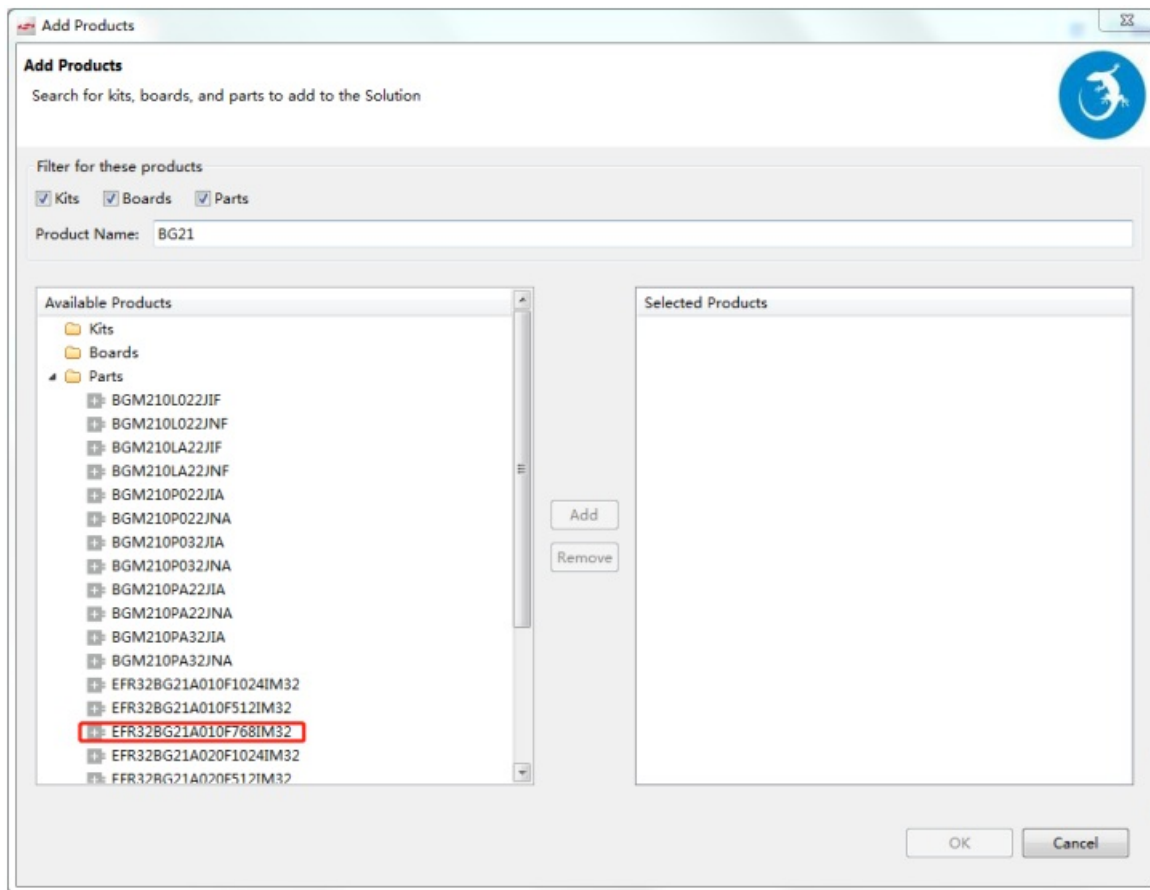


Find the correct Z-Wave module in simplicity studio, then follow the same guide in **section 10**

12. Communication between BLE module

a) For silicon labs BLE module

User can obtain the ble module's NCP program in simplicity studio, the module number is **ERF32BG21**



Find the correct BLE module in simplicity studio, then follow the same guide in **section 10**

b) For Nordic BLE module

If Gateway is implemented the Nordic BLE module NRF52840, follow the official SDK from Nordic.


<https://www.nordicsemi.com/Products/Development-software/nRF5SDK/GetStarted?lang=en#infotabs>

13. System firmware upgrade

a) Using sysupgrade in gateway

```
scp user@serverip:~/openwrt-ramips-mt7688-mt7688-squashfs-sysupgrade.bin /tmp/ sysupgrade -n  
/tmp/openwrt-ramips-mt7688-mt7688-squashfs-sysupgrade.bin
```

b) Using LUCI web server in gateway


[Home](#)
[Interface](#)
[IOT Services](#)
[Application Solution](#)
[Advance](#)
[Logout](#)

Flash operations

[Actions](#)
[Configuration](#)

Backup

Click "Generate archive" to download a tar archive of the current configuration files.

[Download backup](#)
[Generate archive](#)

Restore

To restore configuration files, you can upload a previously generated backup archive here. To reset the firmware to its initial state, click "Perform reset" (only possible with squashfs images).

Reset to defaults

Perform reset

Restore backup

Upload archive...

Custom files (certificates, scripts) may remain on the system. To prevent this, perform a factory-reset first.

Save mtblock contents

Click "Save mtblock" to download specified mtblock file. (NOTE: THIS FEATURE IS FOR PROFESSIONALS!)

Choose mtblock

u-boot

[Download mtblock](#)
[Save mtblock](#)

Flash new firmware image

Upload a sysupgrade-compatible image here to replace the running firmware.

Image

Flash image...

[Logo Setting](#)
[System](#)
[Administration](#)
[Diagnostics](#)
[Backup / Flash Firmware](#)
[Reboot](#)

c) Using Uboot method

Please refer to document **update gateway using uboot.pdf**

14. Others

For more information, please visit OpenWRT official website.


Tel:86-571-86769027/8 8810480

Website:www.dusuniot.com





www.dusunremotes.com

Floor8,buildingA,Wantongcenter,H
angzhou 310004,
chinawww.dusunlock.com

Documents / Resources

	<p>DUSUN DSGW-021 L-Serial Smart Gateway [pdf] User Guide</p> <p>DSGW-021 L-Serial Smart Gateway, DSGW-021, L-Serial Smart Gateway, Smart Gateway, Gateway</p>
---	---

References

-  [Dusun IoT: IoT Gateway Hardware Suppplier & Solutions Vendor- DusunIoT](#)
-  [Dusunremotes | Custom Intelligent Remote Control Manufacturer](#)
-  [GitHub - openwrt/openwrt: This repository is a mirror of https://git.openwrt.org/openwrt/openwrt.git](#)
[It is for reference only and is not active for check-ins. We will continue to accept Pull Requests here.](#)
[They will be merged via staging trees then into op](#)
-  [nRF5 SDK get started - nordicsemi.com](#)

Manuals+.