DUSUN DSGW-010C IoT Edge Computer Gateway User Guide

A DUSUN company
SDK Quick Start Guide
Product Name: IoT Edge Computer Gateway
Model Name: DSGW-010C

Contents

DSGW-010C IoT Edge Computer Gateway

Revision History

Specification		Sect.	Update Description	By
Rev	Date			
1.0	2022-07-07		New version release	

Approvals

Organization	Name	Title	Date

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-010C gateway.
Base on the 4.4 Linux kernel, and leveraging existing open source software, the SDK simplifies the process of adding custom applications. Device drivers, GNU toolchain, Predefined configuration profiles, and sample applications are all in included.

Gateway Information

2.1 Basic information

SOC: PX30 Quad-core ARM Cortex-A53
2GB on-board RAM
32GB eMMC
Base on the LoRa Concentrator Engine: Semtech SX1302
TX power up to 27dBm, RX sensitivity down to -139dBm @SF12, BW125kHz
LoRa Frequency band support: RU864, IN865, EU868, US915, AU915, KR920, AS923.
Support Wi-Fi 2.4G/5G IEEE 802.11b/g/n/ac
Support BLE5.0
Support GPS GLONASS Galileo and QZSS
Support IP66 waterproof housing

2.2 Interface

Target Setup

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

Connecting a gateway – Power

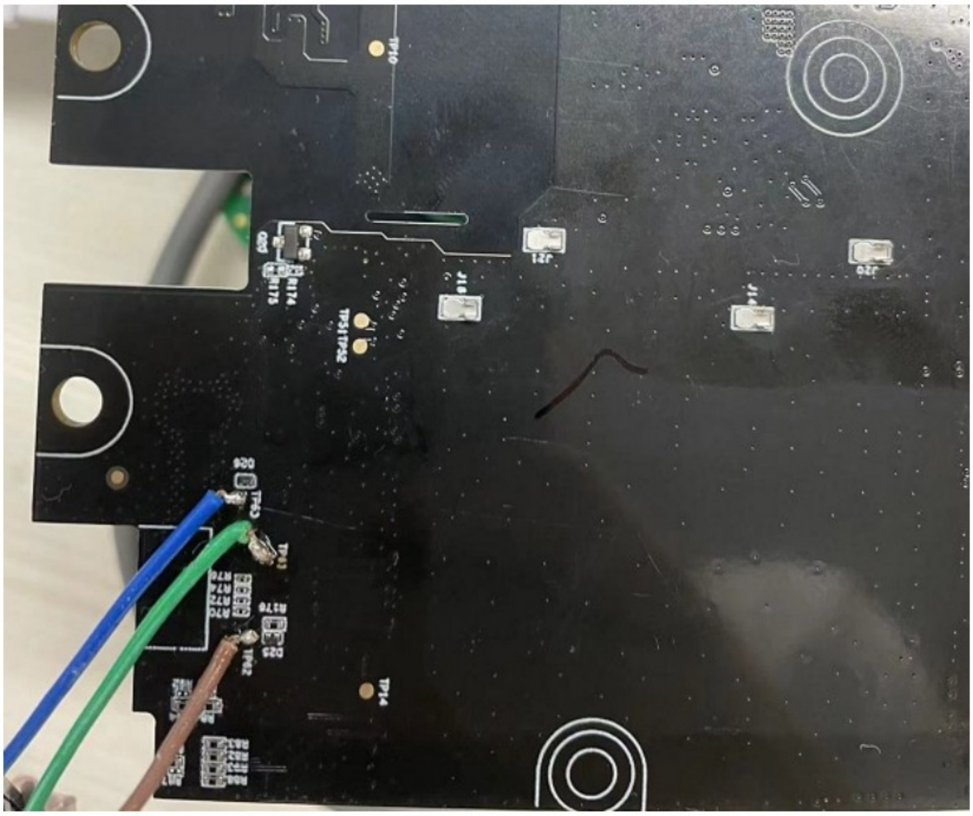
1. Make sure that the power adapter is 5V/3A.
2. Select the appropriate power plug adaptor for your geographical location. Insert it into the slot on the Universal Power Supply; then plug the power supply into an outlet.
3. Connect the output plug of the power supply to the gateway

Connecting a gateway – USB port

1. Connect one end of the USB cable to the USB port on the laptop or desktop
2. Connect the other end of USB cable to the USB port on the gateway.

Connecting a PCBA board – Serial Port

If you want to debug the gateway, you can open the shell, Connect the PC to the PCBA board via Serial to USB tool.
Green: GND
Blue: RX
Brown: TX



Compile the Environment to Build

Please use ubuntu 18.04 .iso image to setup your build environment. You can use a virtual machine or a physical PC to install ubuntu 18.04.

4.1 Virtual Machine

It is recommended that novice users use virtual machines, install ubuntu 18.04 to the virtual machine, and leave enough disk space (at least 100G) for the virtual machine.

4.2 Ubuntu PC Compile the Environment to Build

The use of physical machine compilation users can use a ubuntu PC.

SDK Acquisition and Preparation

5.1 Download the source code from the Dusun FTP

The source package name will be px30_sdk.tar.gz, get it from Dusun FTP.

5.2 Code Compression Package Check

The next step can be taken only after generating the MD5 value of the source compression package and comparing the MD5 value of the MD5 .txt text to confirm that the MD5 value is the same, and if the MD5 value is not the same, the energy code pack is damaged, please download it again.

```
$ md5sum px30_sdk.tar.gz
```

5.3 The Source Compression Package is Unzipped

Copy the source code to the corresponding directory and unzip the source code compression package.

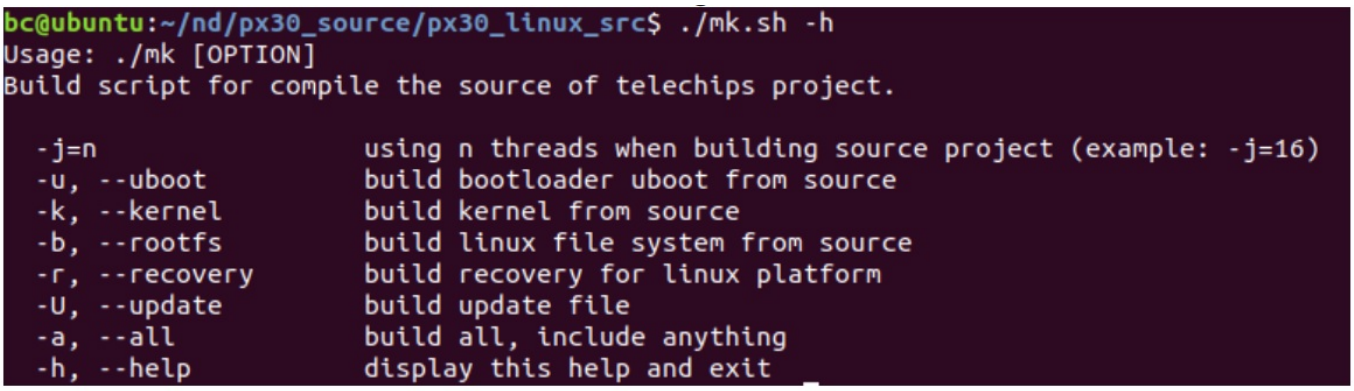
```
$ sudo -i
$ mkdir workdir
$ cd workdir
$ tar -zxvf  /path/to/ px30_sdk.tar.gz
$ cd px30_linux_src
$ git reset --hard
```

Code Compilation

6.1 Getting started, global Compilation

6.1.1 Initialize Compilation Environment Variables (select file system)

You can build buildroot, ubuntu or debian rootfs image. Select it in “./mk.sh”.



```
bc@ubuntu:~/nd/px30_source/px30_linux_src$ ./mk.sh -h
Usage: ./mk [OPTION]
Build script for compile the source of telechips project.

  -j=n                    using n threads when building source project (example: -j=16)
  -u, --uboot             build bootloader uboot from source
  -k, --kernel            build kernel from source
  -b, --rootfs            build linux file system from source
  -r, --recovery          build recovery for linux platform
  -U, --update            build update file
  -a, --all               build all, include anything
  -h, --help              display this help and exit
```

6.1.2 Prepare the Root File System base

This section is for building ubuntu or debian file system.

Compile Ubuntu

Download the root file system image rootfs-ubuntu16_xubuntu_v1.1.img Copy the root file system to the specified path, then run command ./mk.sh

```
$ mkdir output
$ mv ../rootfs-ubuntu16_xubuntu_v1.1.img ./output
```

The build will take a long time, please wait patiently.
Then the image would be placed in ./output/update-ubuntu.img
The update-ubuntu.img can be used to update firmware in gateway

Compile buildroot

Compile the buildroot image by command mk.sh -b

```
$ ./mk.sh -b
```

The build will take a long time, please wait patiently.
Then the image would be placed in ./output/update. img
The update. img can be used to update firmware in gateway

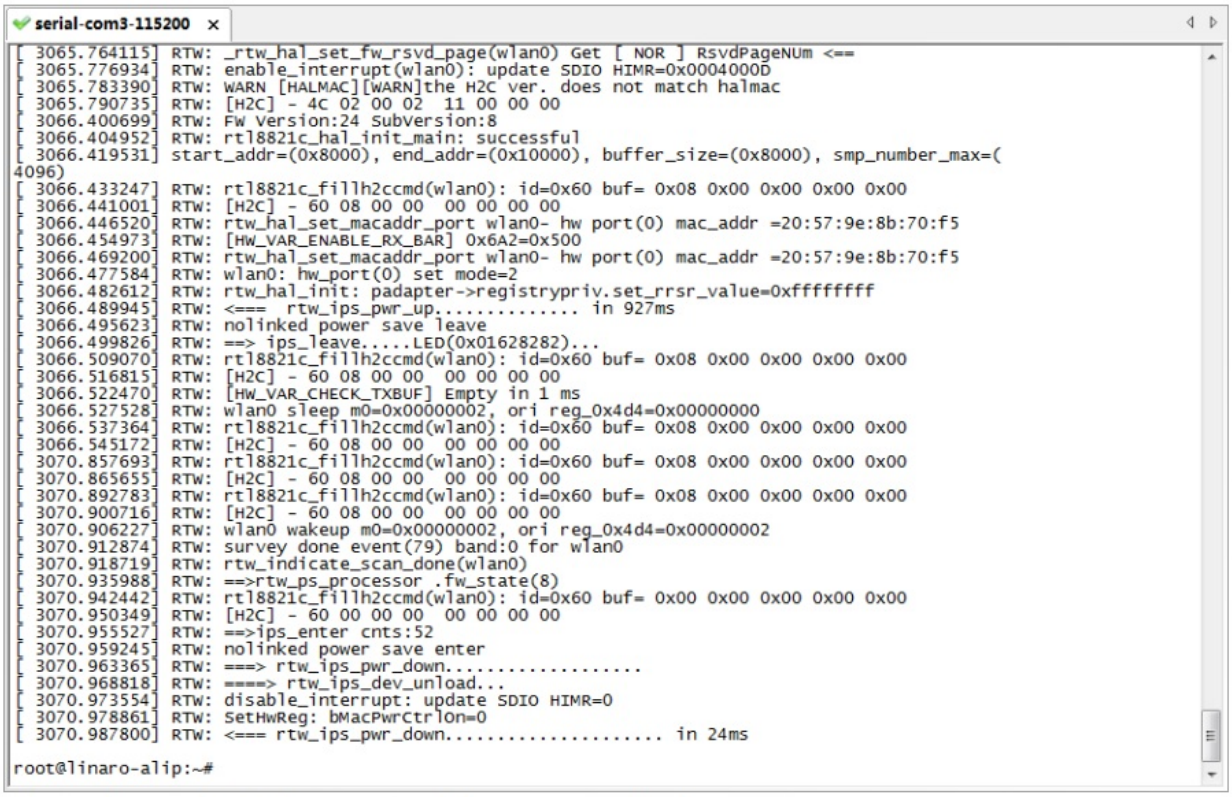
6.1.3 Run The Image on the board

Connect the PX30 board serial port to the PC via a USB to UART Bridge.

Use Putty or other Terminal software as your console tool,
SERIAL CONSOLE SETTINGS:

- 115200/8N1
- Baud: 115200
- Data Bits: 8
- Parity Bit: No
- Stop Bit: 1

Power UP the board, you can see the boot log on console:



```
[ 3065.764115] RTW: _rtw_hal_set_fw_rsvd_page(wlan0) Get [ NOR ] RsvdPageNUm <==
[ 3065.776934] RTW: enable_interrupt(wlan0): update SDIO HIMR=0x0004000D
[ 3065.783390] RTW: WARN [HALMAC][WARN]the H2C ver. does not match halmac
[ 3065.790735] RTW: [H2C] - 4C 02 00 02  11 00 00 00
[ 3066.400699] RTW: FW Version:24 SubVersion:8
[ 3066.404952] RTW: rtl8821c_hal_init_main: successful
[ 3066.419531] start_addr=(0x8000), end_addr=(0x10000), buffer_size=(0x8000), smp_number_max=(
4096)
[ 3066.433247] RTW: rtl8821c_fillh2ccmd(wlan0): id=0x60 buf= 0x08 0x00 0x00 0x00 0x00
[ 3066.441001] RTW: [H2C] - 60 08 00 00  00 00 00 00
[ 3066.446520] RTW: rtw_hal_set_macaddr_port wlan0- hw port(0) mac_addr =20:57:9e:8b:70:f5
[ 3066.454973] RTW: [HW_VAR_ENABLE_RX_BAR] 0x6A2=0x500
[ 3066.469200] RTW: rtw_hal_set_macaddr_port wlan0- hw port(0) mac_addr =20:57:9e:8b:70:f5
[ 3066.477584] RTW: wlan0: hw_port(0) set mode=2
[ 3066.482612] RTW: rtw_hal_init: padapter->registrypriv.set_rrsr_value=0xffffffff
[ 3066.489945] RTW: <===  rtw_ips_pwr_up.............. in 927ms
[ 3066.495623] RTW: nolinked power save leave
[ 3066.499826] RTW: ==> ips_leave.....LED(0x01628282)...
[ 3066.509070] RTW: rtl8821c_fillh2ccmd(wlan0): id=0x60 buf= 0x08 0x00 0x00 0x00 0x00
[ 3066.516815] RTW: [H2C] - 60 08 00 00  00 00 00 00
[ 3066.522470] RTW: [HW_VAR_CHECK_TXBUF] Empty in 1 ms
[ 3066.527528] RTW: wlan0 sleep m0=0x00000002, ori reg_0x4d4=0x00000000
[ 3066.537364] RTW: rtl8821c_fillh2ccmd(wlan0): id=0x60 buf= 0x08 0x00 0x00 0x00 0x00
[ 3066.545172] RTW: [H2C] - 60 08 00 00  00 00 00 00
[ 3070.857693] RTW: rtl8821c_fillh2ccmd(wlan0): id=0x60 buf= 0x08 0x00 0x00 0x00 0x00
[ 3070.865655] RTW: [H2C] - 60 08 00 00  00 00 00 00
[ 3070.892783] RTW: rtl8821c_fillh2ccmd(wlan0): id=0x60 buf= 0x08 0x00 0x00 0x00 0x00
[ 3070.900716] RTW: [H2C] - 60 08 00 00  00 00 00 00
[ 3070.906227] RTW: wlan0 wakeup m0=0x00000002, ori reg_0x4d4=0x00000002
[ 3070.912874] RTW: survey done event(79) band:0 for wlan0
[ 3070.918719] RTW: rtw_indicate_scan_done(wlan0)
[ 3070.935988] RTW: ==>rtw_ps_processor .fw_state(8)
[ 3070.942442] RTW: rtl8821c_fillh2ccmd(wlan0): id=0x60 buf= 0x00 0x00 0x00 0x00 0x00
[ 3070.950349] RTW: [H2C] - 60 00 00 00  00 00 00 00
[ 3070.955527] RTW: ==>ips_enter cnts:52
[ 3070.959245] RTW: nolinked power save enter
[ 3070.963365] RTW: ===> rtw_ips_pwr_down...................
[ 3070.968818] RTW: ====> rtw_ips_dev_unload...
[ 3070.973554] RTW: disable_interrupt: update SDIO HIMR=0
[ 3070.978861] RTW: SetHwReg: bMacPwrCtrlOn=0
[ 3070.987800] RTW: <=== rtw_ips_pwr_down.................... in 24ms

root@linaro-alip:~#
```

There is no default password for system login.

6.2 Compiled Each Image Part Separately

6.2.1 The build system and the image structure

The update.img is composed of several parts. Main parts are uboot. img, boot.img, recovery.img, rootfs.img.
uboot.img contains bootloader uboot boot.img contains the device tree .dtb image, Linux kernel image
recovery.img: The system can boot up to recovery mode, recovery.img is the rootfs used in recovery mode.
rootfs.img: The normal rootfs image. In normal mode, system boot and mount this rootfs image.
You may need to build the images separately, especially when you focus on single module (e.g. uboot or kernel driver) development. Then you can build only that part of image and update that partition in flash.

6.2.2 Build Uboot only

```
$ ./mk.sh -u
```

6.2.3 Build Linux Kernel Only

```
$ ./mk.sh -k
```

6.2.4 Build Recovery File System Only

```
$ ./mk.sh -r
```

More about buildroot system

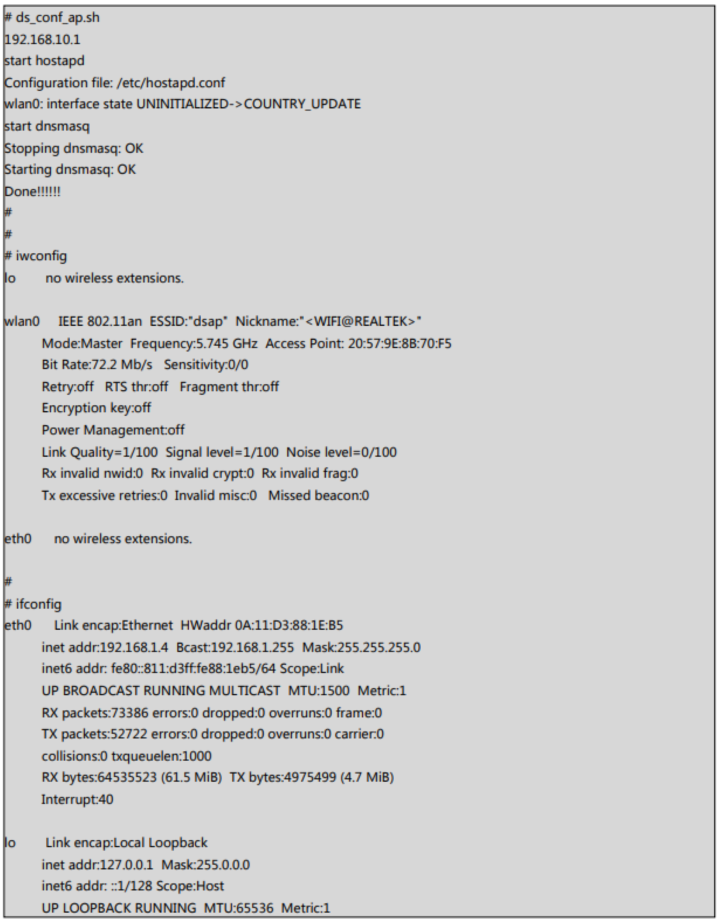
If you use buildroot rootfs, some Dusun test scripts/tools are already installed in the final buildroot rootfs. You can refer to buildroot/dusun_rootfs/add_ds_rootfs.sh

7.1 Test hardware components

The following testing are done under the buildroot system.

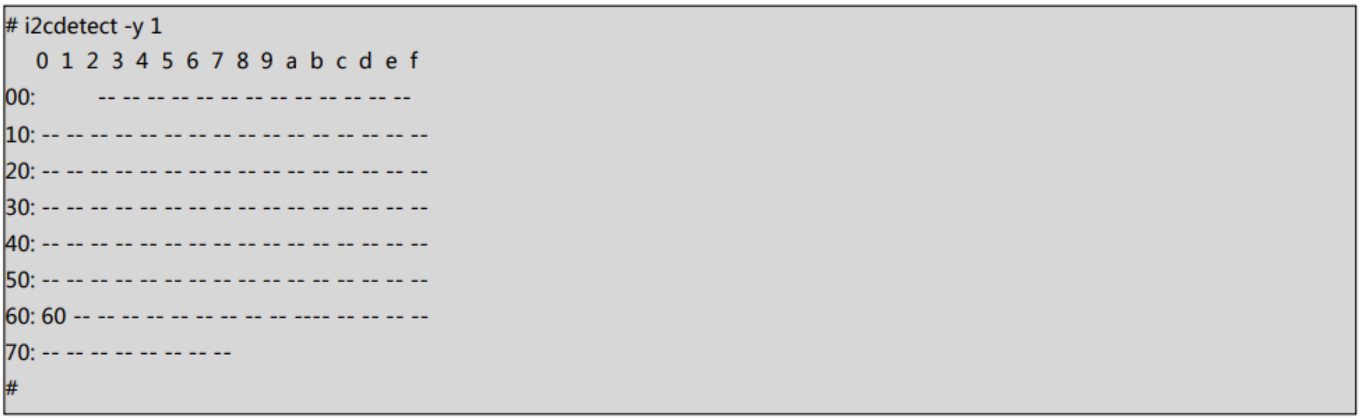
7.1.1 Test Wi-Fi as AP

The “ds_conf_ap.sh” script is for setting up Wi-Fi AP, SSID is “dsap”, password is “12345678”.



```
# ds_conf_ap.sh
192.168.10.1
start hostapd
Configuration file: /etc/hostapd.conf
wlan0: interface state UNINITIALIZED->COUNTRY_UPDATE
start dnsmasq
Stopping dnsmasq: OK
Starting dnsmasq: OK
Done!!!!!!
#
#
# iwconfig
lo        no wireless extensions.

wlan0     IEEE 802.11an  ESSID:"dsap"  Nickname:"<WIFI@REALTEK>"
          Mode:Master  Frequency:5.745 GHz  Access Point: 20:57:9E:8B:70:F5
          Bit Rate:72.2 Mb/s   Sensitivity:0/0
          Retry:off   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=1/100  Signal level=1/100  Noise level=0/100
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

eth0      no wireless extensions.

#
# ifconfig
eth0      Link encap:Ethernet  HWaddr 0A:11:D3:88:1E:B5
          inet addr:192.168.1.4  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::811:d3ff:fe88:1eb5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:73386 errors:0 dropped:0 overruns:0 frame:0
          TX packets:52722 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:64535523 (61.5 MiB)  TX bytes:4975499 (4.7 MiB)
          Interrupt:40

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
```

7.1.2 Test I2C

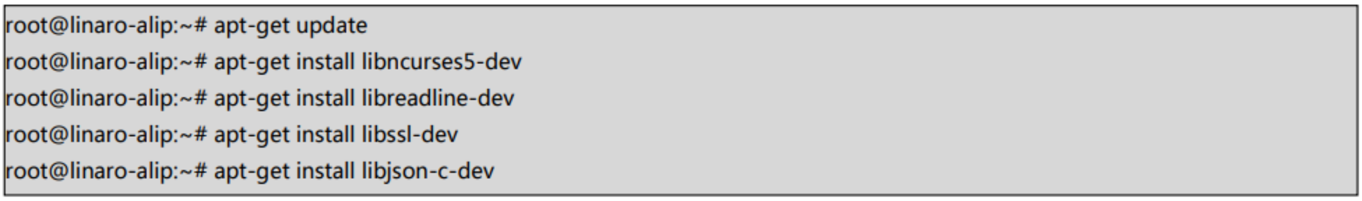


```
# i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: 60 -- -- -- -- -- -- -- -- -- ---- -- -- -- --
70: -- -- -- -- -- -- -- --
#
```

Test of i2c function in gateway

Wireless development (Zigbee, Z-Wave, BLE, LoRaWAN)

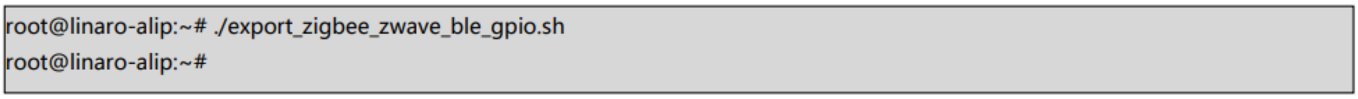
Please use the ubuntu system to do the following steps. The code will be compiled on the board, not on host.



```
root@linaro-alip:~# apt-get update
root@linaro-alip:~# apt-get install libncurses5-dev
root@linaro-alip:~# apt-get install libreadline-dev
root@linaro-alip:~# apt-get install libssl-dev
root@linaro-alip:~# apt-get install libjson-c-dev
```

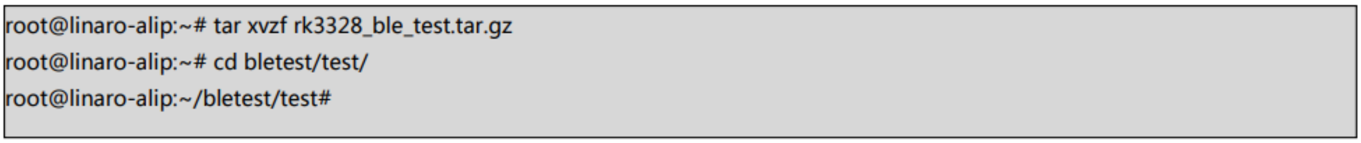
1. Prepare some library on the board
2. scp SDK

8.1 BLE



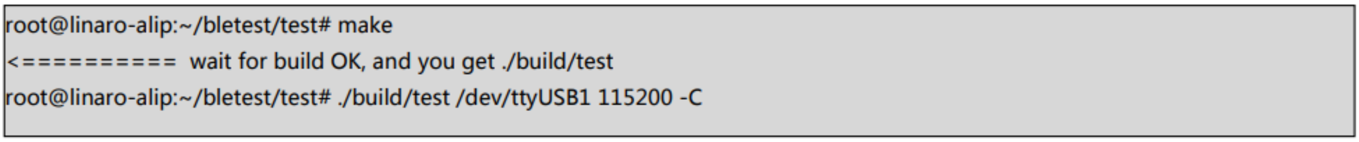
```
root@linaro-alip:~# ./export_zigbee_zwave_ble_gpio.sh
root@linaro-alip:~#
```

BLE interface is /dev/ttyUSB1.
Download “rk3328_ble_test.tar.gz” from Dusun FTP, and copy it to board, under /root.



```
root@linaro-alip:~# tar xvzf rk3328_ble_test.tar.gz
root@linaro-alip:~# cd bletest/test/
root@linaro-alip:~/bletest/test#
```

Unzip it and you can get ./bletest build ble test tool and run:
More information about the BLE test tool, please visit https://docs.silabs.com/ for more information.



```
root@linaro-alip:~/bletest/test# make
<==========  wait for build OK, and you get ./build/test
root@linaro-alip:~/bletest/test# ./build/test /dev/ttyUSB1 115200 -C
```

8.2 LoRaWAN

Choose the correct interface for LoRaWAN, for example /dev/spidev32766.0.

The configuration file for it is in ./sx1302_hal/packet_forwarder/global_conf.json.
Download “sx1302_hal_0210.tar.gz” from Dusun FTP, and copy it to board, under /root.

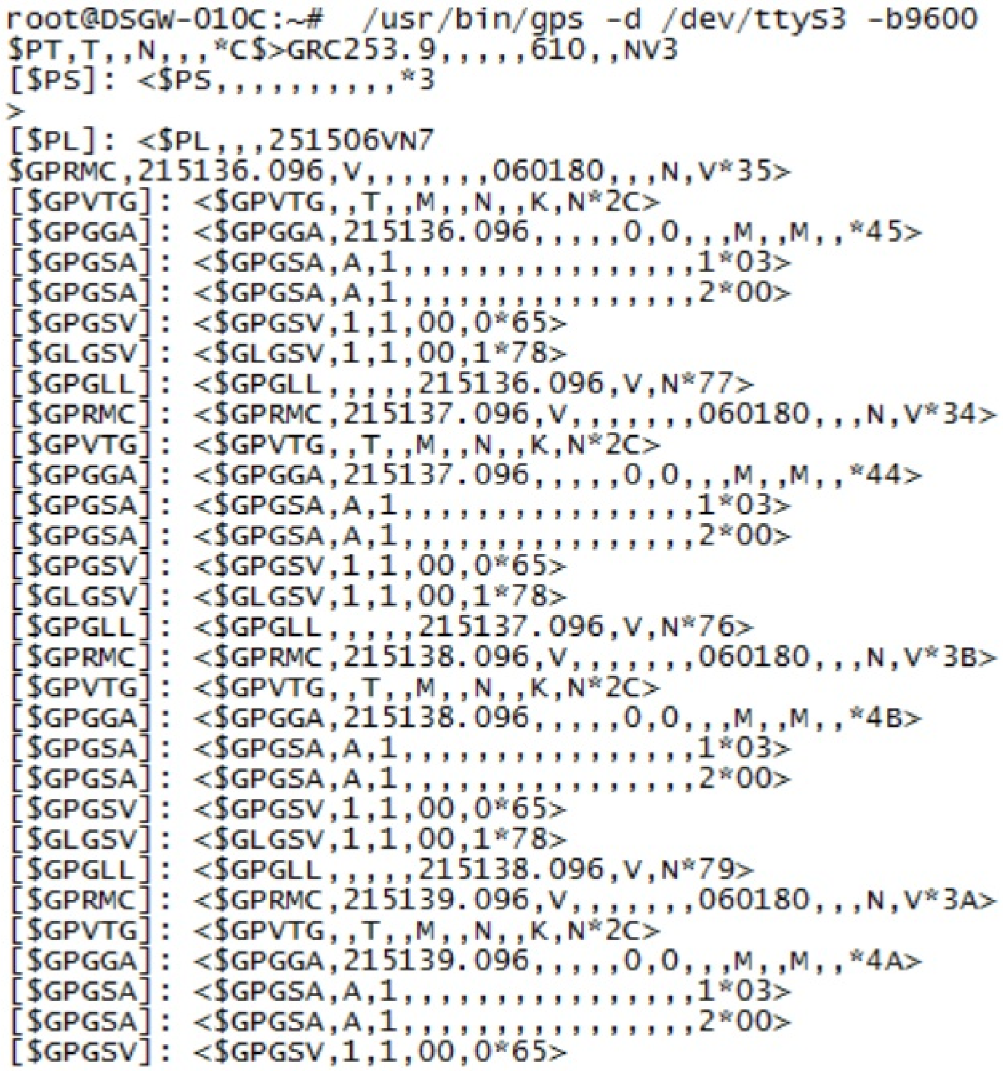
```
root@linaro-alip:~# tar xvzf sx1302_hal_0210.tar.gz
root@linaro-alip:~# cd sx1302_hal/
```

Untar it and you can get ./sx1302_hal build LoRaWAN sample code sx1302_hal and run:
More information about the LoRaWAN code, please visit https://www.semtech.com/products/wireless-rf/lora-core/sx1302 for more information.

```
root@linaro-alip:~/sx1302_hal# make all
<==========  wait for build OK, and you get ./packet_forwarder/
root@linaro-alip:~/sx1302_hal/packet_forwarder# ./lora_pkt_fwd
```

8.3 GPS

Acquire the GPS data from gps program, the default serial port is ttyS3, baud rate 9600



```
root@DSGW-010C:~#  /usr/bin/gps -d /dev/ttyS3 -b9600
$PT,T,,N,,,*C$>GRC253.9,,,,,610,,NV3
[$PS]: <$PS,,,,,,,,,,,*3
>
[$PL]: <$PL,,,251506VN7
$GPRMC,215136.096,V,,,,,,,060180,,,N,V*35>
[$GPVTG]: <$GPVTG,,T,,M,,N,,K,N*2C>
[$GPGGA]: <$GPGGA,215136.096,,,,,0,0,,,M,,M,,*45>
[$GPGSA]: <$GPGSA,A,1,,,,,,,,,,,,,,,,1*03>
[$GPGSA]: <$GPGSA,A,1,,,,,,,,,,,,,,,,2*00>
[$GPGSV]: <$GPGSV,1,1,00,0*65>
[$GLGSV]: <$GLGSV,1,1,00,1*78>
[$GPGLL]: <$GPGLL,,,,,215136.096,V,N*77>
[$GPRMC]: <$GPRMC,215137.096,V,,,,,,,060180,,,N,V*34>
[$GPVTG]: <$GPVTG,,T,,M,,N,,K,N*2C>
[$GPGGA]: <$GPGGA,215137.096,,,,,0,0,,,M,,M,,*44>
[$GPGSA]: <$GPGSA,A,1,,,,,,,,,,,,,,,,1*03>
[$GPGSA]: <$GPGSA,A,1,,,,,,,,,,,,,,,,2*00>
[$GPGSV]: <$GPGSV,1,1,00,0*65>
[$GLGSV]: <$GLGSV,1,1,00,1*78>
[$GPGLL]: <$GPGLL,,,,,215137.096,V,N*76>
[$GPRMC]: <$GPRMC,215138.096,V,,,,,,,060180,,,N,V*3B>
[$GPVTG]: <$GPVTG,,T,,M,,N,,K,N*2C>
[$GPGGA]: <$GPGGA,215138.096,,,,,0,0,,,M,,M,,*4B>
[$GPGSA]: <$GPGSA,A,1,,,,,,,,,,,,,,,,1*03>
[$GPGSA]: <$GPGSA,A,1,,,,,,,,,,,,,,,,2*00>
[$GPGSV]: <$GPGSV,1,1,00,0*65>
[$GLGSV]: <$GLGSV,1,1,00,1*78>
[$GPGLL]: <$GPGLL,,,,,215138.096,V,N*79>
[$GPRMC]: <$GPRMC,215139.096,V,,,,,,,060180,,,N,V*3A>
[$GPVTG]: <$GPVTG,,T,,M,,N,,K,N*2C>
[$GPGGA]: <$GPGGA,215139.096,,,,,0,0,,,M,,M,,*4A>
[$GPGSA]: <$GPGSA,A,1,,,,,,,,,,,,,,,,1*03>
[$GPGSA]: <$GPGSA,A,1,,,,,,,,,,,,,,,,2*00>
[$GPGSV]: <$GPGSV,1,1,00,0*65>
```

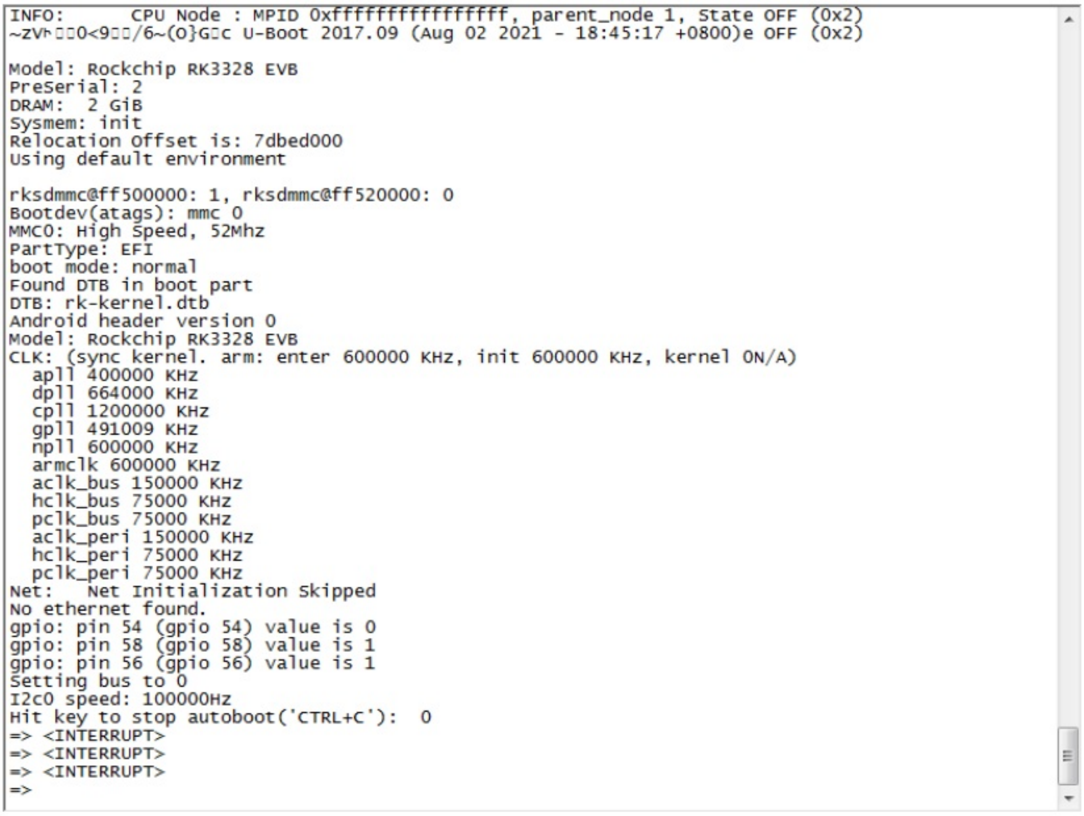
Image Upgrade

9.1 Upgrade Tool

Upgrade tool AndroidTool_Release_v2.69

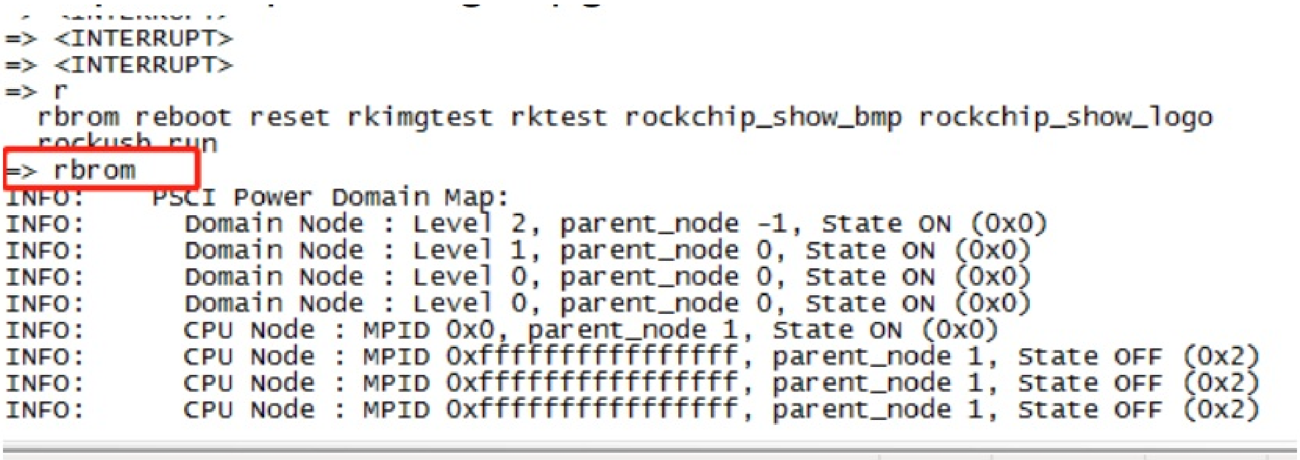
9.2 Go into Upgrade Mode

1. Connect the OTG port to the burning computer USB port, it’s also act as 5V power supply
2. Press “Ctrl+C” when uboot is booting up, to enter uboot:



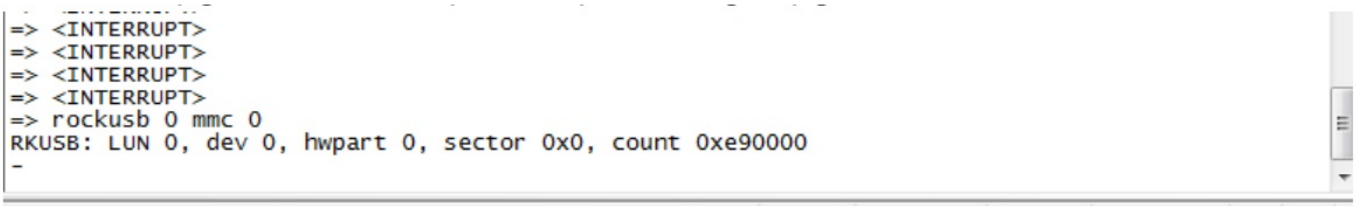
```
INFO:      CPU Node : MPID 0xffffffffffffffff, parent_node 1, State OFF (0x2)
~zVh□□0<9□□/6~(O}G□c U-Boot 2017.09 (Aug 02 2021 - 18:45:17 +0800)e OFF (0x2)

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>
=>
```

3. uboot “rbrom” comand to reboot the board into maskrom mode, for a complete “update.img” upgrade.



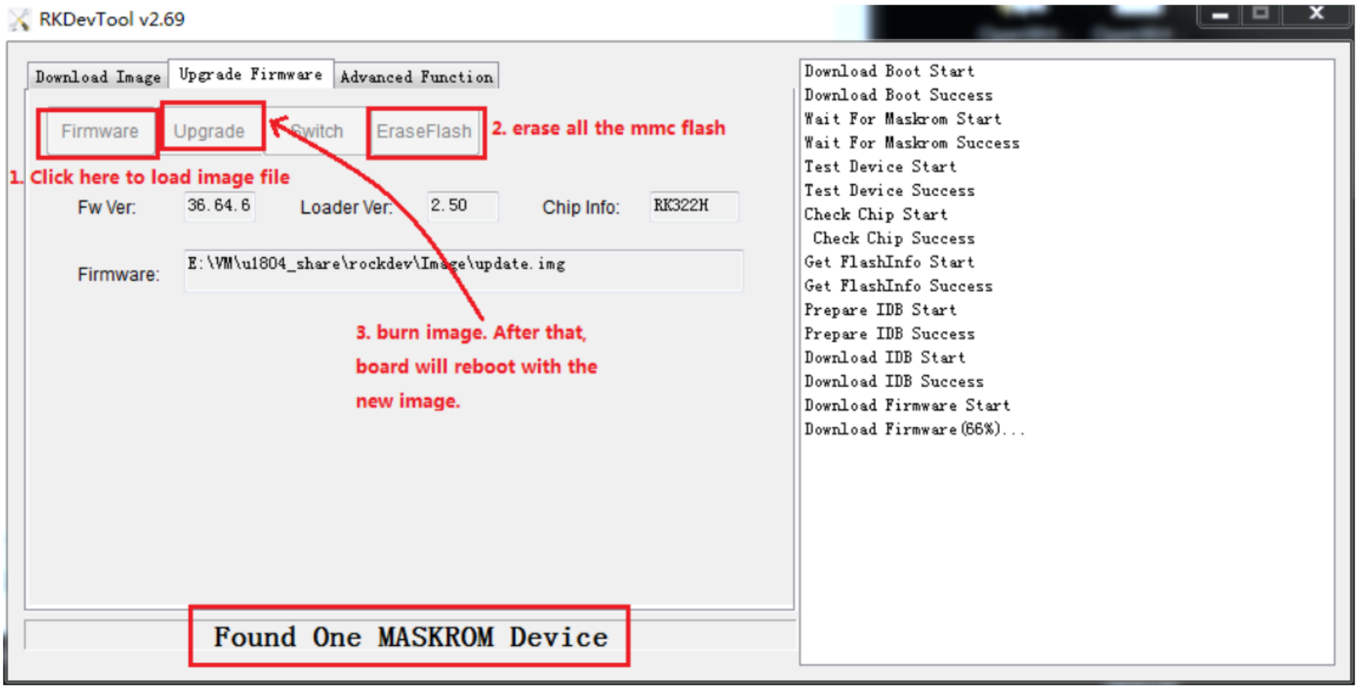
```
=> <INTERRUPT>
=> <INTERRUPT>
=> r
  rbrom reboot reset rkimgtest rktest rockchip_show_bmp rockchip_show_logo
  rockusb run
=> rbrom
INFO:    PSCI Power Domain Map:
INFO:      Domain Node : Level 2, parent_node -1, State ON (0x0)
INFO:      Domain Node : Level 1, parent_node 0, State ON (0x0)
INFO:      Domain Node : Level 0, parent_node 0, State ON (0x0)
INFO:      Domain Node : Level 0, parent_node 0, State ON (0x0)
INFO:      CPU Node : MPID 0x0, parent_node 1, State ON (0x0)
INFO:      CPU Node : MPID 0xffffffffffffffff, parent_node 1, State OFF (0x2)
INFO:      CPU Node : MPID 0xffffffffffffffff, parent_node 1, State OFF (0x2)
INFO:      CPU Node : MPID 0xffffffffffffffff, parent_node 1, State OFF (0x2)
```

4. “rockusb 0 mmc 0” command to reboot board to loader mode, for a partial firmware upgrade or a complete “update. img” upgrade.



```
=> <INTERRUPT>
=> <INTERRUPT>
=> <INTERRUPT>
=> <INTERRUPT>
=> rockusb 0 mmc 0
RKUSB: LUN 0, dev 0, hwpart 0, sector 0x0, count 0xe90000
-
```

9.3 The Entire Package of Firmware “update.img” Upgrade



2. erase all the mmc flash

1. Click here to load image file

Fw Ver: 36.64.6 Loader Ver: 2.50 Chip Info: RK322H

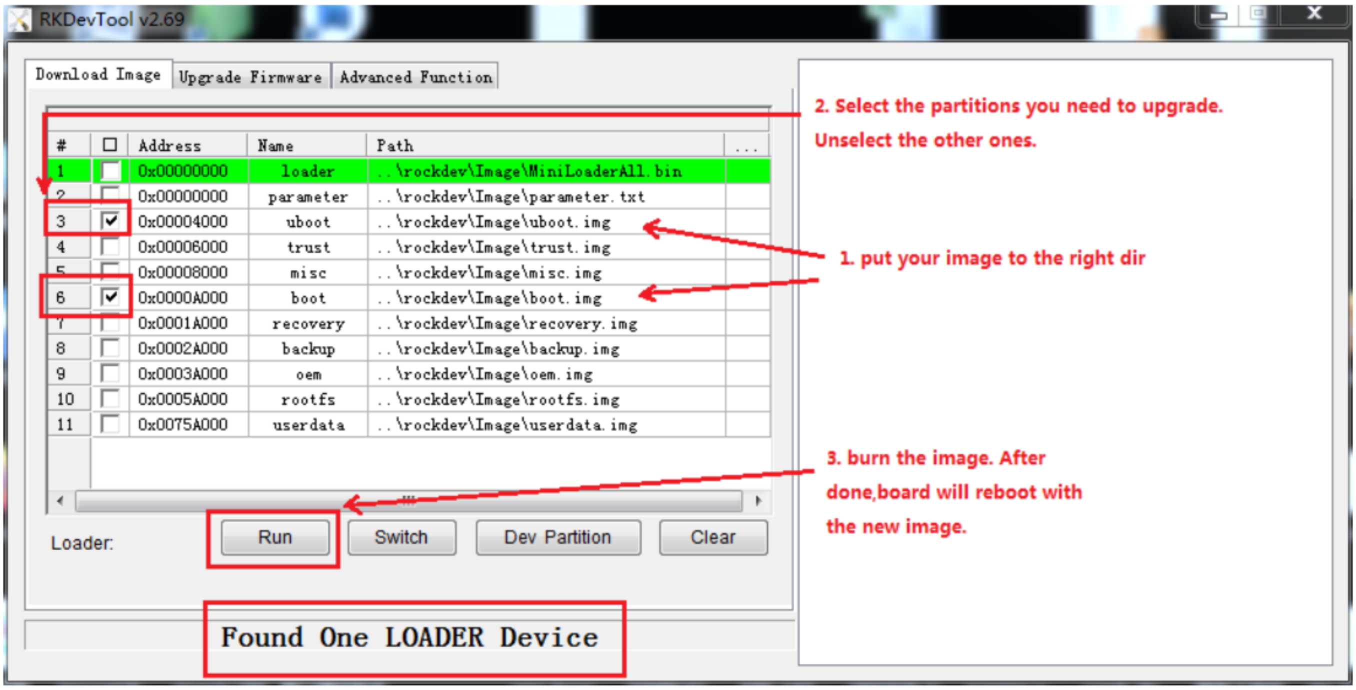
Firmware: E:\VM\u1804_share\rockdev\Image\update.img

3. burn image. After that, board will reboot with the new image.

Found One MASKROM Device

```
Download Boot Start
Download Boot Success
Wait For Maskrom Start
Wait For Maskrom Success
Test Device Start
Test Device Success
Check Chip Start
 Check Chip Success
Get FlashInfo Start
Get FlashInfo Success
Prepare IDB Start
Prepare IDB Success
Download IDB Start
Download IDB Success
Download Firmware Start
Download Firmware(66%)...
```

9.4 Upgrade the Firmware Separately



#	□	Address	Name	Path	...
1	□	0x00000000	loader	..\rockdev\Image\MiniLoaderAll.bin	
2	□	0x00000000	parameter	..\rockdev\Image\parameter.txt	
3	☑	0x00004000	uboot	..\rockdev\Image\uboot.img	
4	□	0x00006000	trust	..\rockdev\Image\trust.img	
5	□	0x00008000	misc	..\rockdev\Image\misc.img	
6	☑	0x0000A000	boot	..\rockdev\Image\boot.img	
7	□	0x0001A000	recovery	..\rockdev\Image\recovery.img	
8	□	0x0002A000	backup	..\rockdev\Image\backup.img	
9	□	0x0003A000	oem	..\rockdev\Image\oem.img	
10	□	0x0005A000	rootfs	..\rockdev\Image\rootfs.img	
11	□	0x0075A000	userdata	..\rockdev\Image\userdata.img	

2. Select the partitions you need to upgrade. Unselect the other ones.

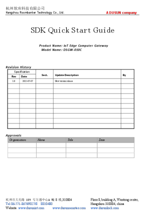
1. put your image to the right dir

3. burn the image. After done,board will reboot with the new image.

Found One LOADER Device

Tel:86-571-86769027/8 8810480
Website: www.dusuniot.com
www.dusunremotes.com
Floor 8, building A, Wantong center,
Hangzhou 310004, china
www.dusunlock.com

Documents / Resources



杭州软库科技有限公司
Hangzhou Roombanker Technology Co., Ltd. A DUSUN company

SDK Quick Start Guide

Product Name: IoT Edge Computer Gateway
Model Name: DSGW-010C

Revision History

Specification		Sect.	Update Description	By
Rev	Date			
1.0	2022-07-07		New version release	

Approvals

Organization	Name	Title	Date

杭州市大关路 189 号万通中心A 幢 8 楼,310004
Tel:86-571-86769027/8 8810480
Website: www.dusuniot.com www.dusunremotes.com www.dusunlock.com

Floor 8, building A, Wantong center,
Hangzhou 310004, china

DUSUN DSGW-010C IoT Edge Computer Gateway [pdf] User Guide
DSGW-010C, DSGW-010C IoT Edge Computer Gateway, IoT Edge Computer Gateway, Edge Computer Gateway, Computer Gateway, Gateway

References

- **Dusun IoT: Embedded Hardware Vendor | IoT Gateway Sepcialist**
- **Dusunremotes | Custom Intelligent Remote Control Manufacturer**
- **Software Developer Docs - Silicon Labs**
- **SX1302 LoRa Core Digital Baseband Chip | Semtech**