Development Document

Product Name: DSGW-210-Ubuntu

Revision History

Specification		Sect.	Update Description	By
Rev.	date			
1.0	2023-04.17		Release	AU

Approvals

Organization	Name	Title	Date

Contents

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
- how to modify and 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-210 gateway for Ubuntu.

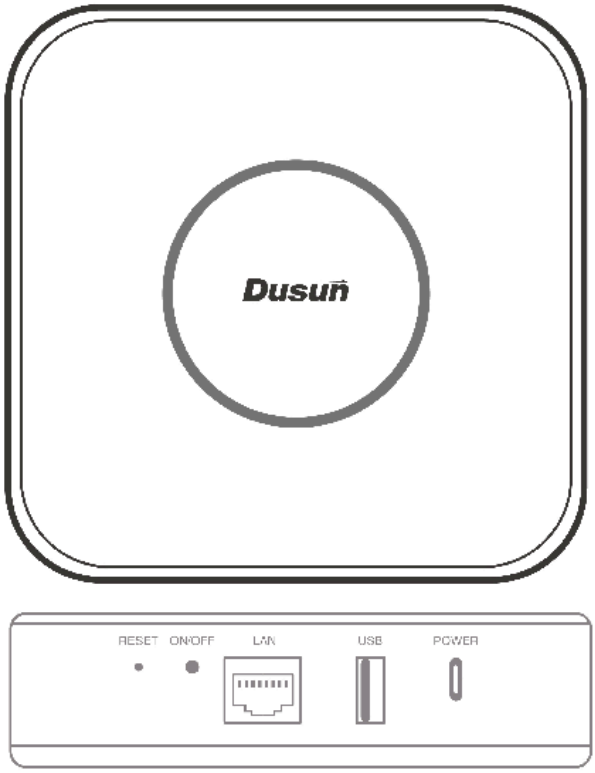
2. Gateway Information

This section describes the Gateway's basic resource information and interfaces.

2.1. Basic Information

- Processor: RK3328 (ARM64)
- Supply: DC-5V/2A
- RAM: DDR2 2G
- eMMC: 8G/16G/32G
- Ethernet: WAN RJ45/10M/100M
- Bluetooth: ERF32BG21
- Zigbee: EFR32MG1B232
- Z-Wave: ZG130S
- LTE: EG25
- LoraWan: SX1302
- USB
- Wi-Fi: RTL8821CS

2.2. Interface



3. Debug Setup

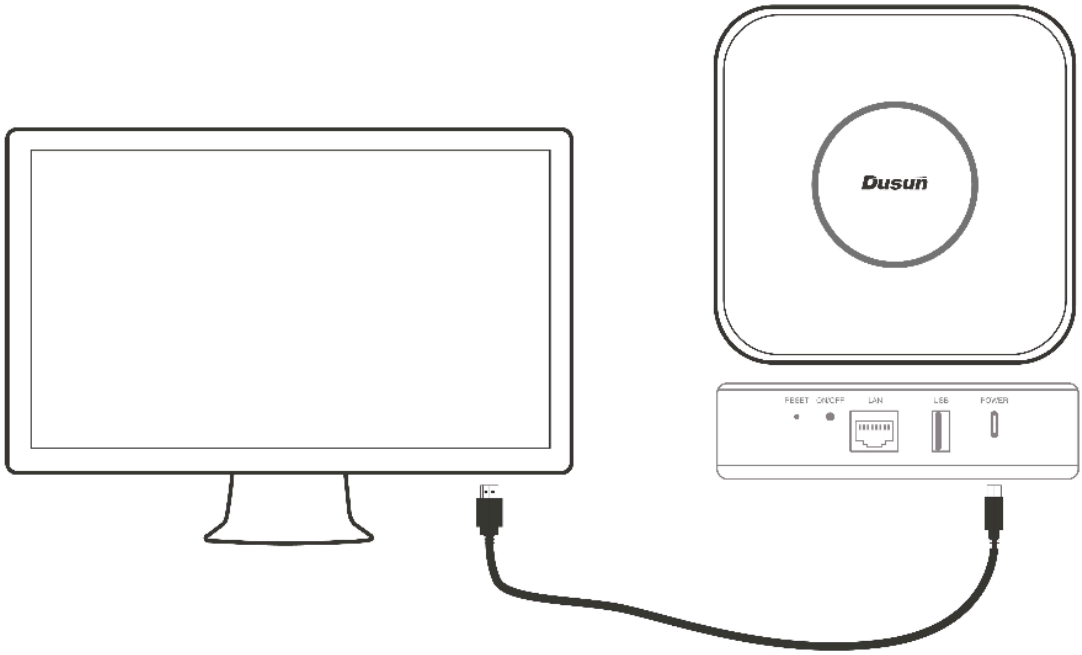
This section describes how to connect the Gateway to your host computer and network to debug for development.

3.1. Power

- Make sure that the power adapter is 5V/2A.
- 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.
- Connect the output plug of the power supply to the gateway

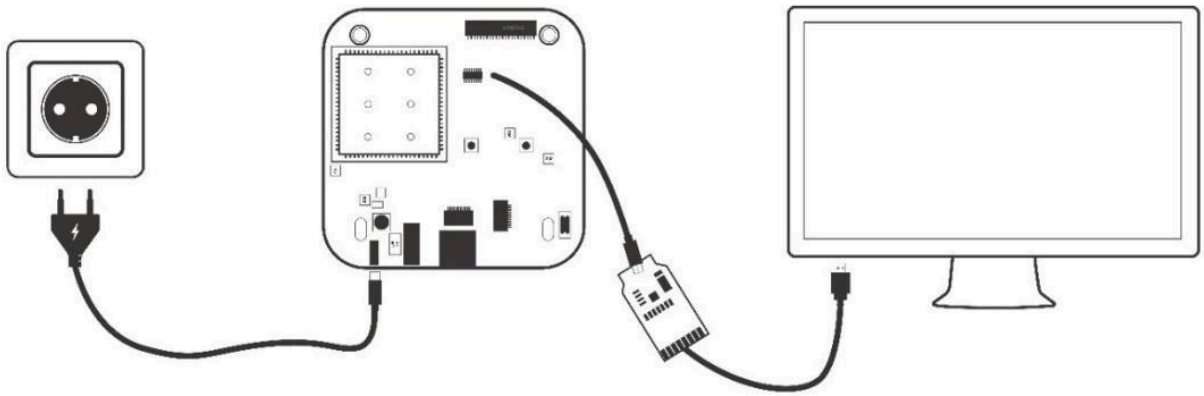
3.2. Wire Connect

Connect Gatewayy to a router for login

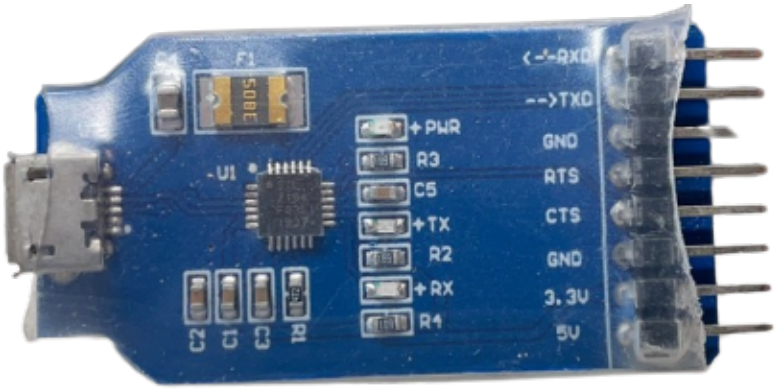


3.3. Debug UART Connect

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



- PCB serial port gateways




```
1. USB-to-serial bridge. Serial port setting:
2. Baud rate: 115200
3. Bits: 8
4. Stop Bits: 1
5. Hardware flow control: None
```

4. SDK Download And Compile

This section describes how to download the SDK and compile it.

4.1. SDK Environment Prepare

Compilation environment: Ubuntu18.04

- Make sure your machine have enough RAM (>2G)

4.2. SDK Download

Get the source code from Dusun FTP server uncompress it under your work directory.
For example:

```
1. mkdir -p ~/workdir/ubuntu210
2. tar zxvf DSGW-210_sdk-ubuntu_AV1.0.0.2.tar.gz -C /workdir/ubuntu210
3. cd ~/workdir/ubuntu210
```

4.3. SDK Compile

```
1. cd ~/workdir/ubuntu210
2. ./build.sh
```

4.4. SDK Output

```
Update.img /*The System Upgrade bin*/
```

```
[au@git rkbuilder]$ ls ./rockdev/ -alh
total 358M
drwxr-xr-x 2 au au 4.0K May 11 12:11 .
drwxr-xr-x 5 au au 4.0K Mar 29 11:57 ..
-rw-r--r-- 1 au au 22M May 11 12:11 boot.img
-rwxr-xr-x 1 au au 0 Mar 29 11:47 FunListTest.sh
-rw-r--r-- 1 au au 187K Mar 29 11:47 MiniLoaderAll.bin
-rwxr-xr-x 1 au au 48K Mar 29 11:47 misc.img
-rw-r--r-- 1 au au 10M Mar 29 11:47 oem.img
-rw-r--r-- 1 au au 512 Mar 29 11:47 parameter.txt
-rw-r--r-- 1 au au 29M Mar 29 11:47 recovery.img
lrwxrwxrwx 1 au au 12 Mar 29 11:55 rootfs.ext4 -> ./rootfs.img
-rw-r--r-- 1 root root 285M May 11 12:11 rootfs.img
-rw-r--r-- 1 au au 4.0M Mar 29 11:47 trust.img
-rw-r--r-- 1 au au 4.0M Mar 29 11:47 uboot.img
-rw-r--r-- 1 au au 5.0M Mar 29 11:47 userdata.img
```

5. Firmware Program And Program

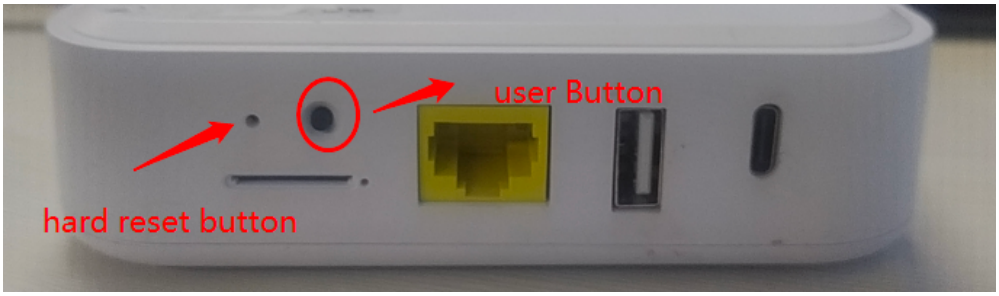
5.1. Firmware Program

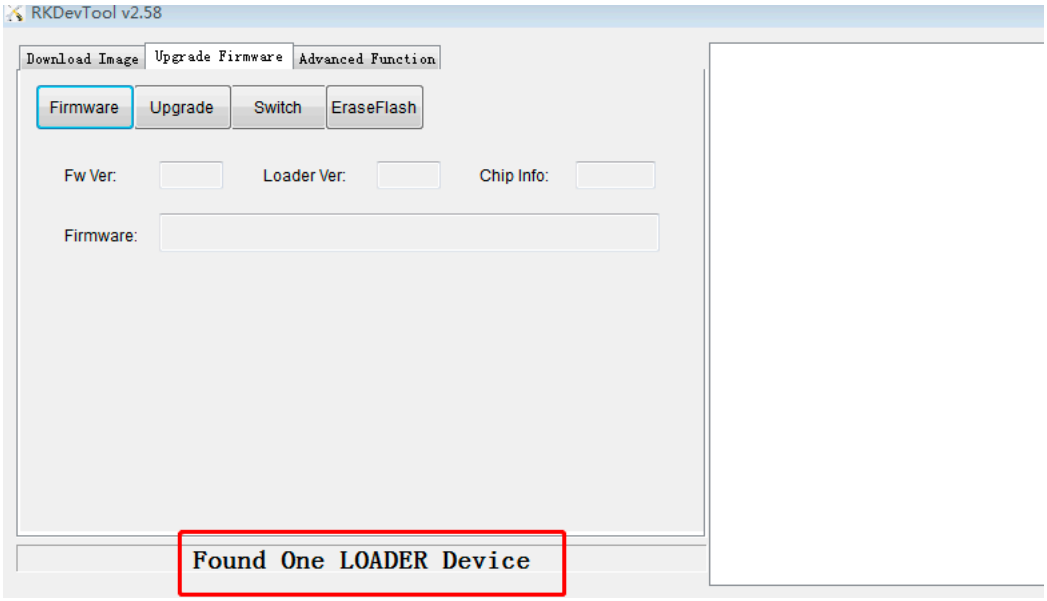
5.1.1. USB OTG Program

- Upgrade Tool

```
1. AndroidTool_Release_v2.69
```

Press the long button and reset the the board to let it go to Loader/Maskrom Mode





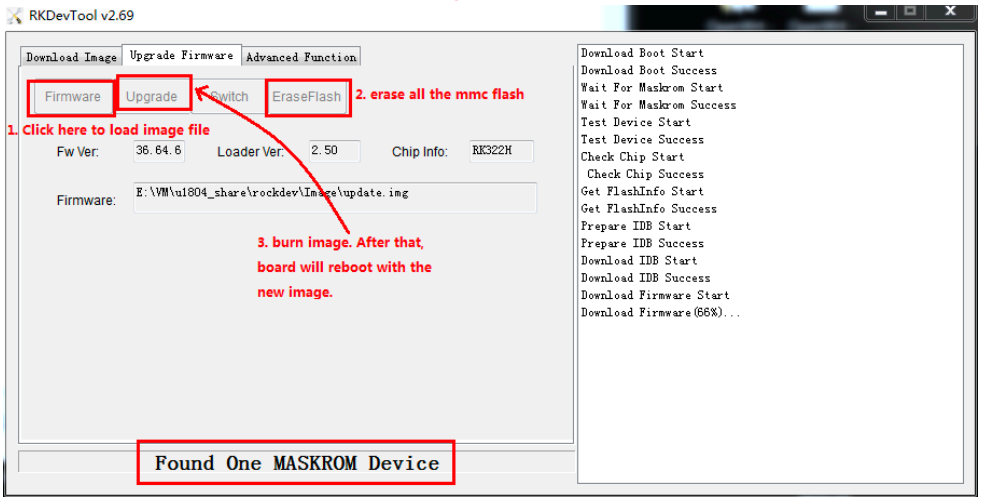
- Use Serial Command to let it to goto Loader/Maskrom Mode

Press "Ctrl+C" when uboot is booting up, to enter uboot:

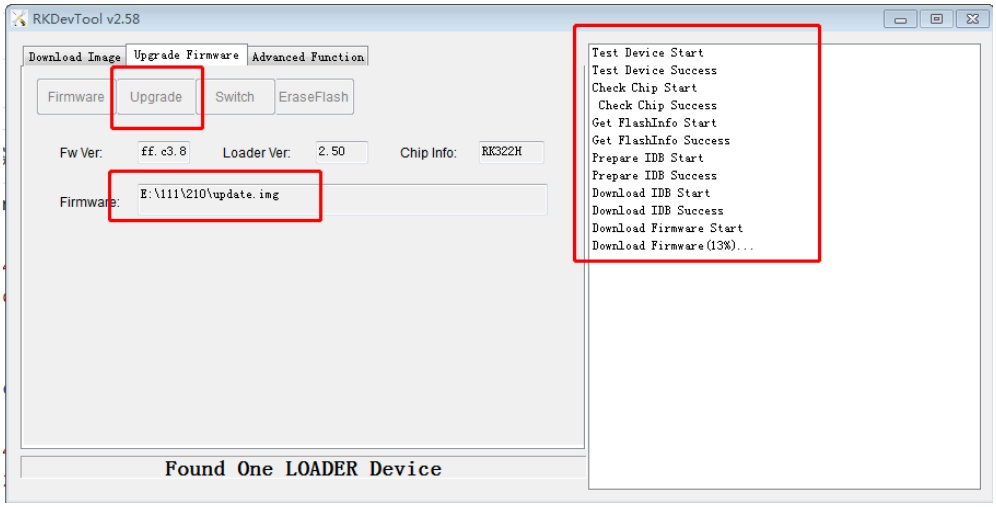
```
INFO:      CPU Node : MPID 0xffffffffffffffff, parent_node 1, State OFF (0x2)
~zV▷□□0<9□□/6~(O}G□c U-Boot 2017.09 (Aug 02 2021 - 18:45:17 +0800)e OFF (0x2)

Model: Rockchip RK3328 EVB
PreSerial: 2
DRAM:  2 GiB
Sysmem: init
Relocation Offset is: 7dbed000
Using default environment

rksdmmc@ff500000: 1, rksdmmc@ff520000: 0
Bootdev(atags): mmc 0
MMC0: High Speed, 52Mhz
PartType: EFI
boot mode: normal
Found DTB in boot part
DTB: rk-kernel.dtb
Android header version 0
Model: Rockchip RK3328 EVB
CLK: (sync kernel. arm: enter 600000 KHz, init 600000 KHz, kernel 0N/A)
  apll 400000 KHz
  dpll 664000 KHz
  cpll 1200000 KHz
  gpll 491009 KHz
  npll 600000 KHz
  armclk 600000 KHz
  aclk_bus 150000 KHz
  hclk_bus 75000 KHz
  pclk_bus 75000 KHz
  aclk_peri 150000 KHz
  hclk_peri 75000 KHz
  pclk_peri 75000 KHz
Net:   Net Initialization Skipped
No ethernet found.
gpio: pin 54 (gpio 54) value is 0
gpio: pin 58 (gpio 58) value is 1
gpio: pin 56 (gpio 56) value is 1
Setting bus to 0
I2C0 speed: 100000Hz
Hit key to stop autoboot('CTRL+C'):  0
=> <INTERRUPT>
=> <INTERRUPT>
=> <INTERRUPT>
=>
```

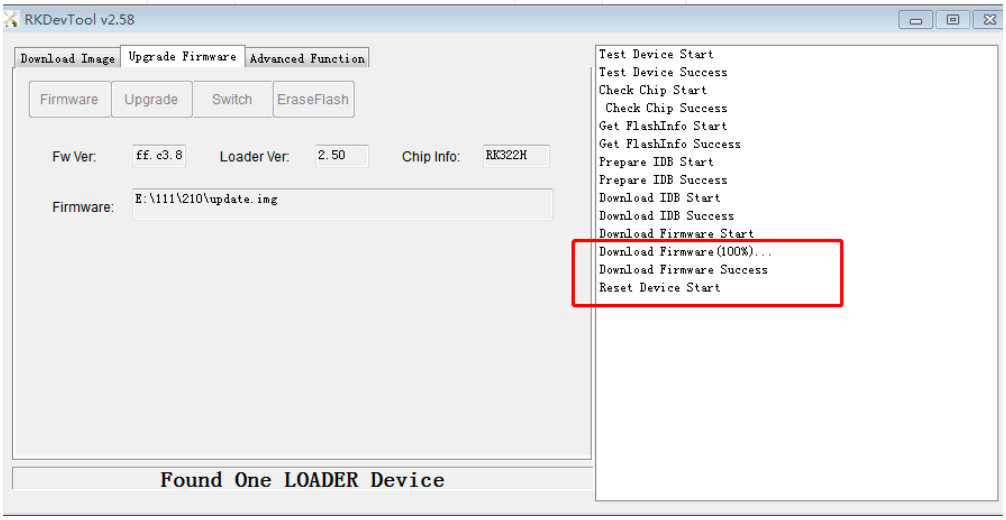
"rockusb 0 mmc 0" command to reboot board to loader mode “rbrom” rbrom commmand to reboot board to maskrom mode

- Select The Firmare update.img



- Press Upgrade to program the board

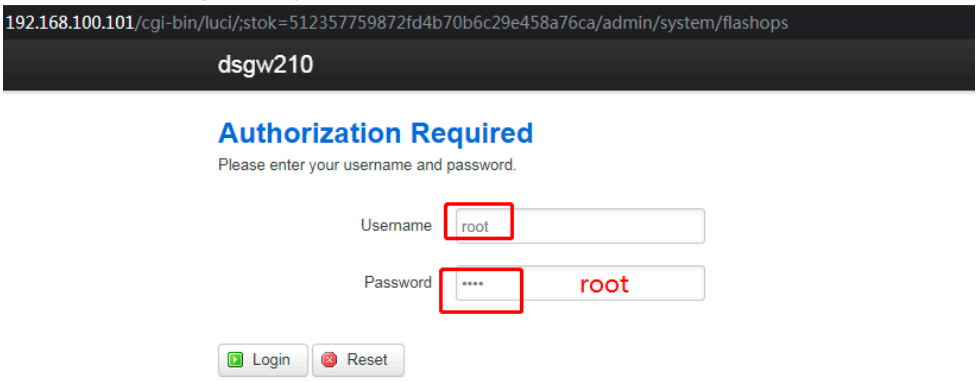




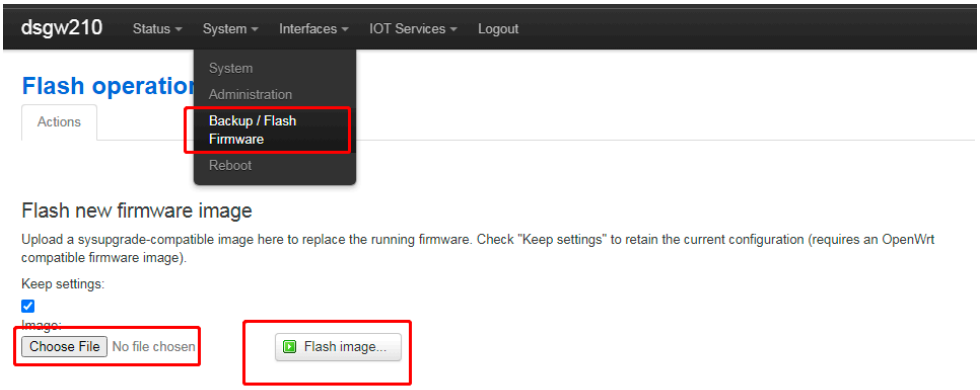
5.2. Firmware Upgrade

5.2.1. System Web Upgrade

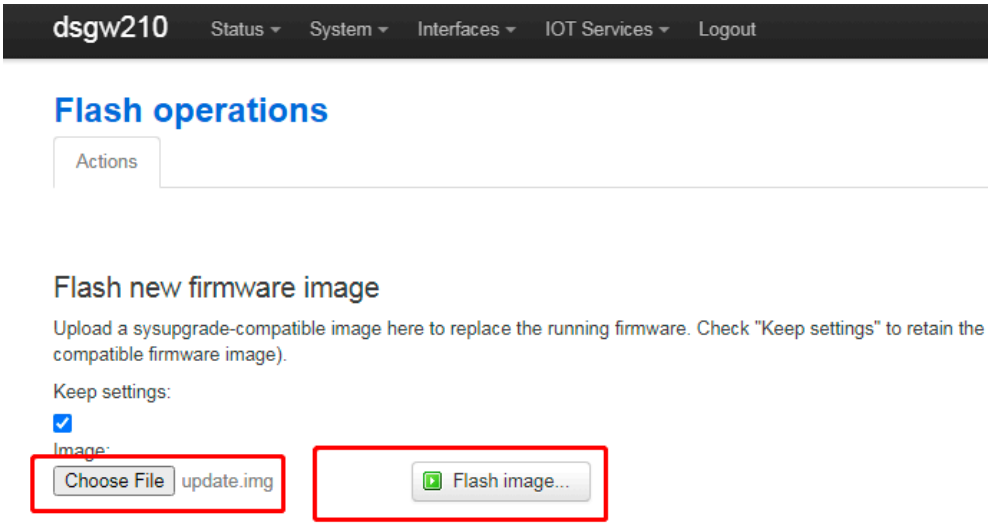
Browser input gateway IP address, Account root, Password root

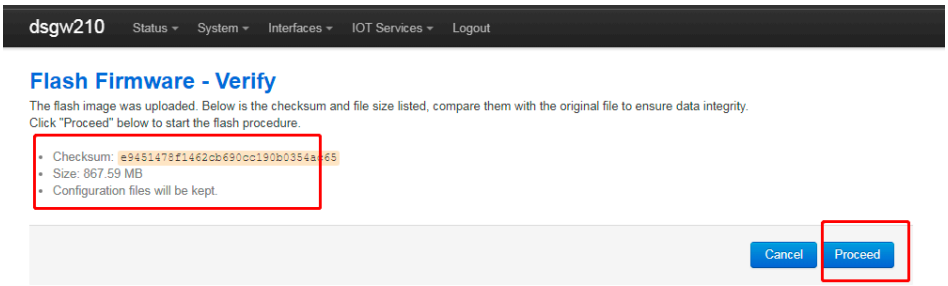


Enter Advance->Backup And Flash Firmware



Select update.img Upgrade the firmware for upgrading, upload will take a few minitues; please wait.





5.2.2. System Command Upgrade

- use scp or winscp tool to put the fw.bin to the board's tmp

```
1. scp update.img root@192.168.xxx.xxx:/userdata/
```

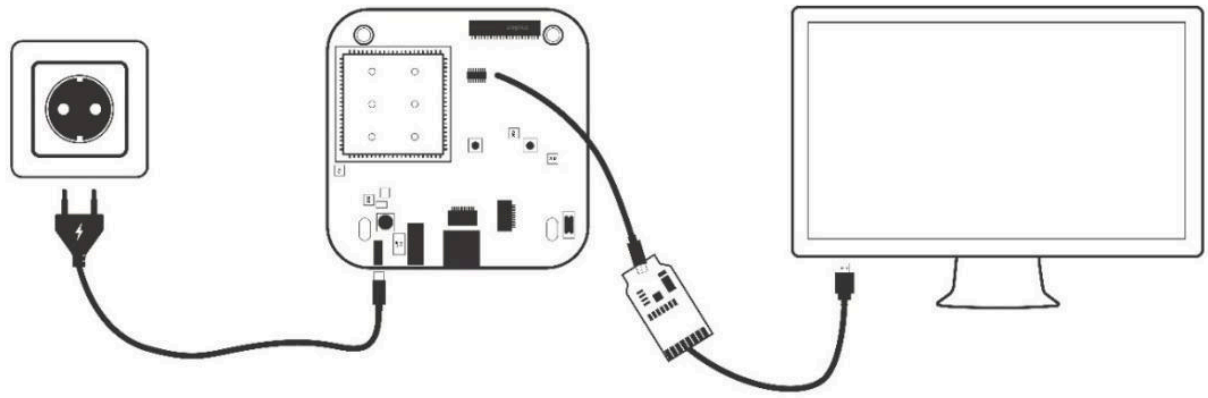
- run sysupgrade command to upgrade the firmware

```
1. sysupgrade -n -F /userdata/update.img
```

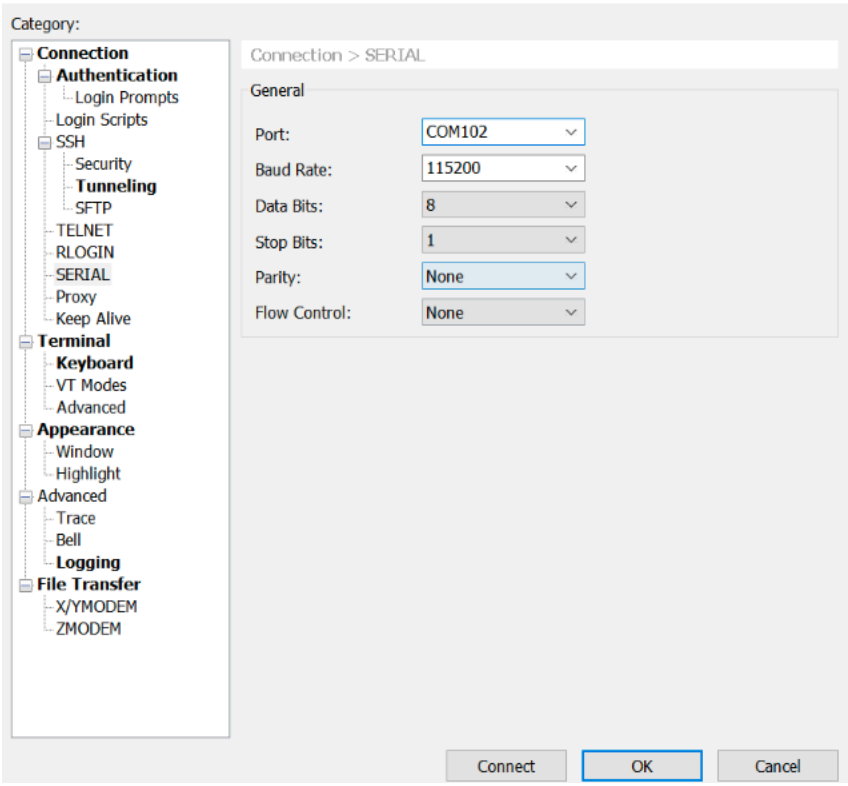
6. Gateway Login

6.1. Login Via Debug UART

- Connect uart serial tool to the board's debug uart port



- Config the serial tools's uart config



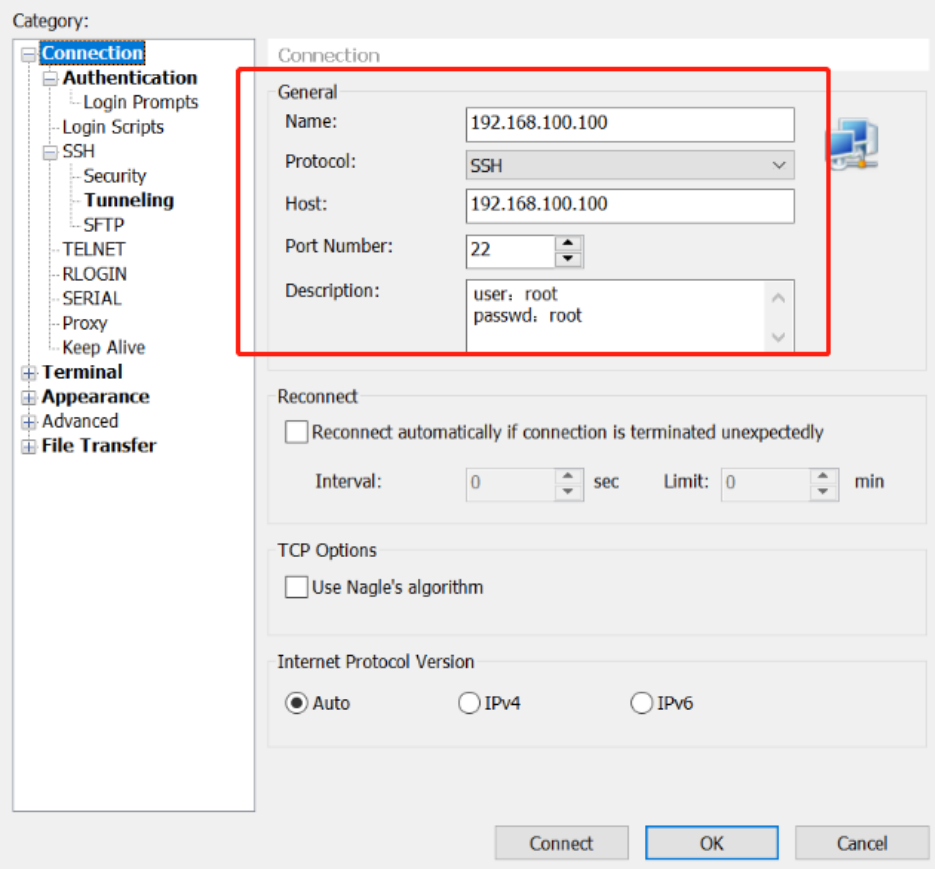
- Power on the gateway

```
1. U-Boot 2017.09 (Sep 14 2022 - 01:47:34 +0000)
2. Model: Rockchip RK3328 EVB
3. PreSerial: 2
4. DRAM:  2 GiB
5. Sysmem: init
6. Relocation Offset is: 7dbf1000
7. Using default environment
8. rksdmmc@ff500000: 1, rksdmmc@ff520000: 0
9. Bootdev(atags): mmc 0
10.MMC0: High Speed, 52Mhz
11.PartType: EFI
12.boot mode: normal
13.Found DTB in boot part
14.DTB: rk-kernel.dtb
15.Android header version 0
16.Model: Rockchip RK3328 EVB
17.CLK: (sync kernel. arm: enter 600000 KHz, init 600000 KHz, kernel 0N/A)
18.  apll 400000 KHz
19.  dpll 664000 KHz
20.  cpll 1200000 KHz
```

```
  gpll 491009 KHz
  npll 600000 KHz
  armclk 600000 KHz
  aclk_bus 150000 KHz
  hclk_bus 75000 KHz
  pclk_bus 75000 KHz
  aclk_peri 150000 KHz
  hclk_peri 75000 KHz
  pclk_peri 75000 KHz
Net:   Net Initialization Skipped
No ethernet found.
Hit key to stop autoboot('CTRL+C'):  0
ANDROID: reboot reason: "(none)"
Not AVB images, AVB skip
Fdt Ramdisk skip relocation
Booting IMAGE kernel at 0x00280000 with fdt at 0x8300000...
## Booting Android Image at 0x0027f800 ...
Kernel load addr 0x00280000 size 21515 KiB
## Flattened Device Tree blob at 08300000
   Booting using the fdt blob at 0x8300000
   XIP Kernel Image ... OK
   Using Device Tree in place at 0000000008300000, end 0000000008313f50
Adding bank: 0x00200000 - 0x08400000 (size: 0x08200000)
Adding bank: 0x0a200000 - 0x80000000 (size: 0x75e00000)
Total: 841.792 ms
Starting kernel ...
[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Initializing cgroup subsys cpuset
[    0.000000] Initializing cgroup subsys cpu
[    0.000000] Initializing cgroup subsys cpuacct
```

6.2. Login Via Network(SSH)

Configure the SSH connection parameters



Connect success

```
root@192.168.0.230's password:
root@dsgw210:~# ps
```

7. Easy Function Test Script

- Download the test script

```
rm -rf /tmp/tools/;wget http://114.215.195.44:8080/au/gwtest/DSGW-210-Yocto.tar.gz -O /tmp/x; tar xvf /tmp/x -C /;
```

- Run the test script

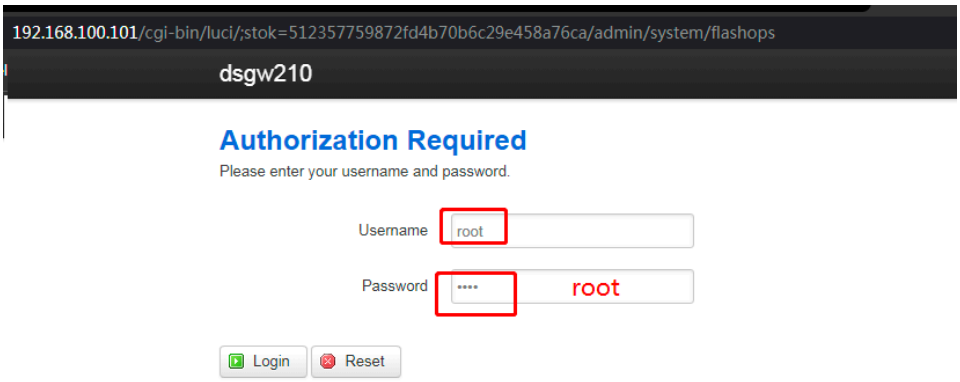
```
root@dsgw210:~# /tmp/tools/test.sh
Stopping Done: done.
Stopping dial: bg96.
Stopping Zigbee: AmberGwZ3.
Stopping Bluetooth: bul.
Stopping zwave: zwdevd.
=====================================================
Testing [   version]..., please wait...
BUILD_VERSION=V1.0.0.0-yocto_b7130bf
BUILD_TIME=Mon Mar  6 12:23:42 CST 2023
BUILD_USER=au
BUILD_HOST=git.roombanker.cn
VOIMI_VERSION=4.0.0.29
        Test Result : OK
=====================================================
Testing [       wan]..., please wait...
        Test Result : OK
=====================================================
Testing [       led]..., please wait...
        Test Result : OK
=====================================================
Testing [       btn]..., please wait...
        Please Press the Hole button ...
        Test Result : OK
=====================================================
Testing [    zigbee]..., please wait...
power on zigbee..
/dev/ttyUSB1
ezsp ver 0x08 stack type 0x02
        Test Result : OK
=====================================================
```

```
Testing [       ble]..., please wait...
power on ble..
/dev/ttyUSB0
        Test Result : OK
====================================================
Testing [     zwave]..., please wait...
power on zwave..
/dev/ttyS1
Vesion: Z-Wave 7.18
        Test Result : OK
====================================================
Testing [      lora]..., please wait...
/home/root
        Test Result : FAIL
====================================================
Testing [       usb]..., please wait...
        Test Result : FAIL
====================================================
Testing [    sdcard]..., please wait...
        Test Result : FAIL
====================================================
Testing [       rtc]..., please wait...
Wed Jan  1 00:00:00 UTC 2003
2003-01-01 00:00:02.010481+00:00
Mon Apr 17 09:04:09 UTC 2023
2003
        Test Result : OK
====================================================
Testing [       lte]..., please wait...
power on lte..
/dev/ttyUSB2
APP RDY
AT+QGMR
BG96MAR02A07M1G_01.017.01.017
OK
AT+CPIN?
+CME ERROR: 10
AT+QCCID
+CME ERROR: 13
AT+CSQ
+CSQ: 99,99
OK
```

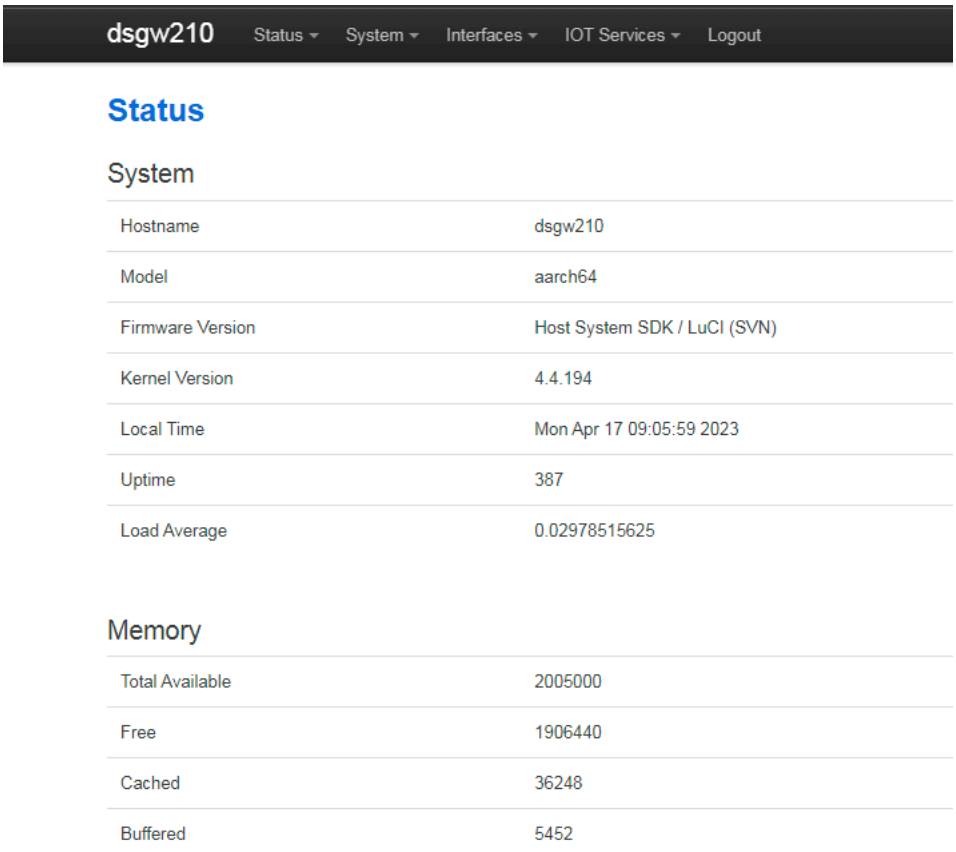
```
74.         Test Result : OK
75.================================================
76.Testing [    wifi24]..., please wait...
77.         Test Result : OK
```

8. Luci Web Function Description

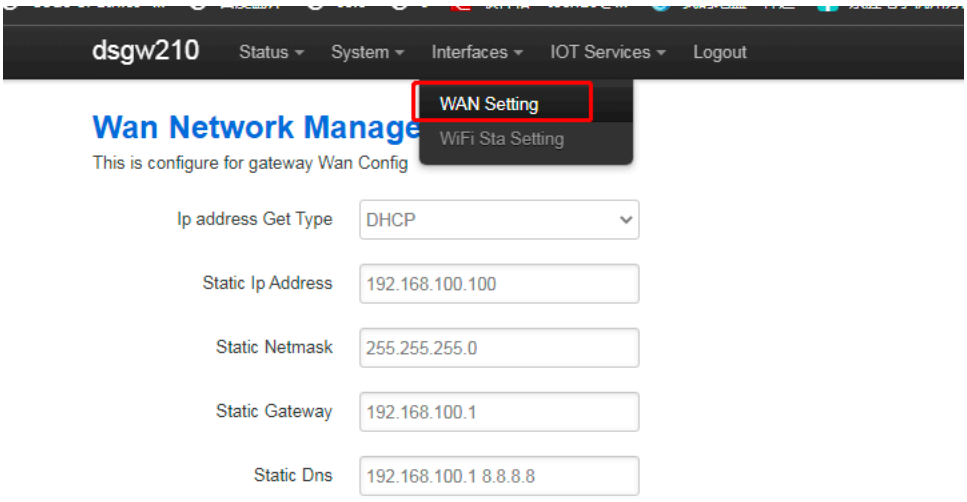
login in use user(root) and password(root)



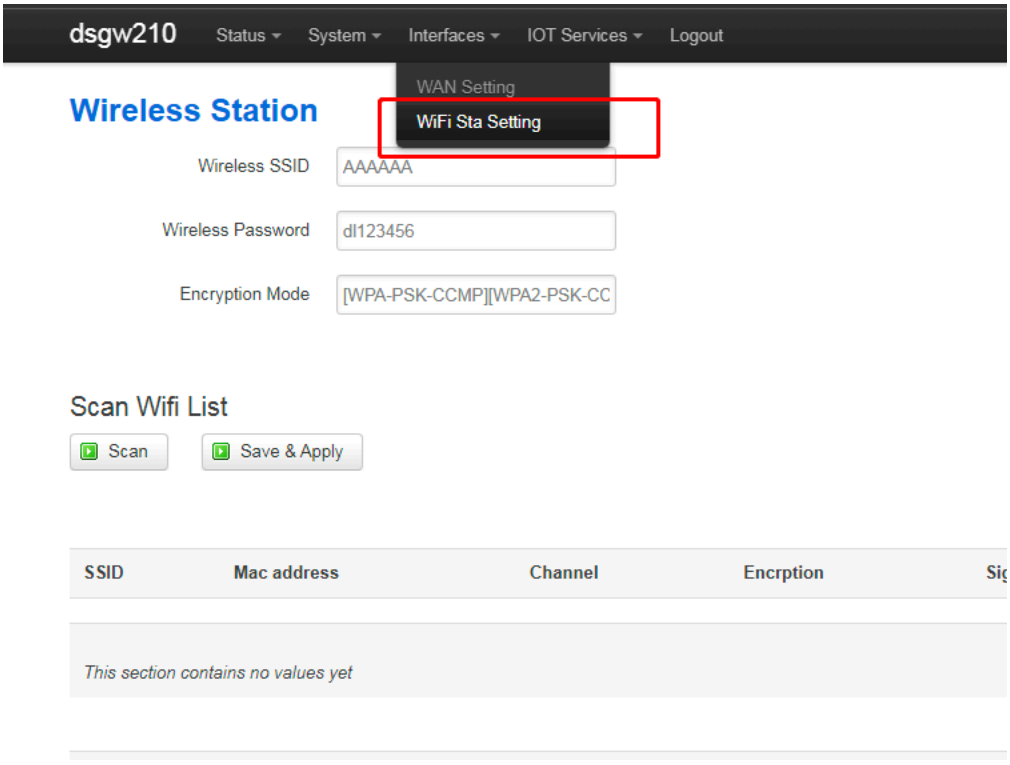
Home Page



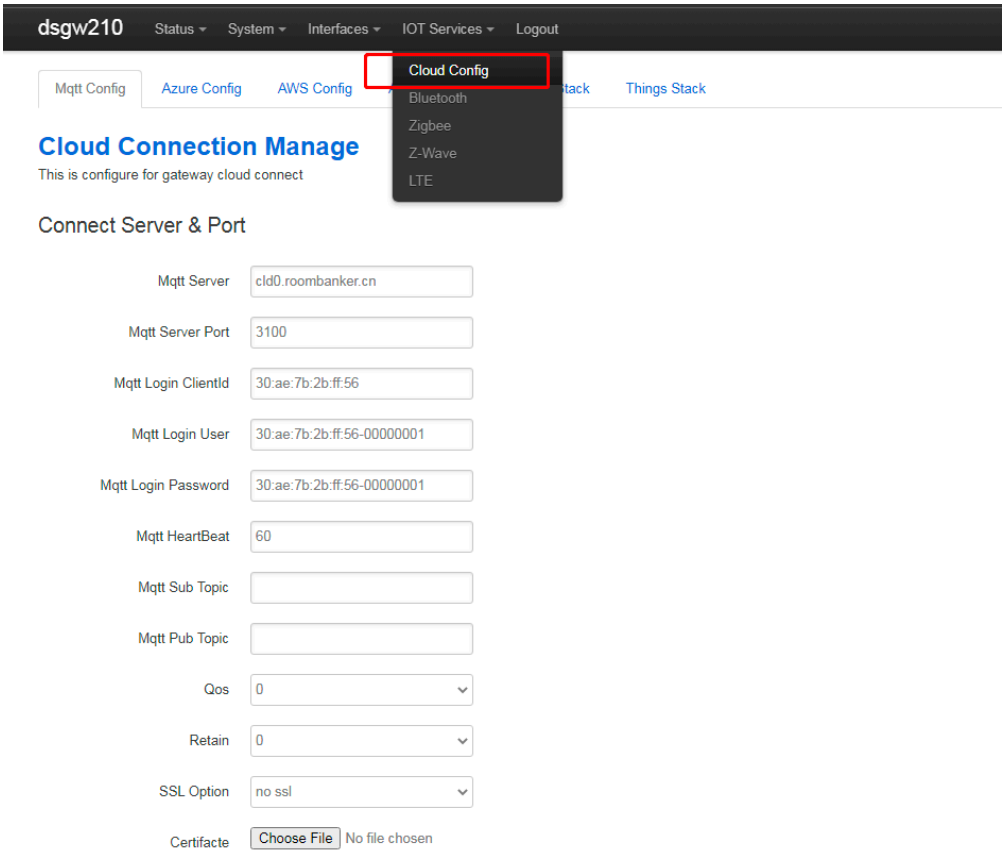
Wan Setting Page



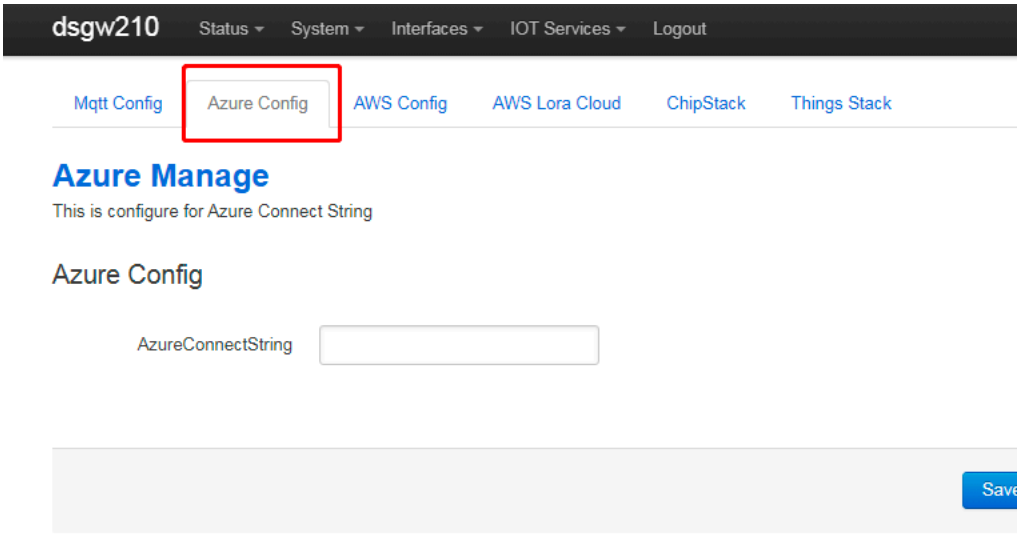
Wi-Fi Setting Page



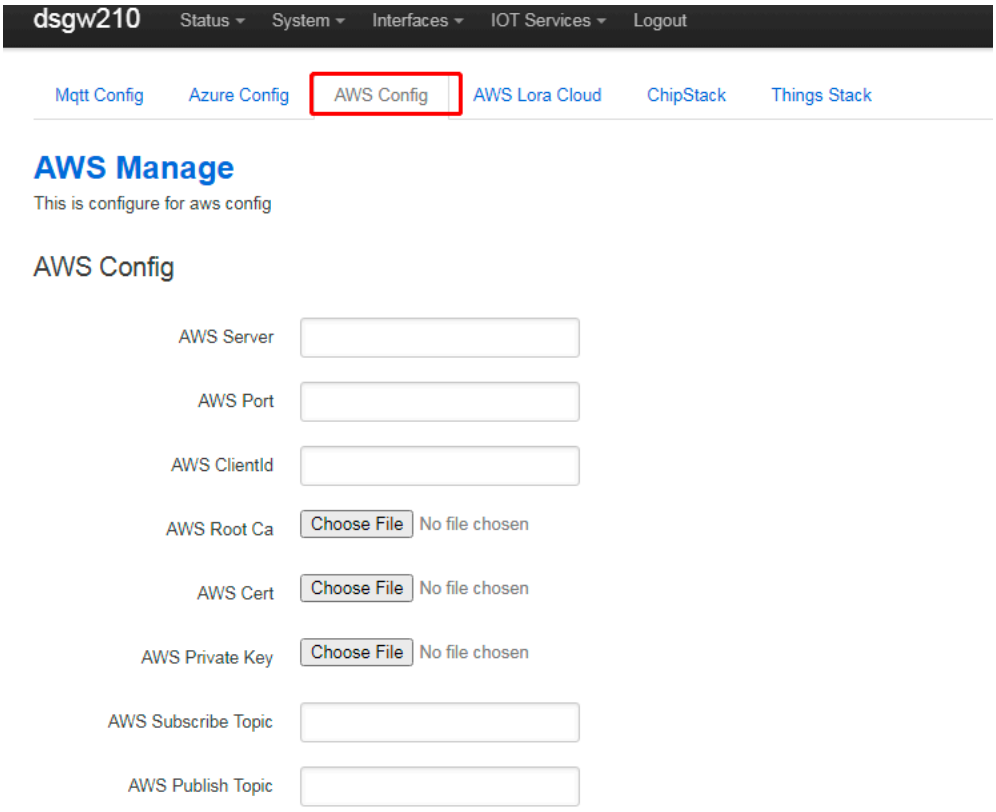
Cloud Mqtt Config Page



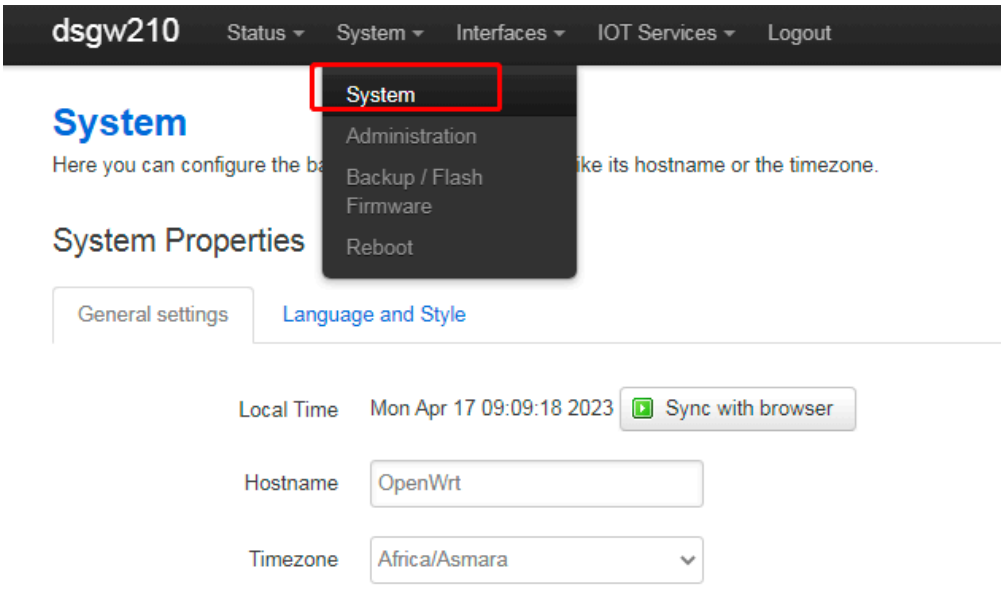
Cloud Azure Config Pag



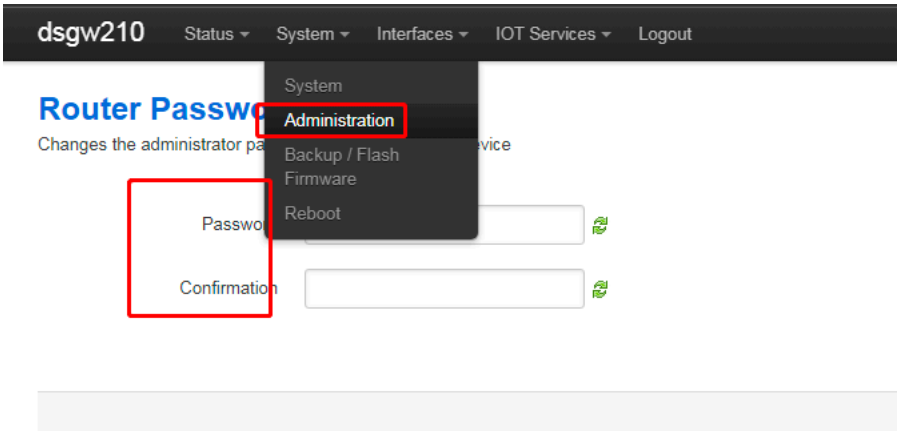
Cloud AWS Mqtt Config Page



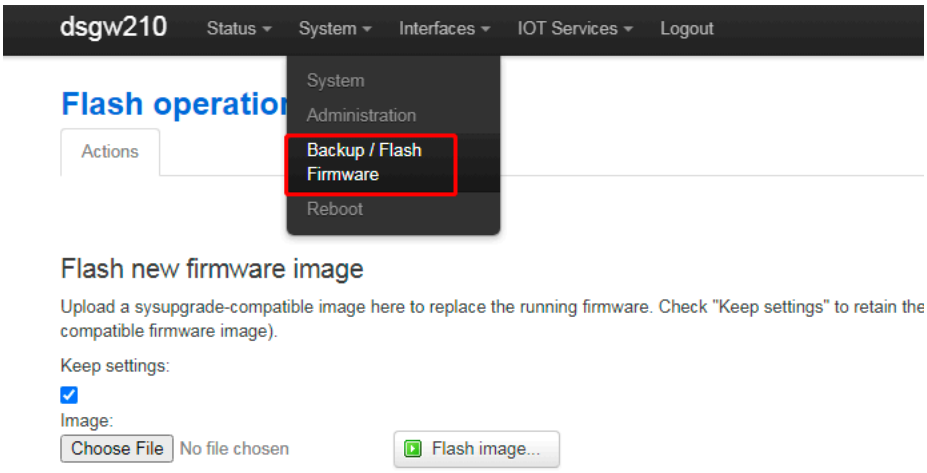
System Setting Page



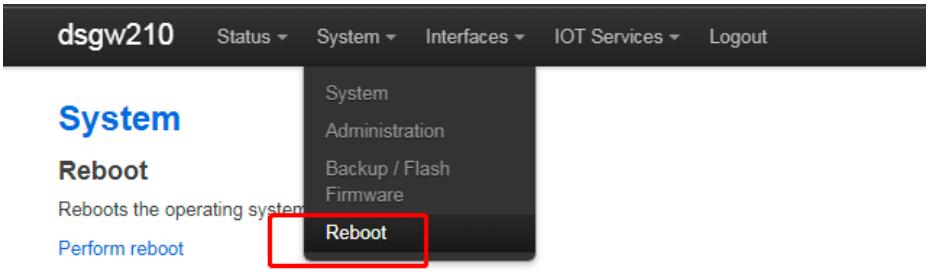
System Administrator Setting page



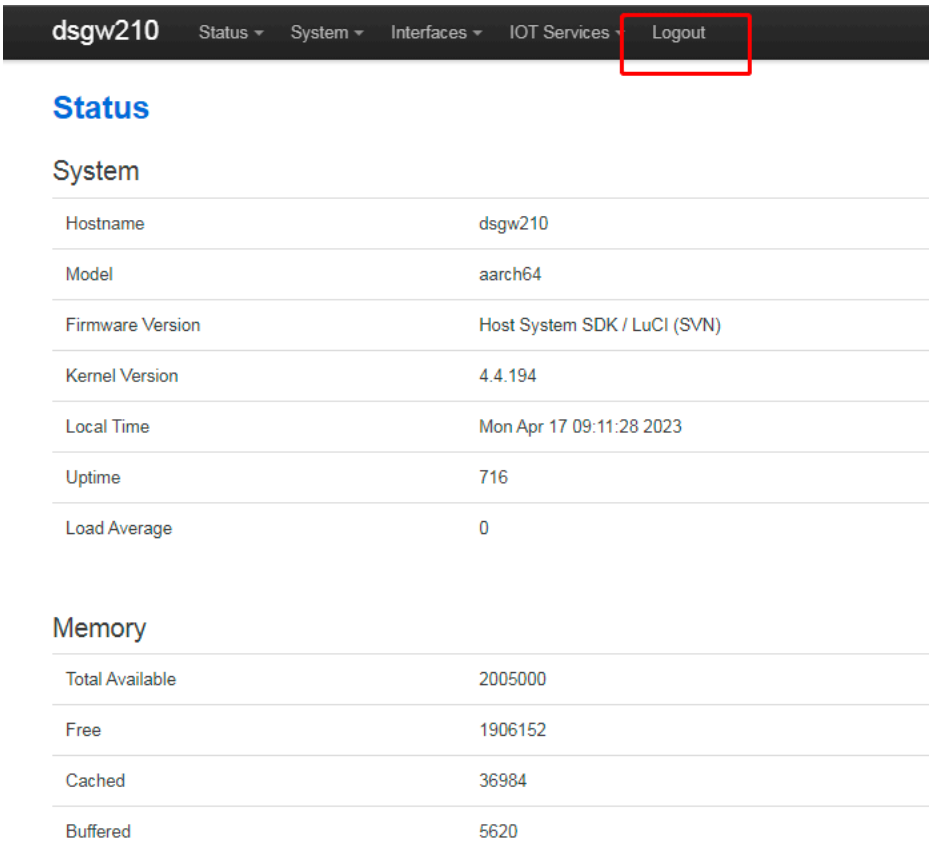
System Upgrade Page



Reboot Page



Logout Page


dsgw210
Status
System
Interfaces
IOT Services
Logout
Status
System
Hostname dsgw210
Model aarch64
Firmware Version Host System SDK / LuCI (SVN)
Kernel Version 4.4.194
Local Time Mon Apr 17 09:11:28 2023
Uptime 716
Load Average 0
Memory
Total Available 2005000
Free 1906152
Cached 36984
Buffered 5620

9. Application Layer Development

9.1. Led

This board has six leds can be controlled by software, they are led0 ~ led5.

9.2. Turn on led with red

```
1. echo 15,0,0 > /sys/class/leds/led0/rgb_value
```

9.3. Turn off led

```
1. echo 0,0,0 > /sys/class/leds/led0/rgb_value
```

9.4. Button

This board has one button that can be used by the software. When pressed, the system will auto call the button script in the /etc/rc.button/battery_power_switch

Here are two incomming parameters:

- SEEN: this is the time, unit seconds
- ACTION: this is the action, it will pressed or release

See, the example has one function in the button script

Long pressed 3 seconds to close the battery power supply:

```
1. #!/bin/sh
2.
3. [ "$ACTION" = "pressed" ] && {
4. logger "battery_power_switch pressed, SEEN:$SEEN"
5. exit 0
6. }
7.
8.
9. [ ! "$ACTION = "released" ] && {
```

```
exit 0
}

logger "battery_power_switch released, SEEN:$SEEN"


[ $SEEN -gt 3 ] && {
logger "battery switch ing"
val=`i2cget -f -y 0 0x6a 0x09`
ret=$?
[ "$ret" = "0" ] && {
if [ "$val" = "0x20" ]; then #
logger "open battery !!"
i2cset -f -y 0 0x6a 0x09 0x44
else
logger "close battery !!"
i2cset -f -y 0 0x6a 0x09 0x20
fi
}
exit 0
}
```

9.5. Ethernet

ThisGatewayy has two Internet ports:

wan port eth0

see the config

```
root@dsgw210:~# cat /etc/network/interfaces
auto lo
iface lo inet loopback

iface wlan1 inet static
address 192.168.66.1
netmask 255.255.255.0

auto wlan0
iface wlan0 inet dhcp
wireless_mode managed
wireless_essid any
wpa-driver wext
wpa-conf /etc/wpa_supplicant.conf
```

```
metric 1

auto eth0
iface eth0 inet dhcp
```

9.6. Wi-Fi

ThisGatewayy have a 2.4g/5.0g radio, it worked at sta mode when startup see the default config:

```
root@dsgw210:~# cat /etc/network/interfaces
auto lo
iface lo inet loopback

iface wlan1 inet static
address 192.168.66.1
netmask 255.255.255.0

auto wlan0
iface wlan0 inet dhcp
wireless_mode managed
wireless_essid any
wpa-driver wext
wpa-conf /etc/wpa_supplicant.conf
metric 1

auto eth0
iface eth0 inet dhcp
```

```
root@dsgw210:~# cat /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1

network={
key_mgmt=NONE
}
```

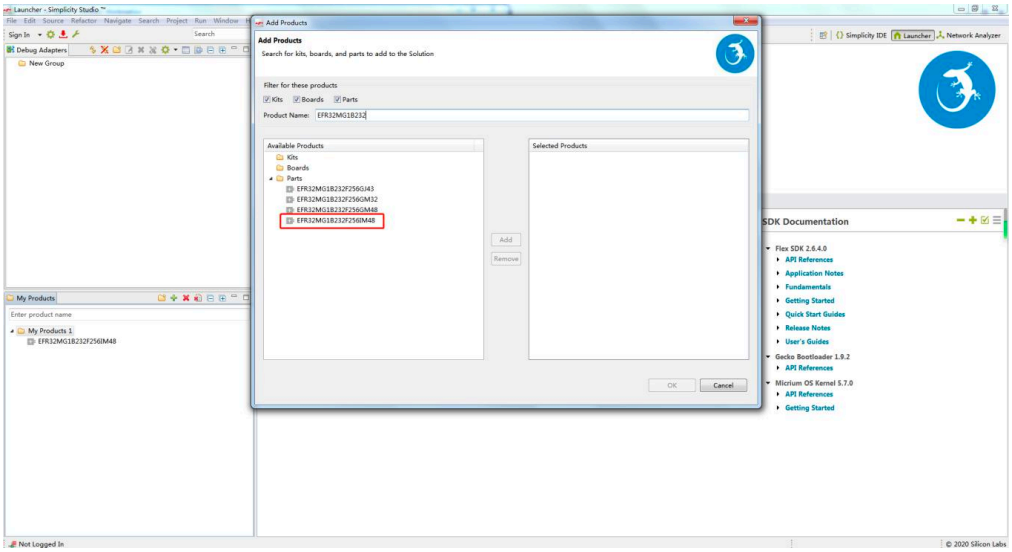
9.7. Zigbee

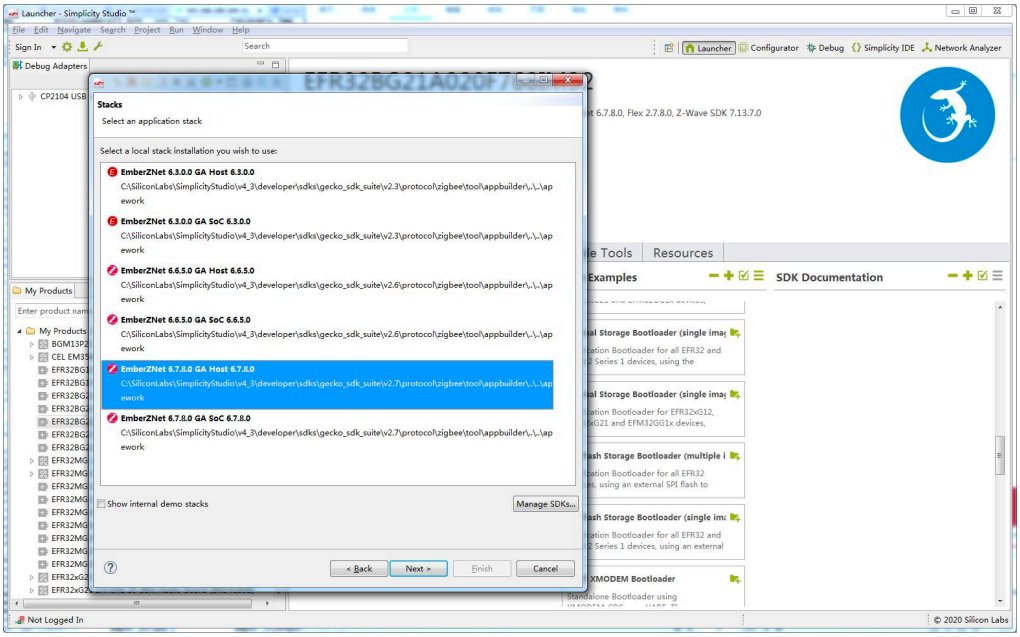
This Gateway supports Zigbee Module EFR32MG1B232 or EFR32MG21

Host Development Demo Example

NCP Development

User can obtained 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 inGatewayy, 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\sdks\gecko_sdk_suite\v2.7\protocol\zigbee\documentation

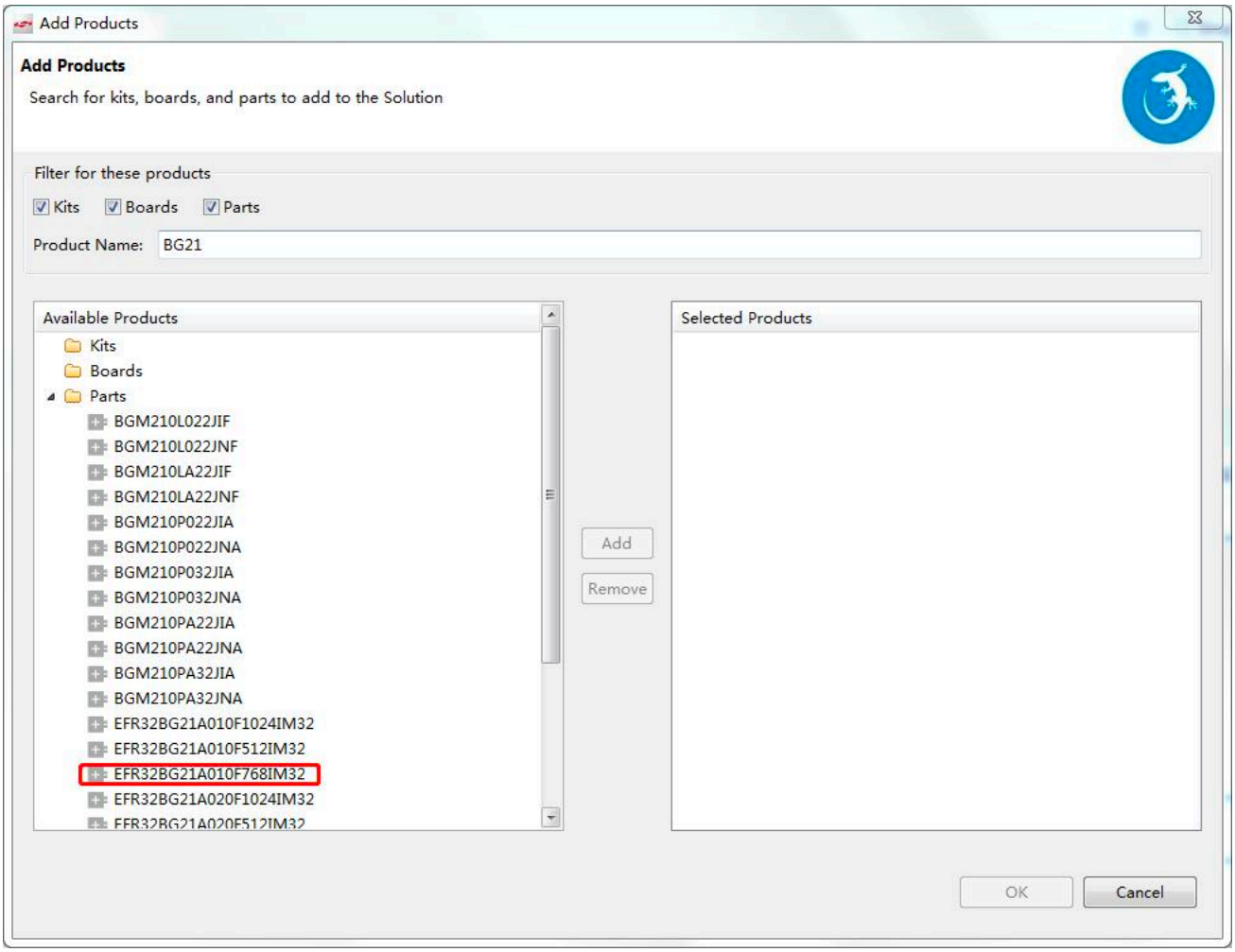
9.8. Bluetooth

ThisGatewayy supports Bluetooth Module EFR32BG21

Host Development Demo Example

NCP Development

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

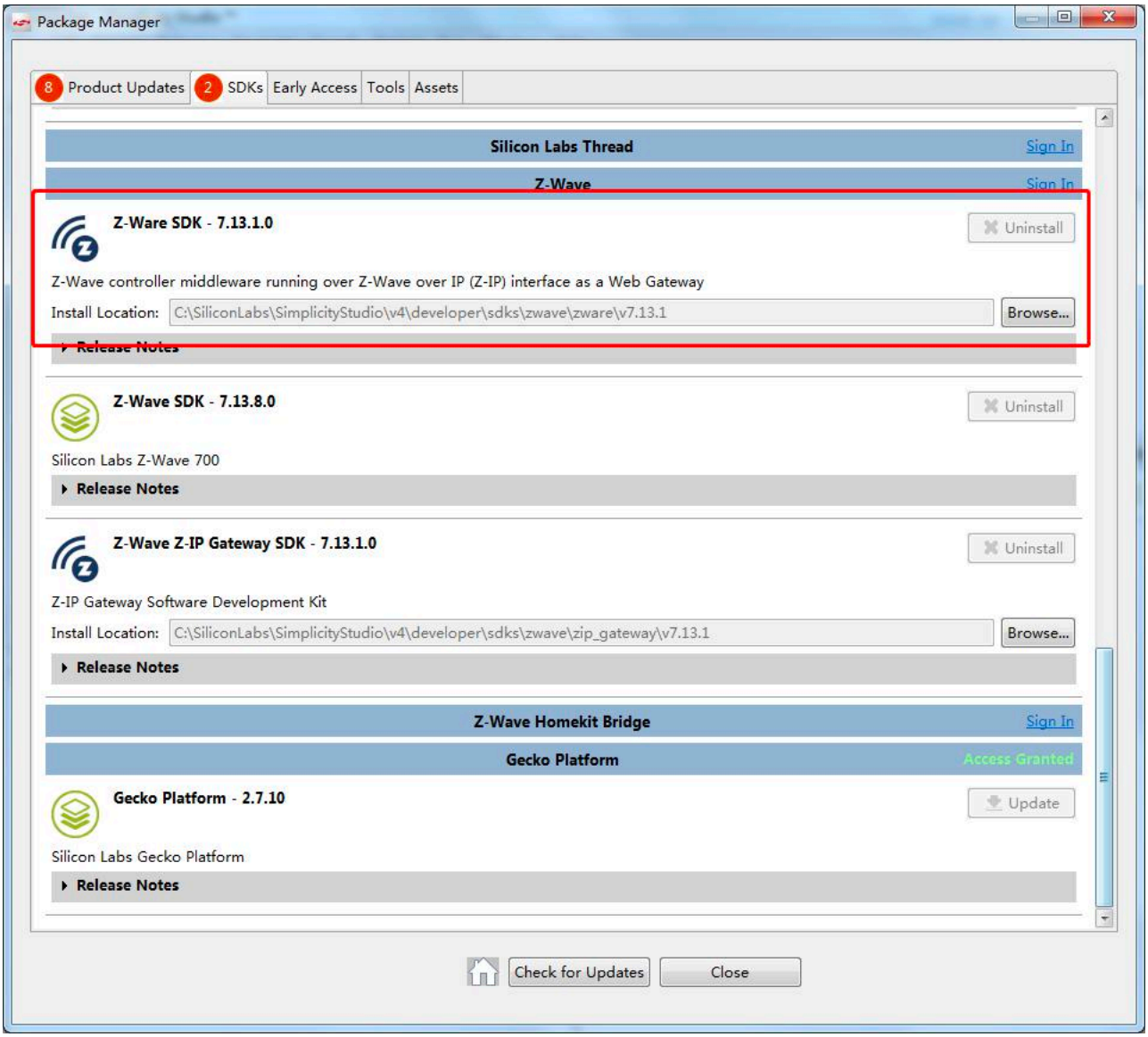


9.9. Z-Wave

ThisGatewayy supports Z-Wave Module ZG130S

Host Development Demo Example

NCP Development



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

9.10. LTE

ThisGatewayy supports LTE Module BG96 or other Module

We use the pppd to dial and manage the bg96.

Here is some config and steps about the lte module

- pppd dial scripts

```
root@dsgw210:~# ls /etc/ppp/peers/* -alh
-rwxr-xr-x 1 root root 366 Apr 7 07:25 /etc/ppp/peers/dial
-rwxr-xr-x 1 root root 163 Apr 7 07:25 /etc/ppp/peers/mo_3gmodule.dial
-rwxr-xr-x 1 root root 1.1K Jan 4 2021 /etc/ppp/peers/provider
```

- Do the pppd call in the foreground

Stop the demo program.

```
/etc/init.d/done stop;/etc/init.d/dial stop;
```

Modify the pppd to debug mode

```
root@dsgw210:~# cat /etc/ppp/options
debug // remove #, open the debug mode
nodetach // add this
#logfile /dev/null // comment this line
noipdefault
```

open the lte module

```
echo 0 > /sys/class/leds/ltepwr/brightness
echo 0 > /sys/class/leds/lterst/brightness
echo 0 > /sys/class/leds/lterf/brightness
sleep 1
echo 1 > /sys/class/leds/ltepwr/brightness
echo 1 > /sys/class/leds/lterst/brightness
echo 1 > /sys/class/leds/lterf/brightness
sleep 15
```

pppd dial

```
root@dsgw210:~# pppd call dial
timeout set to 3 seconds
send (ate0^M)
expect (OK)
^M
OK
-- got it

send (at^M)
expect (OK)
^M
^M
OK
-- got it

send (AT+CSQ^M)
expect (OK)
^M
^M
+CSQ: 99,99^M
^M
OK
-- got it

send (AT+COPS?^M)
expect (OK)
^M
^M
+COPS: 0^M
^M
OK
-- got it

send (AT+CREG?^M)
expect (OK)
^M
^M
+CREG: 0,0^M
^M
OK
```

```
-- got it

send (AT+CEREG?^M)
expect (OK)
^M
^M
+CEREG: 0,0^M
^M
OK
-- got it

send (AT+CPIN?^M)
expect (READY)
```

9.11. LoRaWAN

Stop the run lorawan demo

```
/etc/init.d/done stop; /etc/init.d/lora stop
```

Run the LoRa packet forward

```
root@dsgw210:/usr/bin# cd /usr/bin/lora/
root@dsgw210:/usr/bin/lora# ./lora_pkt_fwd -c ./global_conf.json
*** Packet Forwarder ***
Version: 2.1.0
*** SX1302 HAL library version info ***
Version: 2.1.0;
***
INFO: Little endian host
INFO: found configuration file ./global_conf.json, parsing it
INFO: ./global_conf.json does contain a JSON object named SX130x_conf, parsing SX1302 parameters
INFO: com_type SPI, com_path /dev/spidev32766.0, lorawan_public 1, clksrc 0, full_duplex 0
INFO: antenna_gain 0 dBi
INFO: Configuring legacy timestamp
INFO: no configuration for SX1261
INFO: Configuring Tx Gain LUT for rf_chain 0 with 16 indexes for sx1250
INFO: radio 0 enabled (type SX1250), center frequency 471400000, RSSI offset -207.000000, tx enabled 1, single input mode 1
```

```
INFO: radio 1 enabled (type SX1250), center frequency 475000000, RSSI offset -207.000000, tx enabled 0, single input mode 1
INFO: Lora multi-SF channel 0> radio 0, IF -300000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 1> radio 0, IF -100000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 2> radio 0, IF 100000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 3> radio 0, IF 300000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 4> radio 1, IF -300000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 5> radio 1, IF -100000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 6> radio 1, IF 100000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 7> radio 1, IF 300000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora std channel> radio 1, IF -200000 Hz, 250000 Hz bw, SF 7, Explicit header
INFO: FSK channel> radio 1, IF 300000 Hz, 125000 Hz bw, 50000 bps datarate
INFO: ./global_conf.json does contain a JSON object named gateway_conf, parsing gateway parameters
INFO: gateway MAC address is configured to AA555A0000000000
INFO: server hostname or IP address is configured to "localhost"
INFO: upstream port is configured to "1730"
INFO: downstream port is configured to "1730"
INFO: downstream keep-alive interval is configured to 10 seconds
INFO: statistics display interval is configured to 30 seconds
INFO: upstream PUSH_DATA time-out is configured to 100 ms
INFO: packets received with a valid CRC will be forwarded
INFO: packets received with a CRC error will NOT be forwarded
INFO: packets received with no CRC will NOT be forwarded
INFO: GPS serial port path is configured to "/dev/ttyS0"
INFO: Reference latitude is configured to 0.000000 deg
INFO: Reference longitude is configured to 0.000000 deg
INFO: Reference altitude is configured to 0 meters
INFO: Beaconing period is configured to 0 seconds
INFO: Beaconing signal will be emitted at 869525000 Hz
INFO: Beaconing datarate is set to SF9
INFO: Beaconing modulation bandwidth is set to 125000Hz
INFO: Beaconing TX power is set to 14dBm
INFO: Beaconing information descriptor is set to 0
INFO: ./global_conf.json does contain a JSON object named debug_conf, parsing debug parameters
INFO: got 2 debug reference payload
INFO: reference payload ID 0 is 0xCAFE1234
INFO: reference payload ID 1 is 0xCAFE2345
INFO: setting debug log file name to loragw_hal.log
INFO: [main] TTY port /dev/ttyS0 open for GPS synchronization
```

```
55. Opening SPI communication interface
```

10. Support

Please contact our Sales and FAE colleagues for more information

11. Reference