robustel | Developer Guide

RobustOS Pro SDK Developer Guide

robustOS Pro

Guangzhou Robustel Co., Ltd.
www.robustel.com

Contents

Chapter 1 Document Revision History 5
Chapter 2 Overview 6
Chapter 3 Basic Information 7
3.1 Supported Hardware 7
3.2 Hardware Information 7
3.3 Hardware Resources 8
3.3.1 EG5100 8
3.3.2 LG5100 8
3.3.3 EG5120 8
3.3.4 R1520LG 8
3.3.5 EV8100 9
3.3.6 EG5101 9
3.3.7 EG5200 9
3.3.8 MG460 9
3.4 Recommended Development Environment 10
3.5 Recommended IDE Tools 10
3.6 Supported Programming Languages 10
3.7 C Library 10
Chapter 4 Preparation for software development 11
4.1 Add Sudo User 11
4.2 Generate RSA key 11
4.3 Export Key Pair 12
4.4 Import Signed Certificate 13
4.5 Sign the App package 13
4.6 Install APP from Web Manager 14
Chapter 5 SDK Installation 15
5.1 SDK Directory Structure 15
5.2 Install SDK 15
5.3 Install the Header files 16
5.4 RobustOS Pro Makefile Introduction 17
5.4.1 BuildPackage Variables 17
5.4.2 BuildPackage Defines 17
5.4.3 Other Important Variables 17
Chapter 6 Create a RobustOS Pro APP 18
6.1 APP Package Naming Rules 18
6.2 Create helloworld Package 18
6.3 Add Source Code 18
6.4 Add SDK Script 20
6.5 Generate Debian Files 22
6.6 Edit Makefile for SDK Building System 23
6.7 Build Helloworld Package 23
6.8 Install APP and Restart Router 24
6.9 Check APP Running Status 24
Chapter 7 Integrate into RobustOS Pro WebManager 25
7.1 Configuration Definition 25

7.2 Status Definition 26
7.3 Webpage Layout 27
7.4 UCI Attributes 27
7.5 UCI Tag 28
7.6 How to Get Configuration Settings 29
7.7 How to Change Configuration Settings 30
7.8 How to Find UCI Tags 30
7.9 USI Tag 31
7.10 How to Update Status 31
7.11 How to Get Status 31
7.12 How to Find USI Tags 33
7.13 How to Upload Custom File 34
7.14 How to Customize Navigation Bar. 40
Chapter 8 SDK C-Libraries 41
8.1 Librouter Data Type 41
8.2 Librouter Log API 41
8.3 Librouter UCI API 43
8.4 Librouter Environment Variable API 50
8.5 Librouter IPC API 52
8.6 Librouter Event API 53
Chapter 9 Pre-installed Tools 56
9.1 RobustOS Pro Private Commands 56
9.1.1 uci 56
9.1.2 usi 56
9.1.3 renv 57
9.1.4 sms 57
Chapter 10 Programming Tutorials 58
10.1 How to Debug the Application 58
10.1.1 Use Syslog 58
10.1.2 Use Shell with Root Privileges 58
10.2 How to Get System Status 58
10.3 How to Read/Write Data via Serial Port 58
10.4 How to Get DI Status 60
10.5 How to Trigger DO Changes 62
10.6 How to perform BLE scan 62
10.7 How to Send Data via CAN Interface 63
10.8 How to Mount SD Card Partition 63
10.9 How to Get AI Status 63
10.10 How to Subscribe Event 64
10.10.1 Supported Events 64
10.10.2 Code Example 65
10.11 How to Get new SMS 66
10.12 How to Send SMS 66
10.13 How to Send Email 66
10.14 How to Update DEB Packages 67
10.15 How to Import/Export XML Configuration File 67
10.16 Systemd 67

10.17 How to customize the dashboard area 68
10.18 How to Query Installed Services 69
10.18.1 Using systemctl 69
10.18.2 Using service 70
10.18.3 Using dpkg 71
Chapter 11 Docker Support 72

Chapter 1 Document Revision History

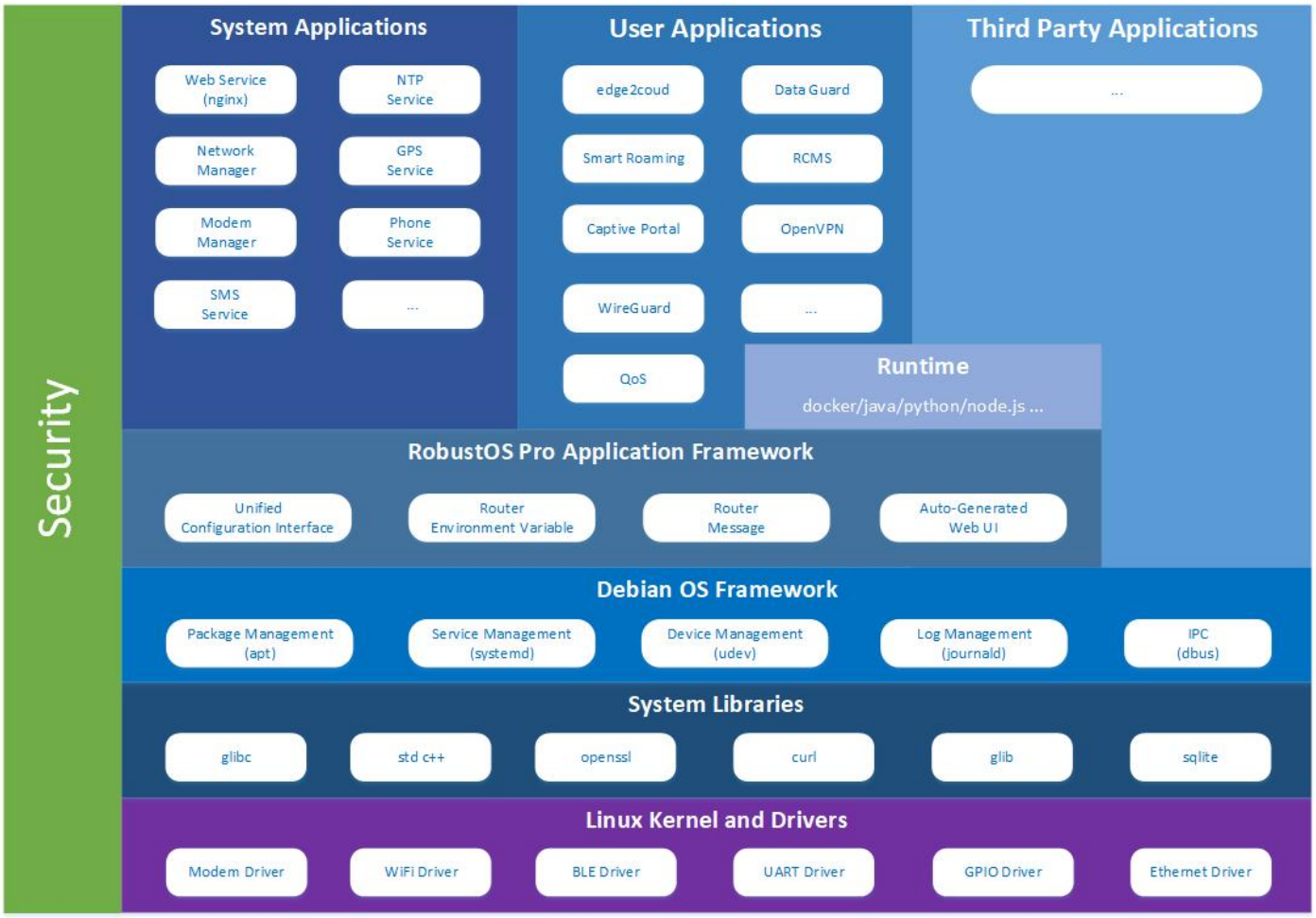
Revision	Author	Date	Description
2.0.0	Yaoguang Yang	2022-06-16	Initial version.
2.1.0	Yaoguang Yang	2023-08-11	Chapter 7.13 How to Upload Custom File (Added) Chapter 7.14 How to Customize Navigation Bar(Added) Chapter 10.4 How to Get DI Status (Modified) Chapter 11 Docker Support (Modified)
2.1.1	Jinquan Xiao	2023-09-01	Chapter 3 Basic Information (Modified) Chapter 4 Preparation for software development (Modified) Chapter 5 SDK Installation (Modified) Chapter 6 Create a RobustOS Pro APP (Modified) Chapter 10.3 How to Read/Write Data via Serial Port (Modified)
2.1.2	Yaoguang Yang	2023-10-11	Chapter 10.3 How to Read/Write Data via Serial Port (Modified)
2.1.3	Yaoguang Yang	2024-06-07	Support EG5101/EG5200/EV8100/R1520LG Chapter 3 Basic Information (Modified) Chapter 10.5 How to Trigger DO Changes (Modified) Chapter 10.6 How to perform BLE scan (Modified) Chapter 10.7 How to send data via CAN interface (Added)
2.1.4	Yaoguang Yang	2024-09-06	Support MG460 Chapter 10.18 How to Query Installed Services(Added)

Chapter 2 Overview

Robustel's series router allows 3rd party to develop their applications. We provide a software development kit (SDK) that offers a simple and fast way to implement customer-specific functions and applications.

RobustOS Pro includes an application framework and a set of system applications suitable for IoT gateways. The application framework provides a unified configuration and status management interface, a unified inter-process communication interface, and automatic generation of WEB pages to facilitate rapid application development.

The developer can directly develop software based on Debian, or based on RobustOS Pro application framework.



Chapter 3 Basic Information

3.1 Supported Hardware

- EG5100
- LG5100
- EG5120
- EG5101
- EG5200
- EV8100
- R1520LG
- MG460

3.2 Hardware Information

	EG5100		LG5100		R1520LG	EV8100	EG5101
CPU	i.MX 6ULL		i.MX 6ULL		i.MX 6ULL	i.MX 6ULL	i.MX 6ULL
ARCH	armv7(armhf)		armv7(armhf)		armv7(armhf)	armv7(armhf)	armv7(armhf)
Total Flash	8GB	8GB	8GB	8GB	8GB	8GB	8GB
Total RAM	1GB	512MB	1GB	512MB	512MB	512MB	512MB
Flash available for SDK	6GB	6GB	6GB	6GB	6GB	6GB	6GB
RAM available for SDK	756MB	256MB	756MB	256MB	256MB	256MB	256MB

	EG5120	EG5200	MG460
CPU	i.MX 8M Plus	i.MX 8M Plus	i.MX 8M Plus
ARCH	armv8(arm64)	armv8(arm64)	armv8(arm64)
Total Flash	16GB	32GB	64GB
Total RAM	2GB	4GB	4GB
Flash available for SDK	14GB	30GB	62GB
RAM available for SDK	1.5GB	2.5GB	2.5GB

3.3 Hardware Resources

3.3.1 EG5100

- **/dev/ttyCOM1** device file of RS232/RS485 serial port
- **/dev/ttyCOM2** device file of RS232/RS485 serial port
- **/dev/ttyGPS** device file of GPS data port (optional)
- **/dev/DI1** device file of DI(Digital Input)
- **/dev/DI2** device file of second DI
- **/sys/class/leds/do1/** file path of first DO(Digital Output)
- **/sys/class/leds/do2/** file path of second DO
- **/mnt/usb** USB disk mount point
- **/dev/ttymxc3** device file of bluetooth(optional)
- **/sys/class/net/can0/** CAN network interface(optional)

3.3.2 LG5100

- **/dev/ttyCOM1** device file of RS232/RS485 serial port
- **/dev/ttyCOM2** device file of RS232/RS485 serial port
- **/dev/ttyGPS** device file of GPS data port (optional)
- **/dev/DI1** device file of DI(Digital Input)
- **/dev/DI2** device file of second DI
- **/sys/class/leds/do1/** file path of first DO(Digital Output)
- **/sys/class/leds/do2/** file path of second DO
- **/mnt/usb** USB disk mount point
- **/dev/spidev1.0** device file of lora

3.3.3 EG5120

- **/dev/ttyCOM1** device file of RS232/RS485 serial port
- **/dev/ttyCOM2** device file of RS232/RS485 serial port
- **/dev/ttyGPS** device file of GPS data port (optional)
- **/dev/DI1** device file of DI(Digital Input)
- **/dev/DI2** device file of second DI
- **/sys/class/leds/do1/** file path of first DO(Digital Output)
- **/sys/class/leds/do2/** file path of second DO
- **/mnt/usb** USB disk mount point
- **/dev/ttymxc3** device file of bluetooth(optional)

3.3.4 R1520LG

- **/dev/ttyCOM1** device file of RS232 serial port
- **/dev/ttyCOM2** device file of RS485 serial port
- **/mnt/usb** USB disk mount point

- **/dev/spidev0.0** device file of lora

3.3.5 EV8100

- **/dev/ttyCOM1** device file of RS485 serial port
- **/dev/ttyCOM2** device file of RS232 serial port
- **/dev/DI1** device file of DI(Digital Input)
- **/dev/DI2** device file of second DI
- **/dev/DI3** device file of third DI
- **/dev/DI4** device file of fourth DI
- **/mnt/usb** USB disk mount point
- **/dev/ttymxc1** device file of bluetooth(optional)
- **/sys/class/net/can0/** CAN network interface

3.3.6 EG5101

- **/dev/ttyCOM1** device file of RS232 serial port
- **/dev/ttyCOM2** device file of RS485 serial port
- **/mnt/usb** USB disk mount point

3.3.7 EG5200

- **/dev/ttyCOM1** device file of RS232/RS422/RS485 serial port
- **/dev/ttyCOM2** device file of RS232/RS422/RS485 serial port
- **/dev/ttyGPS** device file of GPS data port (optional)
- **/dev/DI1** device file of DI(Digital Input)
- **/dev/DI2** device file of second DI
- **/mnt/usb** USB disk mount point
- **/dev/ttymxc3** device file of bluetooth(optional)

3.3.8 MG460

- **/dev/ttyCOM1** device file of RS232/RS422/RS485 serial port
- **/dev/ttyCOM2** device file of RS232/RS422/RS485 serial port
- **/dev/ttyGPS** device file of GPS data port (optional)
- **/dev/DI1** device file of DI(Digital Input)
- **/dev/DI2** device file of second DI
- **/mnt/usb** USB disk mount point

3.4 Recommended Development Environment

1. The device has a built-in compiler toolchain. It is recommended to compile and debug directly in the device, which is more convenient than cross-compilation. The development method is the same as on the computer.
2. Robustel also provides a Docker image to help you compile quickly and avoid the hassle of configuring your environment. To install Docker please refer to Chapter 11.

Loading image.

```
$ docker load -i robustos_pro_arm64_v2.1_docker.tar.tar          #For armv8 devices(EG5120)
$ #or
$ docker load -i robustos_pro_armhf_v2.1_docker.tar              #For armv7 devices(EG5100/LG5100)
```

Creating a shared directory.

```
$ mkdir -p /root/work
```

Starting Docker.

```
$ docker run --privileged -it    -v /root/work:/mnt/ --net=host robustos_pro_arm64:v2.1 /bin/sh    #For armv8 devices(EG5120)
$ #or
$ docker run --privileged -it    -v /root/work:/mnt/ --net=host robustos_pro_armhf:v2.1 /bin/sh    #For armv7 devices(EG5100/LG5100)
```

3.5 Recommended IDE Tools

- Vi
- Notepad++
- Visual Studio Code

3.6 Supported Programming Languages

Most programming languages are supported, such as C/C++/Python/Java/Node.js, and so on.

3.7 C Library

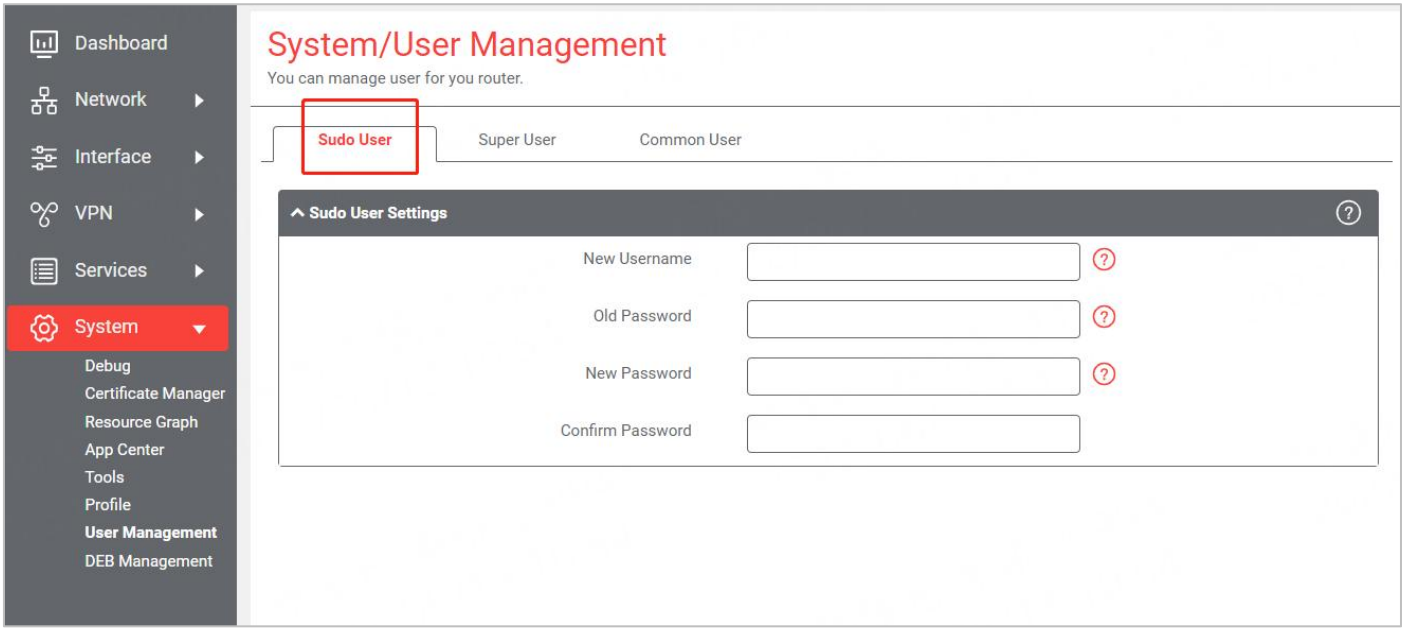
- glibc

Chapter 4 Preparation for software development

Based on security considerations, only authenticated users can publish Apps. The developer needs to generate an RSA key pair and send the public key to Robustel. Robustel will generate a certificate based on the key. The user certificate can be installed on the device via WEB UI.

4.1 Add Sudo User

Login to the WEB GUI to add a sudo user, after which you connect to SSH for development through the sudo user.

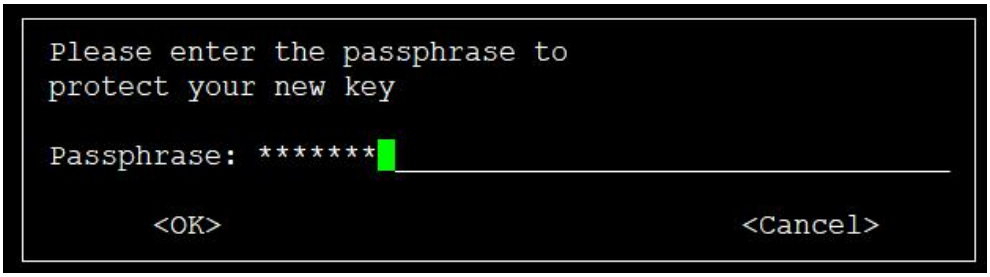


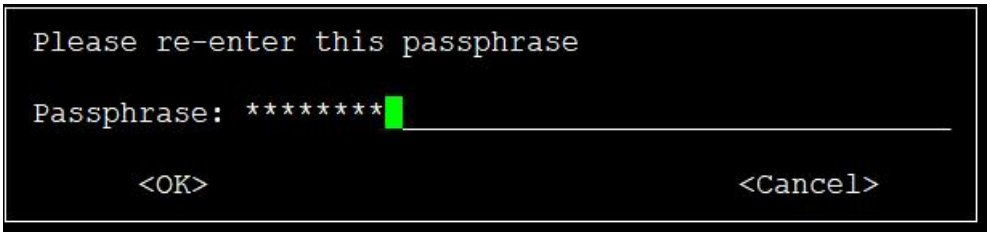
4.2 Generate RSA key

Please use the GnuPG tool to generate RSA key pairs (4096 bits). You can execute the following commands directly on the device.

```
$ gpg --full-generate-key
```

```
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) Y

GnuPG needs to construct a user ID to identify your key.

Real name: robustel-sdk-2023
Email address: support@robustel.com
Comment: robustel-sdk-2023
You selected this USER-ID:
    "robustel-sdk-2023 (robustel-sdk-2023) <support@robustel.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
```





4.3 Export Key Pair

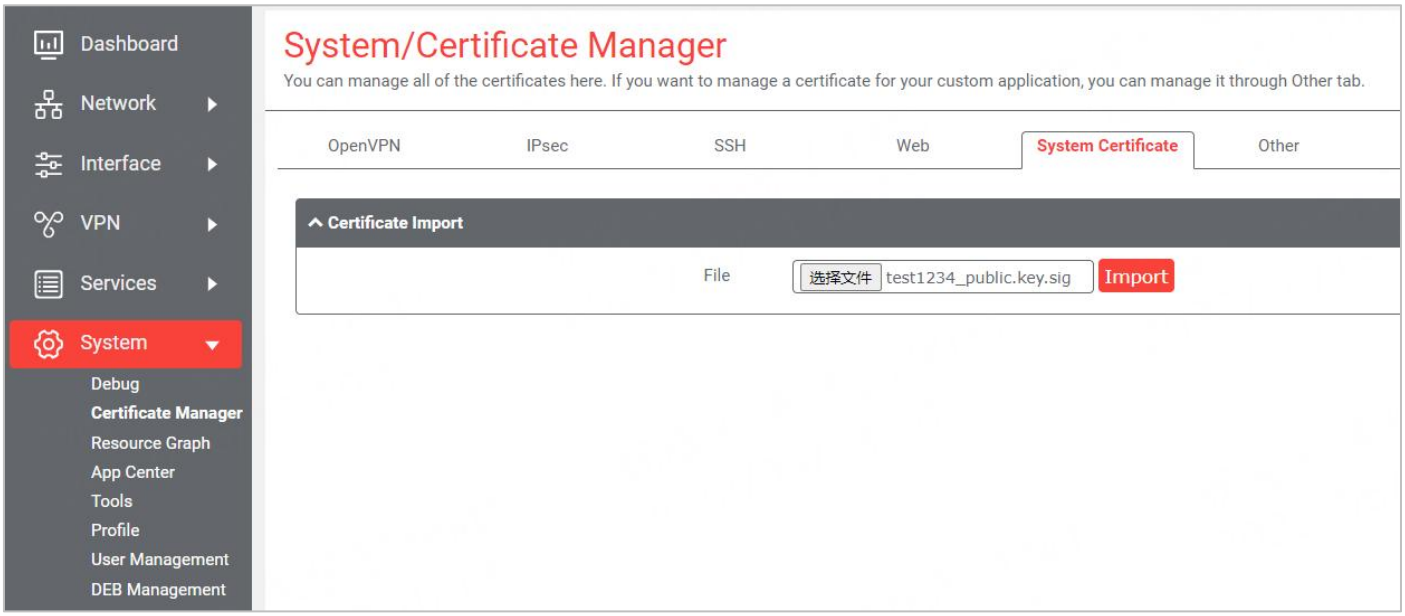
Please send the RSA public key to Robustel for signature and keep the private key.

```
$ gpg --list-keys
$ gpg --list-secret-keys
$ gpg --armor --output "sample_public.key" --export robustel-sdk-2023
```

```
$ gpg --armor --output "sample_private.key" --export-secret-keys robustel-sdk-2023
$ ls
sample_private.key
sample_public.key
```

4.4 Import Signed Certificate

Obtain the signed certificate from Robustel (or use the certificate provided with the Robustel SDK) and import it via the WEB GUI.



4.5 Sign the App package

Import the sample_private.key, and use it to sign the APP package.

```
$ gpg --import sample_private.key
$ gpg --list-keys
$ gpg --list-secret-keys
```

```
gpg --list-secret-keys
/home/xiaojinquan/.gnupg/pubring.kbx
---------------------------------
sec   rsa4096 2023-08-29 [SC]
      5DCCB2AC6063A9CCA019B6A73A4A654BD35C011D
uid           [ultimate] test1234 (test1234) <support@robustel.com>
ssb   rsa4096 2023-08-29 [E]

sec   rsa4096 2023-08-29 [SC]
      DB3413F64E44B08FC52E61D3B64CDA474FDBA30E
uid           [ultimate] test1234 (test1234) <support@robustel.com>
ssb   rsa4096 2023-08-29 [E]

sec   rsa4096 2023-08-29 [SC]
      11C96105CA76AF621F76E866EECFCC19FB80E23C
uid           [ultimate] xjqg1234 (xjqg1234) <support@robustel.com>
ssb   rsa4096 2023-08-29 [E]

sec   rsa4096 2023-09-01 [SC]
      0A249787AD9C035C242E2167E73C4D524E457F85
uid           [ultimate] robustel-sdk-2023 (robustel-sdk-2023) <support@robustel.com>
ssb   rsa4096 2023-09-01 [E]
```

$ export GPG_TTY=$(tty)

$ dpkg-sig -k robustel-sdk-2023 --sign mytest helloworld_2.0.0_armhf.deb

Processing helloworld_2.0.0_armhf.deb...

gpg: using "robustel-sdk-2023" as default secret key for signing

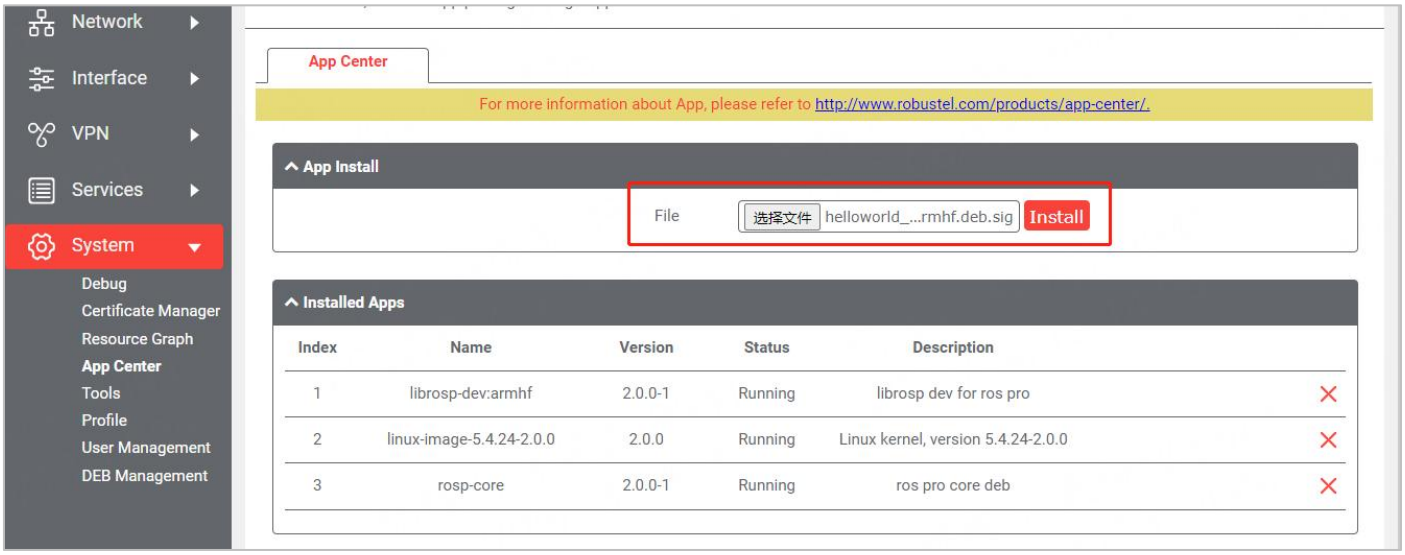
$ cp helloworld_2.0.0_armhf.deb helloworld_2.0.0_armhf.deb.sig

$ ls

helloworld_2.0.0_armhf.deb helloworld_2.0.0_armhf.deb.sig

4.6 Install APP from Web Manager

Install helloworld_2.0.0_armhf.deb.sig which is signed.



Note:

If you do not plan to develop based on the RobustOS Pro application framework, you can skip the following chapters directly.

Chapter 5 SDK Installation

SDK only contains examples and demos, and Debian control files to make a Debian package. The developer can upload the SDK files directly to the device from the SCP/FTP server or USB. The SDK is named ROS-SDK-eg51xx-XX.tgz, where the ”XX” represents the version number.

5.1 SDK Directory Structure

```
ROS-SDK-eg51xx-XX
├── bin
|   ├── eg51xx-firmware-xxx.bin
|   └── librosp-dev_xxx.deb
├── Makefile
├── readme
├── sample-can
├── sample-cpp
├── sample-dido
├── sample-gps
├── sample-serial
├── sample-shell
└── sample-uci
```

- **bin** - librosp-dev:armhf deb and rosp-core.
- **Makfile** – Top Makefile used for package management.
- **sample-can** - SDK sample can for C programms
- **sample-cpp** - SDK example for C++ programms
- **sample-dido** - sample-dido for Debian
- **sample-gps** - sample_gps
- **sample- serial** - SDK sample serial for C programms
- **sample- shell**- SDK example for scripts
- **sample-uci** - SDK example for C programs
- **sample-upload** - SDK example for uploading custom file

5.2 Install SDK

The developer should install compilation toolchains first.

```
$ apt update
$ apt-get install build-essential
```

You can extract the SDK to any place you like, for example:

```
$ tar xzvf ROS-SDK-eg51xx-XX.tgz -C /opt
```

There are some examples distributed with the SDK. You can build an SDK example for test, for example:

```
$ cd /opt/ROS-SDK-eg51xx-XX
$ make sample-shell
$ make deb-install
```

If you get errors compiling, you can try installing the following dependency libraries.

```
$ apt install libevent-dev libsystemd-dev libpam0g-dev libbluetooth-dev
```

The APP package will be put into install_dir/ directory after successful building.

```
$ ls install_dir
$ sample-shell_xxx_armhf.deb
```

You can install the app and check the status with the command.

```
$ dpkg -i --force-confask --force-confnew install_dir/sample-shell_2.0.0_armhf.deb
$ dpkg -l sample-shell
$ dpkg -L sample-shell
```

You can also install the app through the WEB UI. Please refer to Chapter 4.

Restart router.

```
$ systemctl restart router
```

5.3 Install the Header files

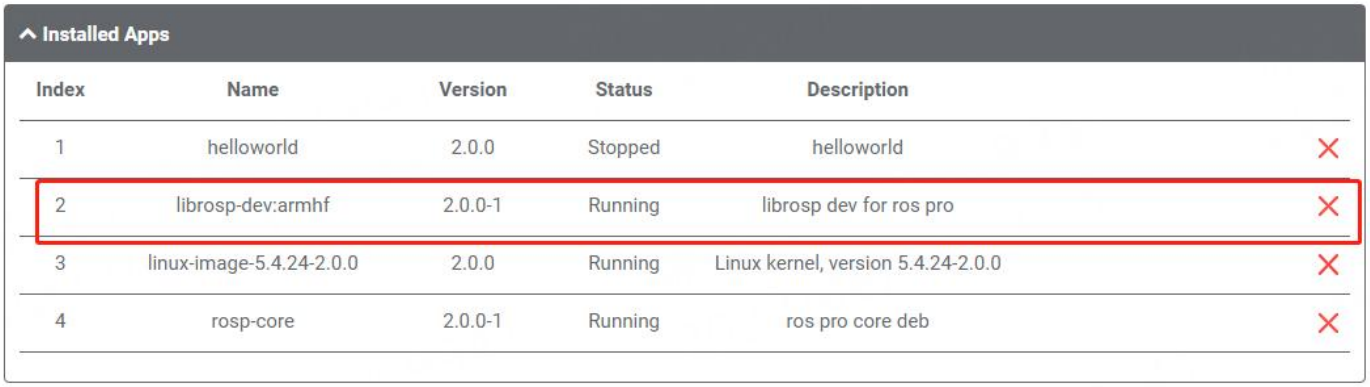
If the customer application needs to be managed by Web, the developer needs to install header files provided by Robustel. LG5100 uses the same as EG5100.

```
$ librosp-dev_2.0.0-1_armhf.deb
```

Install librosp-dev:armhf deb through the Web Manager page and follow the prompts to restart the router.



Check the librosp-dev:armhf version to ensure that the deb file is installed correctly.



5.4 RobustOS Pro Makefile Introduction

An application is called a package in the SDK build system. The package makefile provides the necessary steps to compile the package. The makefile has been transformed into an object-oriented template which simplifies the entire ordeal, and everything is hidden in other makefiles and abstracted to the point where you only need to specify a few variables and define them.

5.4.1 BuildPackage Variables

- **NONE**.

5.4.2 BuildPackage Defines

- **NONE**.

5.4.3 Other Important Variables

- **NONE.**

Chapter 6 Create a RobustOS Pro APP

Here is a step-by-step guide for building a helloworld package with RobustOS Pro SDK.

6.1 APP Package Naming Rules

The app should be named according to the following naming rules:

1. The app package name should not include the midline character “-”.
2. The length of the package name should be less than 18 characters.

6.2 Create helloworld Package

```
$ tar xzvf ROS-SDK-eg51xx-XX.tgz -C /opt
$ cd /opt/ROS-SDK-eg51xx-XX
$ mkdir helloworld/src/
```

6.3 Add Source Code

Edit helloworld.c and save it in /opt/ROS-SDK-eg51xx-XX/helloworld/src.

```
#include <syslog.h>
#include <unistd.h>
int main()
{
	syslog(LOG_DEBUG, “hello world”);
	pause();
	return 0;
}
```

Edit Makefile and save it in /opt/ROS-SDK-eg51xx-XX/helloworld.

```
SRCDIR = src
INCDIR = include
OBJDIR = obj
BINDIR = obj

VER = 0
REV = 1
PLATFORM = eg51xx
PREDEFINED_MACRO_PLATFORM = PLATFORM_EG51XX

TARGET = helloworld

CFLAGS += -Wall -I$(INCDIR) -I/usr/include/libxml2/ -I/usr/include/librouter/
```

```
CFLAGS += -DVERSION=\"$(VER)\" -DREVISION=\"$(REV)\" -DPLATFORM=\"$(PLATFORM)\" -D$(PREDEFINED_MACRO_PLATFORM)

LDFLAGS = -Wl,--as-needed
LIBS = -lrouter -levent_core -lpam -lcurl -lcrypto -lpthread -lxml2

SRCS = $(wildcard $(SRCDIR)/*.c)
OBJS = $(SRCS:$(SRCDIR)/%.c=$(OBJDIR)/%.o)
DEPS = $(OBJS:%.o=%.d)

all:compile
compile:$(BINDIR)/$(TARGET)

$(BINDIR)/$(TARGET):$(OBJS) $(DEPS)
        @mkdir -p $(BINDIR)
        $(CC) $(LDFLAGS) -o $@ $(OBJS) $(LIBS)

$(OBJS):$(OBJDIR)/%.o:$(SRCDIR)/%.c
        @mkdir -p $(OBJDIR)
        $(CC) $(CFLAGS) -c $< -o $@

$(DEPS):$(OBJDIR)/%.d:$(SRCDIR)/%.c
        @mkdir -p $(OBJDIR)
        @$(CC) -MM -MT"$(@:%.d=%.o)" $(CFLAGS) $< > $@

.PHONY : clean
clean :
        rm -rf obj

.PHONY : install
install: compile
        mkdir -p $(DESTDIR)/usr/bin
        install -m 0755 $(BINDIR)/$(TARGET) $(DESTDIR)/usr/bin
        mkdir -p $(DESTDIR)/etc/sdk/
        cp -ar ./files/sdk.sh    $(DESTDIR)/etc/sdk/$(TARGET)
        chmod +x $(DESTDIR)/etc/sdk/$(TARGET)


.PHONY : uninstall
uninstall:

ifeq ($(OBJDIR), $(wildcard $(OBJDIR)))
-include $(DEPS)
Endif
```

6.4 Add SDK Script

In addition to compiling the program, you should write an SDK script to tell the router system how to start and stop the program. The SDK provides some functions so that you can more easily do the script, which is located in usr/lib/functions. There are several commands you should support in the script.

- **start** - basic command, start the program.
- **stop** - basic command, stop the program.
- **restart** - stop and restart the program.
- **status** - check whether the program is running or stopped.
- **on_startup** - run on system startup, usually run start command.
- **on_reboot** - run before the system reboots, usually run stop command.
- **on_link_up** - run when the primary link is up, usually run the start command.
- **on_link_down** - run when the primary link is down, usually run stop command.
- **status_sync** - (optional) run when someone queries status, used for sync status from other files to usi format.

Add an SDK script to tell the router how to start or stop the application.

```
$ mkdir helloworld/files
$ cp sample-gps/files/sdk.sh h helloworld/files/
$ vi helloworld/files/sdk.sh
```

Edit sdk.sh based on the example, just changing the PROG variable is enough for this application.

```
#!/bin/sh

. /usr/lib/functions

RETVAL=0
PROG=helloworld

start()
{
   if pidof ${PROG} | grep -q [0-9]; then
         echo ${PROG} is running
   else
      ${PROG} &
   Fi
   return $?
}

stop()
{
   killproc ${PROG}
   RETVAL=$?
   return $RETVAL
}
```

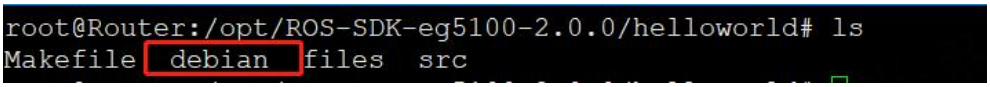
```
restart()
{
   stop
   start
}

status()
{
   pidof -o %PPID ${PROG} >/dev/null 2>&1
   RETVAL=$?
   [ $RETVAL -eq 0 ] && echo ${PROG} is running || echo ${PROG} is stopped
   return $RETVAL
}

status_sync()
{
   return 0
}


case "$1" in
   start)
        start
        ;;
   stop)
        stop
        ;;
   restart)
        restart
        ;;
   status)
        status
        ;;
   on_startup)
        start
        ;;
   on_link_up)
             stop
        start
        ;;
   on_link_down)
        stop
        ;;
   on_reboot)
        stop
        ;;
   status_sync)
```

```
        status_sync
        ;;
    *)
        echo $"Usage: $0 {start|stop|restart|status|on_startup|on_link_up|on_link_down|on_reboot|status_sync}"
        exit 2
esac

exit $?
```

6.5 Generate Debian Files

```
$ cd helloworld/
$ debmake -e    support@robustel.com    -p helloworld    -u 2.0.0 -f "rbt" -m -n -x2
```

Debian files will be generated automatically.



Edit debian/README.Debian

```
helloworld for Debian


      helloworld for Debian


 -- rbt <support@robustel.com>    Mon, 13 Jun 2022 13:50:43 +0800
```

Edit debian/changelog

```
helloworld (2.0.0) unstable; urgency=low


   * Initial release.


 -- rbt <support@robustel.com>    Mon, 13 Jun 2022 13:50:43 +0800
```

Edit debian/control

```
Source: helloworld
Section: admin
Priority: optional
Maintainer: rbt <support@robustel.com>
Build-Depends: debhelper (>=11~)
Standards-Version: 4.1.4
Homepage: http://www.robustel.com.cn/

Package: helloworld
Architecture: any
Depends: ${misc:Depends}, ${shlibs:Depends}
Description: helloworld
      helloworld
```

Edit debian/rules

```
#!/usr/bin/make -f
# You must remove unused comment lines for the released package.
export DH_VERBOSE = 1
export DEB_BUILD_MAINT_OPTIONS = hardening=+all
export DEB_CFLAGS_MAINT_APPEND  = -Wall -pedantic
export DEB_LDFLAGS_MAINT_APPEND = -Wl,--as-needed

%:
	dh $@
#override_dh_auto_install:
#	dh_auto_install -- prefix=/usr

#override_dh_install:
#	dh_install --list-missing -X.pyc -X.pyo
```

Create debian/compat

```
$ cat /etc/os-release
```

```
PRETTY_NAME="Debian GNU/Linux 11 (bullseye)"
NAME="Debian GNU/Linux"
VERSION_ID="11"
VERSION="11 (bullseye)"
VERSION_CODENAME=bullseye
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

```
$ echo 11 > ./debian/compat
```

6.6 Edit Makefile for SDK Building System

```
$ vi Makefile
```

Edit Makfile based on the example.

```
……
helloworld:
	$(call func1,helloworld,helloworld,$(BUILD_DIR))
……
.PHONY: …… helloworld
```

6.7 Build Helloworld Package

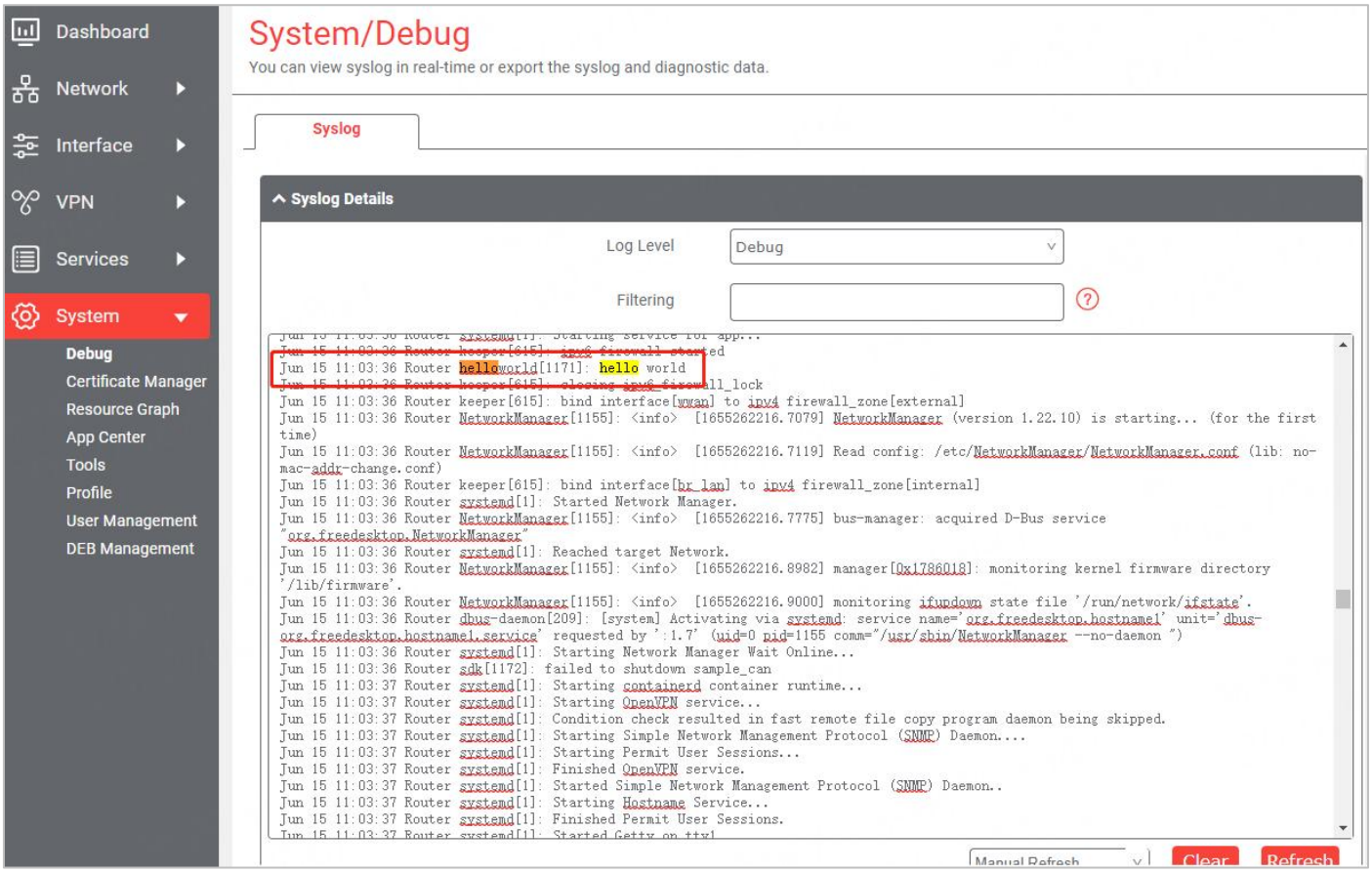
```
$ make helloworld
$ make deb-install
```

Then you can find the deb file in install_dir/.

6.8 Install APP and Restart Router

```
$ dpkg -i --force-confask --force-confnew install_dir/helloworld_2.0.0_armhf.deb
$ systemctl restart router
```

6.9 Check APP Running Status



Chapter 7 Integrate into RobustOS Pro WebManager

For a generic application, the operators need a user interface to change configuration and show running status. The SDK provides an easy way to integrate the configuration and status of your application into the Web Manager of the router. You do not need to know any web programming languages, just add a simple file in a pre-defined format, then the web page will be generated automatically.

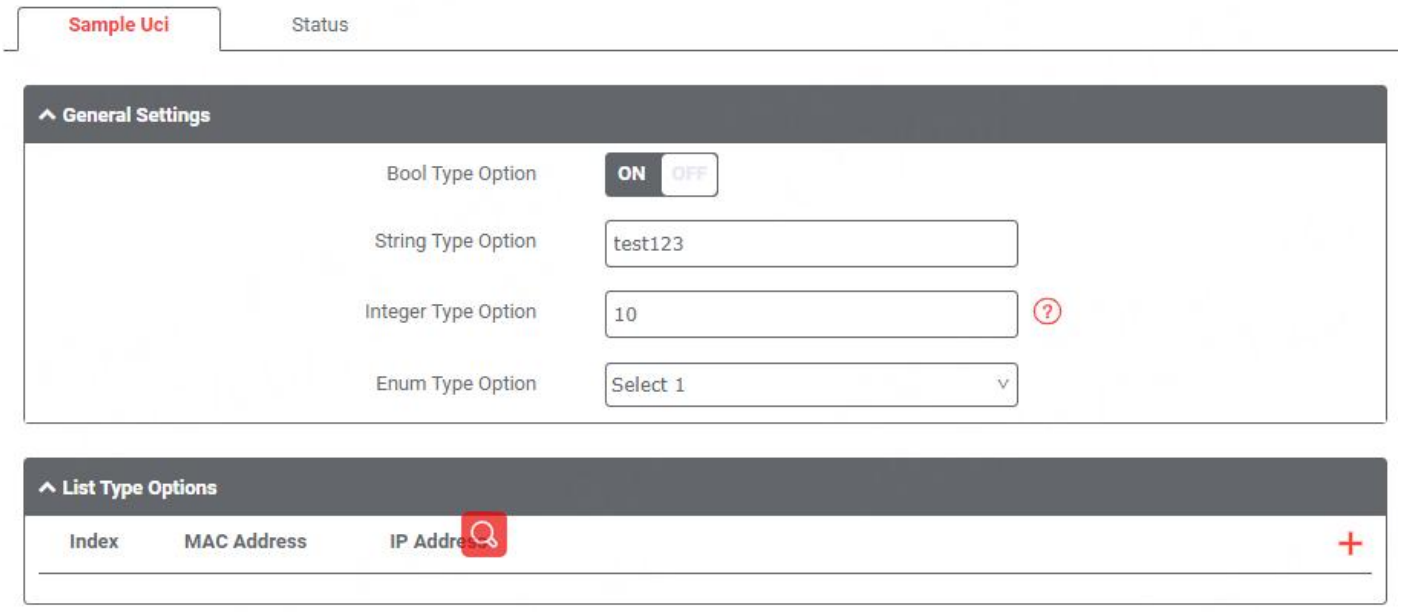
If the application you are going to develop does not have any configuration data, or you are planning to handle configurations in some other way, you can skip this chapter.

If you would like to integrate your APP into the Web Manager of the router, just as the APP was released officially, you should follow the rules called the UCI subsystem. The abbreviation UCI stands for Unified Configuration Interface and is intended to centralize the configuration of a router. The UCI subsystem uses XML to describe the data structure of configuration or items of status. Here we use an example to explain the UCI subsystem, and you can find the example in the package/sample_uci directory.

7.1 Configuration Definition

```
<sample_uci desc="configuration of SDK Sample Uci" group="Services" group_weight="40" menu="SDK Sample Uci" menu_weight="901" oid="1001">
    <option_bool type="bool" desc="Bool Type Option" default="false" tab="Sample Uci" tab_weight="0" area="General Settings" oid="1" />
    <option_string default="test" desc="String Type Option" oid="2" />
    <option_int type="int" desc="Integer Type Option"    range="1..100" default="10" row_help="This is an integer type config option." oid="3" />
    <option_enum type="enum" desc="Enum Type Option" range="select1 select2" default="select1" range_desc="Select_1 Select_2" oid="4" />
    <option_list desc="List Example" area="List Type Options" max_entry_num="20" oid="5" >
        <id type="int" range="1..20" desc="Index" col_width="80" area="Popup Window Title" oid="2" />
        <mac type="MAC_ADDR" unique="true" desc="MAC Address" col_width="140" oid="3" />
        <ip type="IP_ADDR" unique="true" desc="IP Address" col_width="140" oid="4" />
    </option_list>
</sample_uci>
```

The web page corresponding to the above XML is shown below:



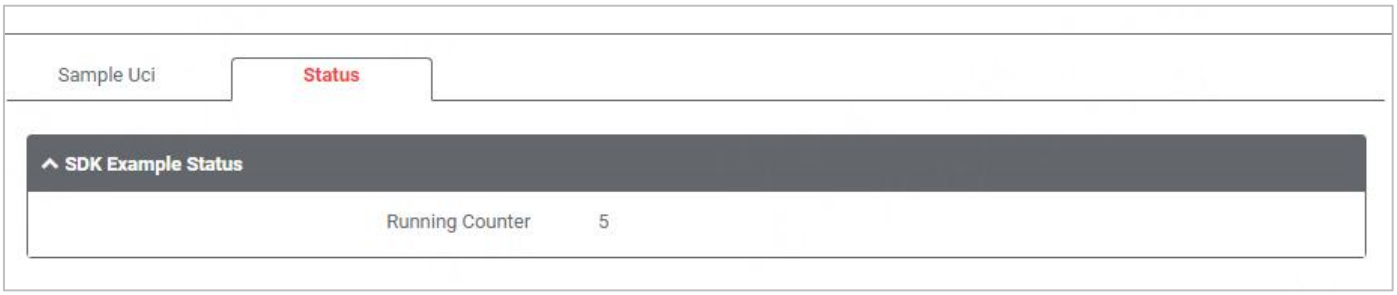
And you can also get it in CLI:

```
# show sample_uci all
option_bool = true
option_string = test123
option_int = 10
option_enum = select1
```

7.2 Status Definition

```
<sample_uci cli_help="status of SDK Sample Uci" group="Services" group_weight="40" menu="SDK Sample Uci" menu_weight="901"
        oid="1000" >
    <counter desc="Running Counter" tab="Status" tab_weight="20" area="SDK Example Status" oid="1" />
</sample_uci>
```

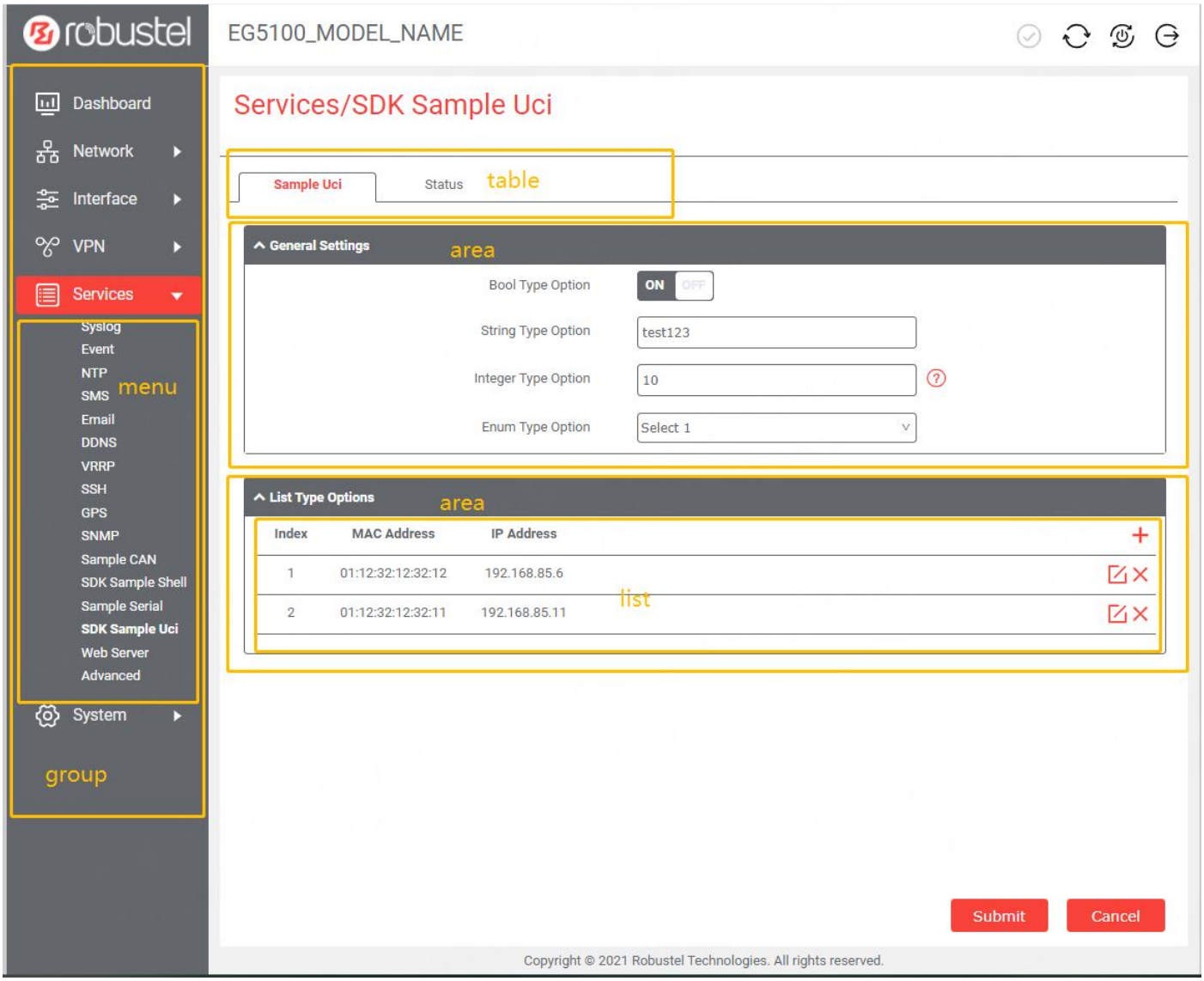
The web page corresponding to the above XML is shown below:



And you can also get it in CLI:

```
# status sample_uci
counter = 5
```

7.3 Webpage Layout



7.4 UCI Attributes

- **group** - the group name in the navigation bar, please set it to 'Services'.
- **group_weight** - specifies the location of the group, please set it to 40.
- **menu** - the menu name in the navigation bar.
- **menu_weight** - specifies the location of the menu, the default is 1000.
- **tab** – the name of the tab on the page, you can define more than 1 tab if you need it, e.g. one tab for configuration, one tab for status.
- **area** - defines an area, used for grouping the configuration options.
- **area_help** - optional, you can add some help info for this area.
- **row_help** - optional, one config option takes up one row, you can add some help info for this option.
- **desc** - description of config option.
- **type** - the type of config option, including string, int, enum, and bool. The default type is a string. The string type and int type element will be displayed as a one-line text input field in the browser. The enum type element will be displayed as a drop-down list. The bool type element will be displayed as a radio button. There are also some

internal types used for special string verification, including IP_ADDR, NETMASK, MAC_ADDR. You can use regular expressions for any other string verification.

- **range** - specifies the range of the config value. It can be the length range of a string, or numerical range of an integer, or the available options for enum type.
- **range_desc** - optional, you can specify the description of enumerated options for enum type.
- **default** - specifies the default value of this config option.
- **max_entry_num** - specifies the maximum rows of the list, also used for indicates a list type configuration.
- **pattern** - specifies the regular expression used for verification.
- **unique** - the setting value of this option should be unique in all list entries.
- **col_width** - optional, all the config options in a list will be displayed in a pop-up window, you can add a col_width attribute to show this config option in the overview table. The col_width attribute also specifies the width of this row in a table. The width range is 10-200.
- **oid** - optional, specifies the OID used for SNMP management. The top level OID should be greater than 1000. The MIB file for this application will be generated automatically if oid attribute is specified.

7.5 UCI Tag

It's important to understand UCI tag before coding. Every configuration option can be referenced by a unique tag, which is mostly like C-style variable reference. For example, use syslog.level to refer to the debug level, or use ipsec.tunnel[1].mode to refer to the mode of the first IPSec tunnel. The list ID starts from 1, not 0. For more detailed information about UCI API, please refer to the next chapter. Tagged method for status is the same as configuration.
There is an easy way to find out the corresponding tag of a certain configuration option or status item. You can check the tag of configuration with command shown in the CLI system, and status command for tag of status.

```
// IP address of br_lan, the UCI tag is lan_links.link[1].ip4_setting.ip4_addrs.ip4_addr[1]
# show lan_links all
link {
    id = 1
    name = LAN1
    type = bridge
    interface_bridge = br_lan
    interface_ethernet = ""
    interface_vlan = ""
    desc = "default lan"
    firewall_zone = internal
    ip4_setting {
        ip4_dummy = ""
        ip4_addrs {
            ip4_addr = 192.168.0.1/24
        }
    }
    dhcp {
        pool_start = 192.168.0.2
        pool_end = 192.168.0.100
        primary_dns = ""
        secondary_dns = ""
        lease_time = 120
```

```
        }
        ip6_setting {
                mode = auto
                nat = false
                ip6_addr = fd00::1/64
        }
}

// get MAC address from status, the UCI tag is lan_links.link_status[1].mac
// get RX bytes stats of br_lan, the UCI tag is lan_links.link_status[1].rx_bytes
# status lan_links
link_status {
        id = 1
        interface = br_lan
        mac = 00:E1:7C:68:31:F5
        ip4_addr = 192.168.0.1
        ip6_addr = fe80::19fb:f7ea:b3b0:ad65
        rx_bytes = 359737
        rx_packets = 2436
        tx_bytes = 1301436
        tx_packets = 2010
}
connected_device {
        id = 1
        ip = 192.168.0.56
        mac = 00:E0:4C:40:99:41
        interface = br_lan
        inactive_time = 0s
}
connected_device {
        id = 2
        ip = fe80::98c7:fb0c:11ee:4c1b
        mac = 00:E0:4C:40:99:41
        interface = br_lan
        inactive_time = 66s
}
```

7.6 How to Get Configuration Settings

```
int i;
int max_entry_num;
char *p;

# get string type config setting
p = uci_get("sample_uci.option_string");
```

```
# get integer type config setting
i = uci_get_int("sample_uci.option_int");
# check bool type config option
if (uci_check_bool("sample_uci.option_bool"))
{
    // do something
}
# check bool type config option
if (uci_match("sample_uci.option_enum", "select1"))
{
    // do something
}
# list traversal
max_entry_num = uci_list_get_max_entry_num("sample_uci.option_list");
for (i=1; i<= max_entry_num; i++)
{
    p = uci_list_get("sample_uci.option_list", i, "ip");
    printf("IP Address = %s\n", p);
}
```

7.7 How to Change Configuration Settings

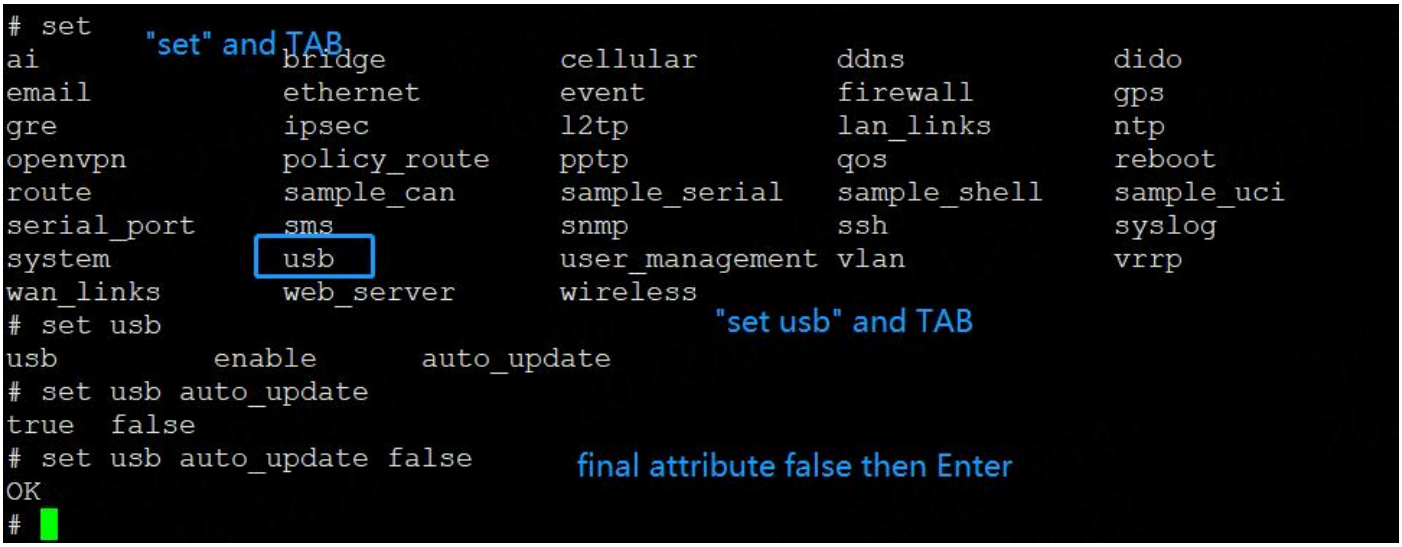
Find tag of config option in the CLI firstly.

```
# # show usb all
enable = true
auto_update = false
```

```
uci_init();
# enable usb auto update
uci_set("usb.auto_update", "true");
# just apply the changes, do not save
system("uci apply");
```

7.8 How to Find UCI Tags

Using the administrator account, login the router by telnet and it will enter CLI mode. In CLI mode, CLI shell will automatically prompt the next valid menu by typing the TAB key. For UCI tag, type the UCI command "set", then TAB key, select the menu and TAB again, till reach the final attribute. For example, try to disable usb auto_update .



Till now, we can know the UCI tag name for setting usb auto_update is “usb.auto_update” and the value is “false”, example code as below.

```
uci_init();
# disable usb auto update
uci_set(“usb.auto_update”, “false”);
# just apply the changes, do not save
system(“uci apply”);
```

7.9 USI Tag

USI tag is for the status information, similar with UCI tag, also define by XML file and the attribute is the same. Please refer to Chapter6.5 UCI tag.

7.10 How to Update Status

```
int counter = 10;
usi_update(“sample_uci.counter”, “10”);
usi_printf(“sample_uci.counter”, “%d”, counter);
```

7.11 How to Get Status

Find tag of status in the CLI firstly.

```
# status lan_links
link_status {
    id = 1
    interface = br_lan
    mac = 00:E1:7C:68:31:F5
    ip4_addr = 192.168.0.1
    ip6_addr = fe80::f551:943f:d444:b2f5
    rx_bytes = 8016
    rx_packets = 118
```

```
        tx_bytes = 9214
        tx_packets = 95
}
connected_device {
        id = 1
        ip = 192.168.0.56
        mac = 00:E0:4C:40:99:41
        interface = br_lan
        inactive_time = 1s
}
connected_device {
        id = 2
        ip = fe80::98c7:fb0c:11ee:4c1b
        mac = 00:E0:4C:40:99:41
        interface = br_lan
        inactive_time = 81s
}
```

```
# status system
firmware_version = "2.0.0"
firmware_version_full = "2.0.0 (e8cda59)"
kernel_version = 5.4.24-2.0.0
hardware_version = 0.0
operation_system = "Debian GNU/Linux 11.3"
device_model = EG5100-A
serial_number = "1234567890"
temperature_interval = 60.0
uptime = "0 days, 03:58:51"
system_time = "Wed Jun 15 15:35:21 2022 (NTP not updated)"
ram_usage = 326M/448M
cpu_usage = "1123539s Idle/1413746s Total /1 cpus"
disk_usage = 2.5G/3.6G
char *ip4addr;
char *model;
model = usi_get("system.device_model");

if (NULL != model)
{
        printf("Device Model = %s\n", model);
        free(model);
}

ip4addr= usi_get("lan_links.link_status[1].ip4_addr");
if (NULL != ip4addr)
{
        printf("ip4_addr= %s\n", ip4addr);
        free(ip4addr);
```

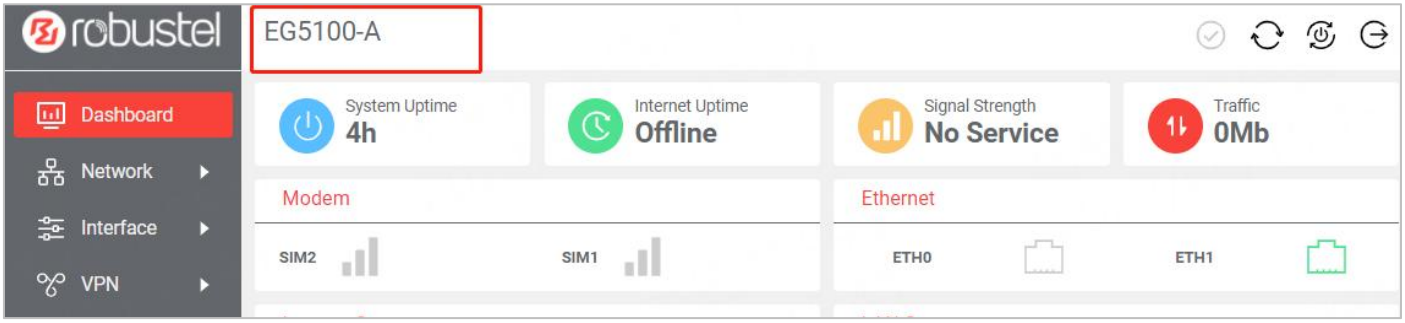
```
}
```

7.12 How to Find USI Tags

Similar with chapter 6.8 “How to find UCI tags”. For USI tag, type the USI command “status”, then TAB key, select the menu and TAB again, till reach the final attribute. For example, try to get device_model showed in web page.

```
# status
ai              bridge          cellular        ddns            dido
email           ethernet        event           firewall        gps
gre             ipsec           l2tp            lan_links       ntp
openvpn         policy_route    pptp            qos             reboot
route           sample_can      sample_serial   sample_shell    sample_uci
serial_port     sms             snmp            ssh             syslog
system          usb             vlan            vrrp            wan_links
web_server      wireless
# status system
firmware_version = ""
firmware_version_full = " ()"
kernel_version = 5.4.24-2.0.0
hardware_version = 0.0
operation_system = "Debian GNU/Linux 11.3"
device_model = EG5100-A
serial_number = ""
temperature_interval = 60.0
uptime = "0 days, 04:08:26"
system_time = "Wed Jun 15 15:44:56 2022 (NTP not updated)"
ram_usage = 336M/448M
cpu_usage = "1123539s Idle/1471228s Total /1 cpus"
disk_usage = 2.5G/3.6G
#
```


It is the same as showed in web page.



Get the device_model in code:

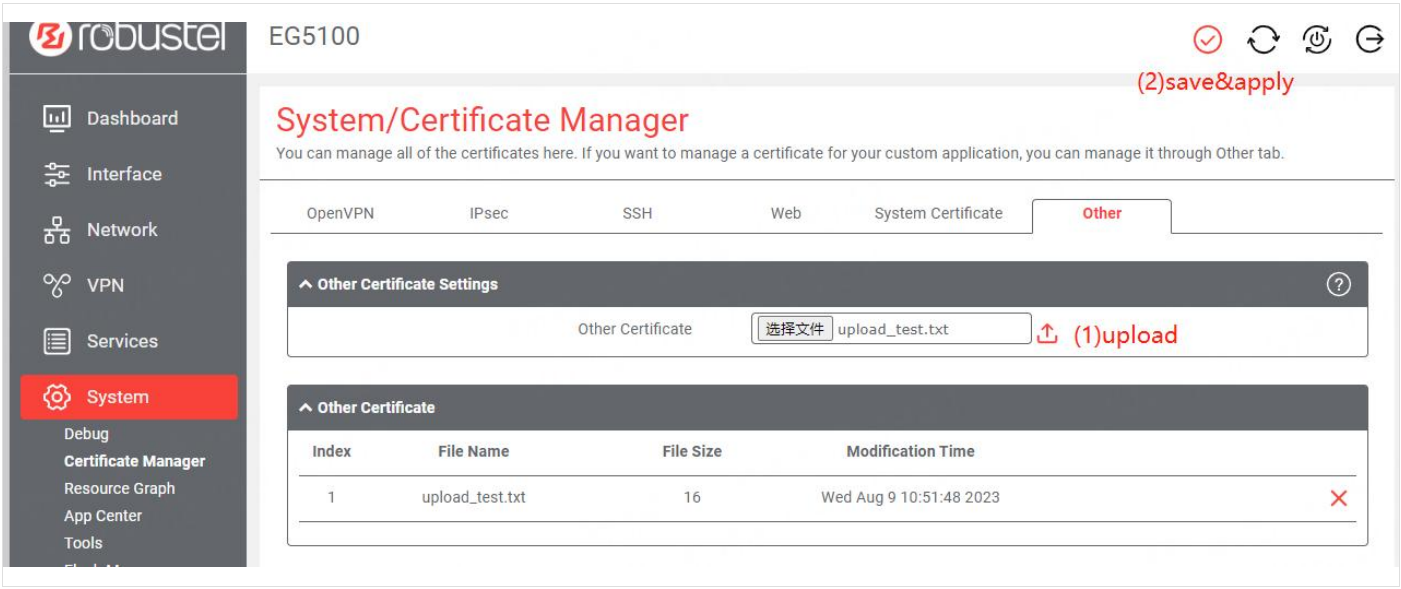
```
char *imei;
char *model;
int sim;


model = usi_get(“system.device_model”);
```

```
if (NULL != model)
{
    printf("Device Model = %s\n", model);
    free(model);
}
```

7.13 How to Upload Custom File

Method 1:

Use the reserved page (System->Certificate Manager->other) ,this page typically provide a set of functionalities that allow developers to integrate applications.



Step 1: Choose the file you want to upload, click on the upload control, and the file will be uploaded to the device's tmpfs file system path(/tmp/other/other_other_cert/).

```
root@router:~# ls /tmp/other/other_other_cert/*
/tmp/other/other_other_cert/upload_test.txt
root@router:~#
```

Step 2: Click on the save&apply control, and the file will be copied to the device's ext4 file system path (/app/config/other/).

```
root@router:~# ls /app/config/other/*
upload_test.txt
root@router:~#
```

After the device restarts, the system will copy the files from the /app/config/other/ directory to /tmp/other/other_other_cert/. Therefore, both the app and other third-party applications can access the files in the /tmp/other/other_other_cert/ directory before and after the device restarts.

Method 2:

Customizing the menu bar and pages

Step 1: Define the menu bar and add the file /etc/router/www/navigation/test01.xml.

```
<test group="Test02" group_weight="50" menu="Test03" menu_weight="30">
    <custom_tab tab="Test04" tab_weight="0" />
```

```
</test>
```

Step 2: Add a new JS file named <group>_<menu>_<tab>.js.

Add the file /www/web/js/test02_test03_test04.js.

```
var other_cert_path = "/tmp/other/";
function upload_end(flag){
    var desc;
    var bgcolor;
    if (flag)
    {
        desc = lang.submit_success;
        if (config_ischanged())
        {
            $("#id_save").removeClass("cs_not_changed").addClass("cs_is_changed");
        }
        else
        {
            $("#id_save").removeClass("cs_is_changed").addClass("cs_not_changed");
        }
        bgcolor = bottom_notification_normal_bgcolor;
    }
    else
    {
        desc = lang.operation_failure;
        bgcolor = bottom_notification_abnormal_bgcolor;

    }

    $.layer({
        shade: [0],
        title: false,
        area: ['1280', '30px'],
        maxWidth: 1004,
        closeBtn: false,
        time: 4,
        bgcolor: bgcolor,
        border: [0],
        dialog: {
            type: -1,
            msg: desc
        },
        success: function(){
            layer.shift('bottom', 400);
        }
    });

    if (!flag)
    {
```

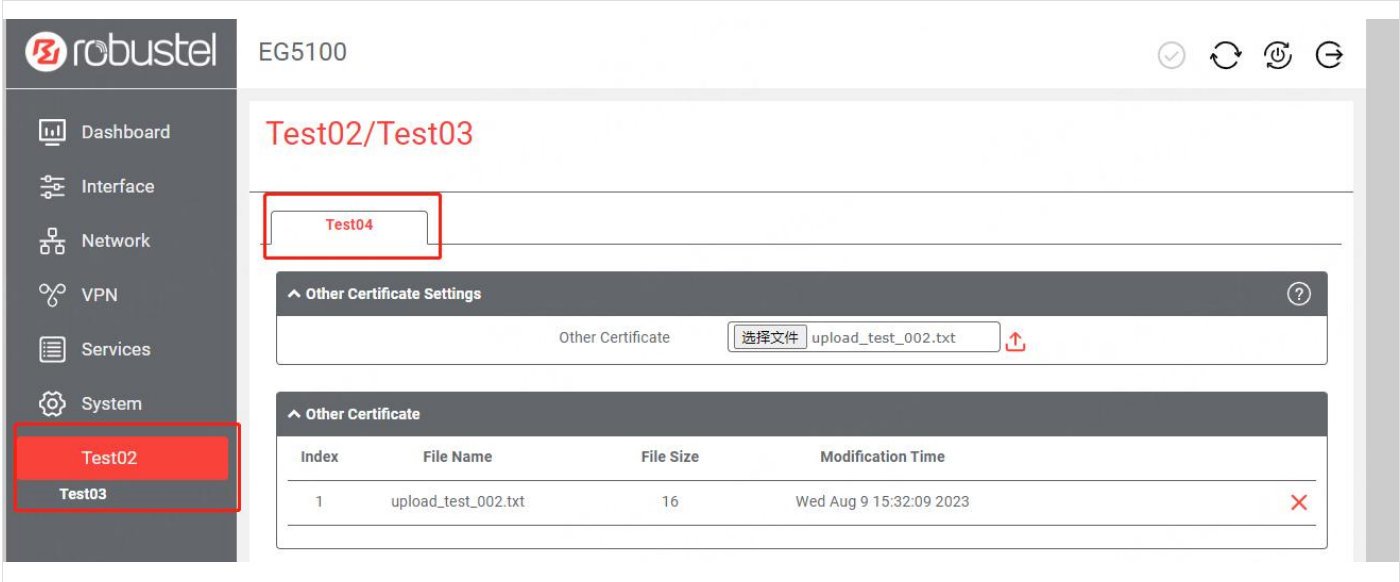
```
            $(".xubox_text").css("color", "#fff");
        }
}

function vpn_other_x509_download_item(id){
        return;
}

function vpn_other_x509_del_row(id){
        //var vpn_other_x509_cert_path = other_cert_path +"other_"+id.substring(id.indexOf("-")+1,id.indexOf("_val_"));
        var sidx = id.substring(id.indexOf("-")+1,id.indexOf("_val_"));
        var list_obj = $("#" + id);
        // var file_name = vpn_other_x509_cert_path + "/" + list_obj.children().children().eq(1).text();
        var file_name = list_obj.children().children().eq(1).text();
        $.ajax({
                type : 'POST',
                url : "/ajax/remove/",
                data : {"dir_index": 23, "dir_sub_index": sidx, "file_name" : file_name, "hash" : ck_id},
                async : true
        });
        $("#"+id).remove();

        //update config_change status
        $.ajax({
                type : 'POST',
                url : "/ajax/set_config_changed/",
                data : {"hash" : ck_id},
                async : false
        });

        if (config_ischanged())
        {
                $("#id_save").removeClass("cs_not_changed").addClass("cs_is_changed");
        }
        else
        {
                $("#id_save").removeClass("cs_is_changed").addClass("cs_not_changed");
        }
}

function get_file_list(desc){
        var path = other_cert_path + "other_"+ desc;
        console.log(path);
        $.ajax({
                type : 'POST',
                url : "/ajax/get_x509_cert_manager_files/",
                data : {"path" : path, "target": "other", "desc": desc ,"hash" : ck_id},
```

```
            data : {"dir_index" : 23, "dir_sub_index":desc, "target": "other", "desc": desc ,"hash" : ck_id},
            async : false,
            success : function(data){
                var i = 0;
                var xml_doc;
                var list_nodes;
                if (data !== "null" && data !== "" && data !== "ERROR")//data is String, not null object
                {
                    xml_doc = $.parseXML(data);
                    list_nodes = xml_doc.documentElement.childNodes;
                    for (i=0;i<list_nodes.length;i++)
                    {
                        if (list_nodes[i].nodeType === 1)
                        {
                            console.log(list_nodes[i]);
                            //parse_xml_list_node(list_nodes[i], "Download_Delete", ["vpn_other_x509_download_item",
"vpn_other_x509_del_row"], path);
                            parse_xml_list_node(list_nodes[i], "Delete_Only", ["vpn_other_x509_del_row"], path);
                        }
                    }
                }
            }
        });
}

function import_file(desc){
        var max_firmware_size = 1024*1024*1;
        var upload_files_size = 0;
        var agent = window.navigator.userAgent;
        var isIE8 = agent.indexOf('MSIE 8.0') != -1;
        var target_file = "ca.crt";
        var input_file_id = "id_" + desc    + "_val";
        if (!isIE8)
        {
            upload_files_size = document.getElementById(input_file_id).files[0].size;
        }

        //size limit verify
        if (upload_files_size > max_firmware_size)
        {
            rbt_alert(lang.normal_upload_file_too_large_desc);
            return;
        }
        var target_file_name=document.getElementById(input_file_id).files[0].name;
        var attr_desc = "/action/import_cert_file?dir_index=23&dir_sub_index="+ desc.replace(/other_/i,'')+ "&target_file=" +
target_file_name.replace('-','_');
        $("#id_" + desc + "_hash").val(ck_id);
```

```
    $("#id_" + desc + "_form").attr("action", attr_desc).submit();
}

function add_input_file(border_desc, desc)
{
    var border_id = "id_" + border_desc;
    var html_str = "<form id=\"id_" + desc + "_form\" method=\"post\" enctype=\"multipart/form-data\" target=\"upload_feedback\"> \
                        <div class=\"cs_arearow\"> \
                            <ul> \
                                <li class=\"cs_coltip\" id=\"id_" + desc + "_tip\" title=\"" + lang[ desc + "_desc" ] + "\"> \
                                    <span>" + lang[ desc + "_desc" ] + "</span> \
                                </li> \
                                <li class=\"cs_colval\"> \
                                    <span><input type=\"file\" name=\"file\" id=\"id_" + desc + "_val\"/> \
                                    <input type=\"hidden\" name=\"hash\" id=\"id_" + desc + "_hash\"></span> \
                                    <span class=\"cs_x509_upload icon-svg-file-op\" onclick=\"import_file('" + desc + "');\"> \
                                        <svg class=\"icon-svg-normal\" aria-hidden=\"true\">\
                                            <use xlink:href=\"#icon-upload\"></use>\
                                        </svg>\
                                    </span> \
                                </li> \
                            </ul> \
                        </div> \
                    </form>";
    $("#" + border_id).append(html_str);
    $("#rightbody_footer").append("<iframe id=\"id_upload_feedback\" name=\"upload_feedback\" style=\"display:none\"></iframe>");
}

var area_index=["2"];
var cert_name=["other_cert"];

function draw_certificate_area(area_desc ,item ,desc)
{
    add_area_border(area_desc, false, "false", "false", "false", "false", "false", "false");
    add_list_base(area_desc, item, desc);
    add_list_col(item, desc, "id", "60", "config", "true", "false", "false");
    add_list_col(item, desc, "name", "200", "config", "true", "false", "false");
    add_list_col(item, desc, "size", "200", "config", "true", "false", "false");
    add_list_col(item, desc, "lmtime", "200", "config", "true", "false", "false");
}

function refresh_cert_list(){
    for (i=0;i<area_index.length;i++)
    {
        $("#id_"+"cert_other_x509_area_"+area_index[i]).remove();
        draw_certificate_area("cert_other_x509_area_"+area_index[i],"other",cert_name[i]);
        get_file_list(cert_name[i]);
```

```
    }
}

$(document).ready(function(){
    $("#rightbody_content").empty();
    add_area_border("other_cert_area_0", true, "false", "false", "false", "false", "false", "false");

    for (i=0;i<cert_name.length;i++)
    {
        add_input_file("other_cert_area_0", "other_"+cert_name[i]);
    }
    refresh_cert_list();

});
```

Step 3: Restart the device or restart the service, then log in to the webpage to see the effect.



The file upload path remain consistent with the previously mentioned method 1. The application can access the files in the directory /tmp/other/other_other_cert/.

```
root@router:/www/web/js# ls /tmp/other/other_other_cert/
upload_test_002.txt
root@router:/www/web/js#
```

7.14 How to Customize Navigation Bar.

According to the description in chapters 7.3 and 7.4, users can customize the webpage navigation bar.
Example is shown as below.
Add XML file /etc/router/uci/test001.xml.

```
<test001 desc="Test002 desc" group="test002" group_weight="40" menu="test003" menu_weight="80" menu_explain="test003 menu explain." oid="1" >
    <enable type="bool" default="true" desc="Enable" tab="test004 table" tab_weight="0" area="test005 settings" oid="1" />
</test001>
```

The following webpage was generated based on the XML file.



Chapter 8 SDK C-Libraries

RobustOS Pro contains a private library named librouter, used for router functions. The librouter APIs give you programmatic access to UCI system, router environment variables, router IPC system, etc.
The header are located in the /usr/include/librouter/ directory after install librosp-dev:armhf deb. You can check the headers for detailed API description, or check the examples for instructions.

```
root@Router:~# ls /usr/include/librouter/
define.h          rlib_bn_mul.h     rlib_dir.h      rlib_list.h  rlib_random.h       rlib_string.h        rlib_user_password.h
librouter.h       rlib_can_utils.h  rlib_env.h      rlib_lock.h  rlib_ring_buffer.h  rlib_switch.h        web_server.h
librouter_open.h  rlib_cert_mgr.h   rlib_env_var.h  rlib_log.h   rlib_rsa.h          rlib_types.h
rlib_base64.h     rlib_crypto.h     rlib_errno.h    rlib_md5.h   rlib_rsa_config.h   rlib_uci.h
rlib_bignum.h     rlib_define.h     rlib_event.h    rlib_msg.h   rlib_shared.h       rlib_uci_internal.h
rlib_bits.h       rlib_des.h        rlib_file.h     rlib_net.h   rlib_shell_utils.h  rlib_uci_private.h
```

8.1 Librouter Data Type

```
typedef signed char                 int8_t;
typedef short int                   int16_t;
typedef int                         int32_t;
typedef long long int               int64_t;
typedef unsigned char               uint8_t;
typedef unsigned short int          uint16_t;
typedef unsigned int                uint32_t;
typedef unsigned long long int      uint64_t;
typedef int                         intptr_t;

typedef int                         status_t;
typedef unsigned int                bool_t;



#define TRUE                        1
#define FALSE                       0



#define ERROR                       -1
#define OK                          0

#ifndef NULL
#define NULL ((void *)0)
#endif
```

8.2 Librouter Log API

Name:

log_open(n)

Description:

log_open() is similar to openlog(), it opens a connection to the system logger for a program. The string pointed to by n is prepended to every message, and is typically set to the program name. The use of log_open() is optional; it will automatically be called by syslog() if necessary, in which case identity will default to NULL.

INPUT:

n

Output:

N/A

RETURN VALUE:

N/A

Name:

log_close()

Description:

closes the descriptor being used to write to the system logger. The use of log_close() is optional.

INPUT:

N/A

OUTPUT:

N/A

RETURN VALUE:

N/A

Name:

log_debug(fmt, ...)

Description:

Debug message log to syslog.

INPUT:

Message to log.

OUTPUT:

Output to syslog

RETURN VALUE:

N/A

Name:

log_info(fmt, ...)

Description:

Info message log to syslog.

INPUT:

Message to log.

OUTPUT:

Output to syslog

RETURN VALUE:

N/A

Name:

log_notice(fmt, ...)

Description:

Notice message log to syslog.

INPUT:

Message to log.

OUTPUT:

Output to syslog

RETURN VALUE:

N/A

Name:

log_warn(fmt, ...)

Description:

Warning message log to syslog.

INPUT:

Message to log.

OUTPUT:

Output to syslog

RETURN VALUE:

N/A

Name:

log_error(fmt, ...)

Description:

Error message log to syslog.

INPUT:

Message to log.

OUTPUT:

Output to syslog

RETURN VALUE:

N/A

8.3 Librouter UCI API

Name:

status_t uci_init(void)

Description:

If you want to modify the configuration file, you should call this init function firstly. There is no need to call uci_init if you just want to read configuration.

INPUT:

N/A

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

status_t uci_close(void)

Description:

User had better call this before exit if you had called uci_init.

INPUT:

N/A

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

const char *uci_get(const char *tag)

Description:

Get configuration by specified tag, it does not need to calling free() after get.

INPUT:

tag - tag with UCI format, eg: lan.ip means ip of lan.

OUTPUT:

N/A

RETURN VALUE:

The pointer of value, or NULL if an error occurred.

Name:

int32_t uci_get_int(const char *tag)

Description:

Get integer type configuration data.

INPUT:

tag - tag with UCI format

OUTPUT:

N/A

RETURN VALUE:

The integer value, or 0 if an error occurred.

Name:

bool_t uci_match(const char *tag, const char *value)

Description:

Check configuration data is matched or not.

INPUT:

tag - tag with UCI format

value - the value to compare

OUTPUT:

N/A

RETURN VALUE:

TRUE if matched, otherwise return FALSE.

Name:

bool_t uci_check_bool(const char *tag)

Description:

Check bool type configuration data is TRUE or FALSE.

INPUT:

tag - tag with UCI format

OUTPUT:

N/A

RETURN VALUE:

TRUE if the value is "true", otherwise return FALSE.

Name:

status_t uci_set(const char *tag, const char *value)

Description:

Set specified configuration item to specified value.

INPUT:

tag - tag with UCI format

value - string type value

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

int32_t uci_list_get_max_entry_num(const char *tag)

Description:

Get the maximum entry number of list, use to traversal the list.

INPUT:

tag - tag with UCI format

value - string type value

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

bool_t uci_list_entry_empty(const char *tag, uint32_t id)

Description:

Check list entry is empty or not.

INPUT:

tag - tag with UCI format

id - list ID

OUTPUT:

N/A

RETURN VALUE:

TRUE if the list entry is empty, otherwise return FALSE.

Name:

status_t uci_list_add(const char *tag, uint32_t id)

Description:

Setup the new list, you must to add a new entry to the list before.

INPUT:

tag - tag with UCI format

id - list ID

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

status_t uci_list_del(const char *tag, uint32_t id)

Description:

Delete a list entry with specified ID.

INPUT:

tag - tag with UCI format

id - list IDto delete

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

const char *uci_list_get(const char *list_tag, uint32_t id, const char *member_name)

Description:

Get the value of specified list member.

INPUT:

list_tag - tag with UCI format

id - list ID

member_name - member name of list

OUTPUT:

N/A

RETURN VALUE:

The pointer of value, or NULL if an error occurred.

Name:

status_t uci_list_set(const char *list_tag, uint32_t id, const char *member_name, const char *value)

Description:

Set the value of specified list member.

INPUT:

list_tag - tag with UCI format

id - list ID

member_name - member name of list

value - string type value

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

bool_t uci_list_match(const char *list_tag, uint32_t id, const char *member_name, const char *value)

Description:

Set the value of specified list member.

INPUT:

tag - tag with UCI format

id - list ID

member_name - member name of list

value - value to compare

OUTPUT:

N/A

RETURN VALUE:

TRUE if matched, otherwise return FALSE.

Name:

char *usi_get(const char *tag)

Description:

Get status by tag, it's up to the caller to free the memory.

INPUT:

tag - tag with USI format, be the same as UCI format

OUTPUT:

N/A

RETURN VALUE:

The pointer of value, or NULL if an error occurred.

Name:

bool_t usi_match(const char *tag, const char *value)

Description:

check status data is matched or not.

INPUT:

tag - tag with USI format

value - the value to compare

OUTPUT:

N/A

RETURN VALUE:

TRUE if matched, otherwise return FALSE.

Name:

status_t usi_update(const char *tag, const char *status)

Description:

Update specified status item to specified value, if tag is not define it will be created.

INPUT:

tag - tag with USI format

status- string type value

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

status_t usi_printf(const char *tag, const char *fmt, ...)

Description:

printf-style function to update status.

INPUT:

tag - tag with USI format

fmt - printf-style format

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

status_t usi_clear(const char *tag)

Description:

clear a status file, or clear all items of specified list, or clear an item.

INPUT:

tag - tag with USI format

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

status_t usi_list_set(const char *list_tag, uint32_t id, const char *member_name, const char *value)

Description:

update specified list member to specified value.

INPUT:

list_tag - tag with UCI format

id - list ID

member_name - list member

value - string type value

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

status_t usi_list_printf(const char *list_tag, uint32_t id, const char *member_name, const char *fmt, ...)

Description:

Update specified list member to specified value.

INPUT:

list_tag - tag with UCI format

id - list ID

member_name - list member

fmt - printf-style format

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

char *usi_list_get(const char *list_tag, uint32_t id, const char *member_name)

Description:

Get status of specified list member, it's up to the caller to free the memory.

INPUT:

list_tag - tag with UCI format

id - list ID

member_name - list member

OUTPUT:

N/A

RETURN VALUE:

The pointer of value, or NULL if an error occurred.

Name:

status_t usi_list_delete(const char *list_tag, uint32_t id)

Description:

Delete a list entry with specified ID.

INPUT:

list_tag - tag with UCI format

id - list ID to delete

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

bool_t usi_list_match(const char *list_tag, uint32_t id, const char *member_name, const char *value)

Description:

Check the status value of specified list member is matched or not.

INPUT:

list_tag - tag with UCI format

id - list ID

member_name - member name of list

value - value to compare

OUTPUT:

N/A

RETURN VALUE:

TRUE if matched, otherwise return FALSE.

Name:

int32_t usi_list_get_max_entry_num(const char *tag)

Description:

Get the maximum entry number of list, use to traversal the list.

INPUT:

list_tag - tag with UCI format

OUTPUT:

N/A

RETURN VALUE:

maximum entry number of list, 0 if an error occurred.

Name:

bool_t usi_list_entry_empty(const char *tag, uint32_t id)

Description:

Check list entry is empty or not.

INPUT:

list_tag - tag with UCI format

id - list ID

OUTPUT:

N/A

RETURN VALUE:

TRUE if the list entry is empty, otherwise return FALSE

8.4 Librouter Environment Variable API

Name:

status_t renv_set(const char *name, const char *value)

Description:

Add or change a name/value pair to router environment variable

INPUT:

name - name of environment variable

value - value of environment variable

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

status_t renv_printf(const char *name, const char *fmt, ...)

Description:

printf-style function to set environment variable

INPUT:

name - name of environmentvariable

fmt - printf-style format

OUTPUT:

N/A

RETURN VALUE:

OK means successful, or ERROR if an error occurred.

Name:

const char *renv_get(const char *name)

Description:

Get environment variable by name, it does not need to calling free() after get.

INPUT:

name - name of environment variable

OUTPUT:

N/A

RETURN VALUE:

The pointer of value, or NULL if an error occurred.

Name:

int32_t renv_get_int(const char *name)

Description:

Get integer type environment variable by name.

INPUT:

name - name of environment variable

OUTPUT:

N/A

RETURN VALUE:

The integer value, or 0 if an error occurred.

Name:

uint32_t renv_get_uint(const char *name)

Description:

Get unsigned integer type environment variable by name.

INPUT:

name - name of environment variable

OUTPUT:

N/A

RETURN VALUE:

The integer value, or 0 if an error occurred.

Name:

status_t renv_del(const char *name)

Description:

Delete environment variable by name.

INPUT:

name - name of environment variable

OUTPUT:

N/A

RETURN VALUE:

OK if success, or ERROR if an error occurred.

Name:

bool_t renv_match(const char *name, const char *value)

Description:

Check environment variable is matched or not.

INPUT:

name - name of environment variable

value– the value to compare

OUTPUT:

N/A

RETURN VALUE:

TRUE if matched, otherwise return FALSE.

8.5 Librouter IPC API

Name:

int rmsg_init(const char *sock_name)

Description:

For IPC you can open a UNIX Domain Socket for message communication. You should call rmsg_init before rmsg_send.

INPUT:

sock_name - socket name, without path

OUTPUT:

N/A

RETURN VALUE:

Socket descriptor if success, or -1 if an error occurred.

Name:

status_t rmsg_close(void)

Description:

You had better call rmsg_close if you do not want to send any more messages.

INPUT:

N/A

OUTPUT:

N/A

RETURN VALUE:

OK if success, or ERROR if an error occurred.

Name:

status_t rmsg_send(dst, msgs, ...)

Description:

Send message parts to destination.

INPUT:

dst - destination socket name

msgs - key-value style message pairs

OUTPUT:

N/A

RETURN VALUE:

OK if success, or ERROR if an error occurred.

Name:

status_t rmsg_recv(char **msg, char **from)

Description:

Receive message.

INPUT:

msg - pointer to the message buffer

from - pointer to the sender

OUTPUT:

N/A

RETURN VALUE:

OK if success, or ERROR if an error occurred.

Name:

char *rmsg_get(const char *key)

Description:

Get the corresponding value of specified key, it's up to the caller to free the memory.

INPUT:

key - key of message item

OUTPUT:

N/A

RETURN VALUE:

Pointer to the value if success, NULL if an error occurred

8.6 Librouter Event API

Name:

status_t event_subscribe(const char *event)

Description:

Send a message to event service to subscribe an event, rmsg_init should be called firstly.

INPUT:

event - event name

OUTPUT:

N/A

RETURN VALUE:

OK if success, or ERROR if an error occurred.

Name:

status_t event_unsubscribe_all(void)

Description:

unsubscribe all events, user had better call this before exit if you had called event_subscribe.

INPUT:

N/A

OUTPUT:

N/A

RETURN VALUE:

OK if success, or ERROR if an error occurred.

Name:

status_t event_report(const char *event, const char *data)

Description:

send a message to event service to report an event, rmsg_init should be called firstly.

INPUT:

event - event name

data - extra data of the event

OUTPUT:

N/A

RETURN VALUE:

OK if success, or ERROR if an error occurred.

Name:

status_t event_report_with_formatted_data(const char *event, const char *fmt, ...)

Description:

printf-style's event_report

INPUT:

event - event name

fmt - printf-style format

OUTPUT:

N/A

RETURN VALUE:

OK if success, or ERROR if an error occurred.

Name:

status_t event_report_full(const char *from, const char *event, const char *data)

Description:

printf-style's event_report

INPUT:

from - same as socket name in rmsg_init

event - event name

data - extra data of the event

OUTPUT:

N/A

RETURN VALUE:

OK if success, or ERROR if an error occurred.

Name:

status_t event_report_full_with_formatted_data(const char *from, const char *event, const char *fmt, ...)

Description:

printf-style's event_report_full

INPUT:

from - same as socket name in rmsg_init

event - event name

fmt - printf-style format

OUTPUT:

N/A

RETURN VALUE:

OK if success, or ERROR if an error occurred.

Chapter 9 Pre-installed Tools

9.1 RobustOS Pro Private Commands

9.1.1 uci

```
$ uci -h
Usage: uci [<options>] <command> [<arguments>]

Commands:
        get             <tag> (get config by tag, e.g. get lan.ip)
        set             <tag> <value> (update config, e.g. set openvpn.tunnel[1].enable true)
        add             <tag> <id> (add a new list entry, e.g. add openvpn.tunnel 1)
        del             <tag> <id> (delete a list entry)
        load            (load config from xml)
        default     (generate default config)
        save            (save config changes)
        apply       (apply config changes)
        commit      (save and apply config changes)
        export      [<xmlfile>] (export config to xml, default to /tmp/config.xml)
        import      <xmlfile> (import config from xml)
        check       <category> (check config has changed or not)
        refresh     [<category>] (reload config from file to RAM)
        print       (show all uciconfig in the RAM)
Options:
                -c          when importing, change other config to default value
                -d          when exporting, output detailed information
                -s          when exporting, ignore some disabled feature to get a simpler xml file
                -h          display this help and exit
```

9.1.2 usi

```
$ usi
Usage: usi<command> [<arguments>]

Commands:
        get <tag> (get status by tag, e.g. get system.firmware_version)
        set <tag> <value> (update status, e.g. set cellular.imei 123456789012345)
        clear <tag> (clear status, e.g. clear openvpn.tunnel_status)
        sync <category> (sync status to unified format)
```

9.1.3 renv

```
$ renv –h
Usage: renv [<command>] [<arguments>]

Print the current router environment if no command specified.
Commands:
        get<name>
        set<name><value>
        del<name>
```

9.1.4 sms

```
$ sms -h
sms: unknown option -h

usage: sms [options...] <phone_number> <sms_content>

Options:
            -s    use the second modem
```

Chapter 10 Programming Tutorials

10.1 How to Debug the Application

10.1.1 Use Syslog

The RobustOS Pro supports debugging via syslog. You can check syslog in the Web Manager, which locates in System->Debug.

10.1.2 Use Shell with Root Privileges

A more convenient way is debugging with shell. You can check syslog using “journalctl/journalctl -f” with root privileges.

10.2 How to Get System Status

Please refer to Chapter 7

10.3 How to Read/Write Data via Serial Port

Serial port programming in RobustOS Pro is the same as in common Linux systems. The only thing you should notice is the device file(/dev/ttyCOM1 /dev/ttyCOM1) of serial port, which has been listed in Chapter 3.
You can configure the serial port properties through the code api and then read or write the device files directly.
Set the serial port to RS232 or RS485:

```
#### Setting ttyCOM1 to work in RS232 mode
uci set serial_port.port_setting[1].enable false
uci set serial_port.port_setting[1].port_type RS232
uci commit
  #### Setting ttyCOM1 to work in RS485 mode
uci set serial_port.port_setting[1].enable false
uci set serial_port.port_setting[1].port_type RS485
uci commit
#### Setting ttyCOM2 to work in RS232 mode
uci set serial_port.port_setting[2].enable false
uci set serial_port.port_setting[2].port_type RS232
uci commit
#### Setting ttyCOM2 to work in RS485 mode
uci set serial_port.port_setting[2].enable false
uci set serial_port.port_setting[2].port_type RS485
uci commit
```

Open the serial port and set the serial port properties:

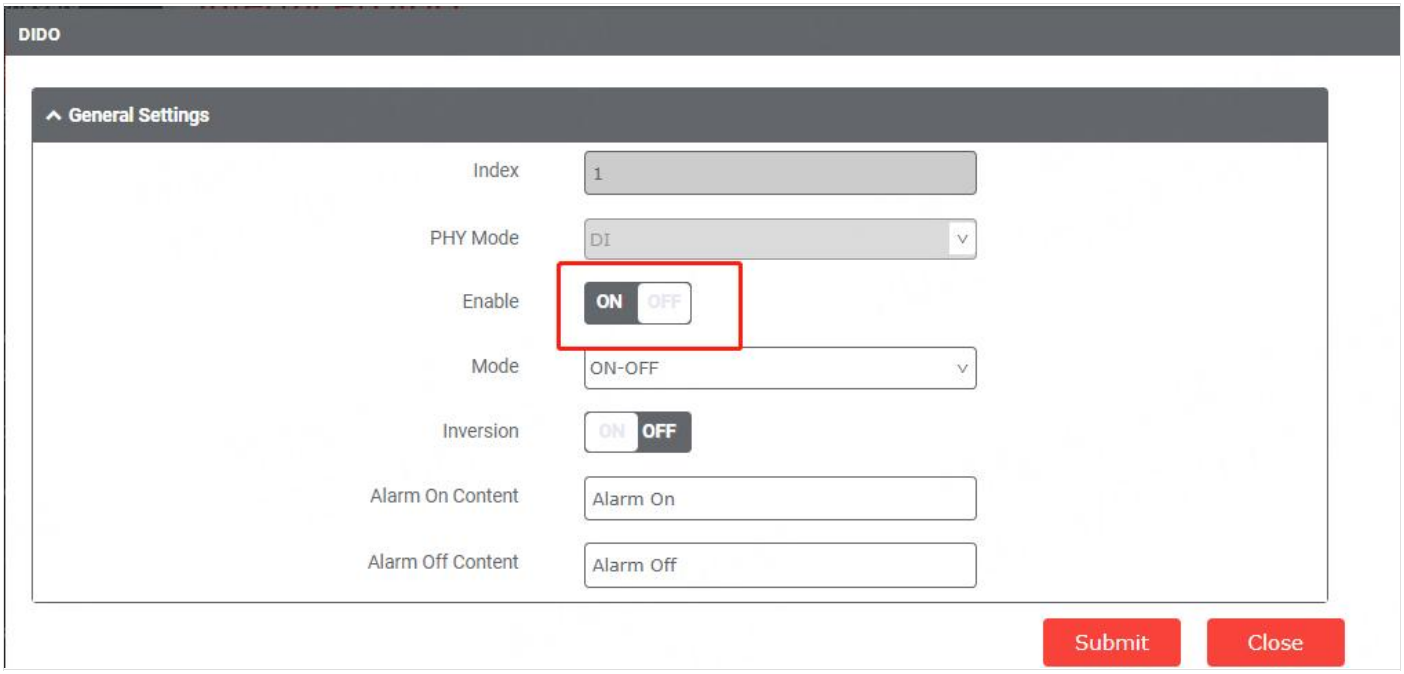
```
Name:
    int serial_port_open_full(char *tty_device, uint baud_rate, uint data_bits, uint stop_bits, uint parity, uint flow_ctrl)
Description:
    open the serial port and set the serial port properties
INPUT:
    tty_device - device file path (/dev/ttyCOM1 /dev/ttyCOM1)
    baud_rate    - (300 600 1200 2400 4800 9600 19200 38400 57600 115200)
    data_bits    - (7 8)
    stop_bits    - (1 2)
    parity         -    none odd even(0 1 2)
    flow_ctrl    -    none hardware software(0 1 2)
OUTPUT:
    N/A
RETURN VALUE:
    file descriptor if success, or -1 if an error occurred.
```

Code example is shown as below:

```
// Setting ttyCOM1 to work in RS232 mode
system("uci set serial_port.port_setting[1].enable false");
system("uci set serial_port.port_setting[1].port_type RS232");
system("uci commit");

// Setting ttyCOM1 and open it
com_fd = serial_port_open_full("/dev/ttyCOM1", baud_rate, data_bits, stop_bits, parity, flow_ctrl);

// Reading ttyCOM1
data_len = read(com_fd, recv_buff, sizeof(recv_buff));

// Writing ttyCOM1
write(com_fd, recv_buff, data_len);

// Closing ttyCOM1
close(com_fd);
```

10.4 How to Get DI Status

In order to get DI status, DI functionality needs to be enabled first.



Alternatively, DI functionality can be enabled using the UCI command.

```
uci set dido.io[1].enable true
uci commit
```

Application just need to subscribe DI ON/OFF event.

Example is shown as below.

```
#include "librouter.h"
#include "rlib_shell_utils.h"
#include "rlib_define.h"
#include "rlib_shared.h"
#include "rlib_string.h"
#include "event.h"

void di_event_msg_proc(int sock, short ev, void *arg)
{
    char *msg;
    char *from;
    char *cmd = NULL;
    char *event = NULL;

    if (OK != rmsg_recv(&msg, &from))
    {
        log_err_location();
        return;
    }
    log_debug("recv msg %s from %s", msg, from);
```

```
cmd = rmsg_get("cmd");

	if (!cmd)
	{
		log_err_location();
		goto END;
	}
	if (string_matched(cmd, "report"))
	{
		event = rmsg_get("event");
		if (!event)
		{
			log_err_location();
			goto END;
		}
		if (string_matched(event, "di1_on"))
		{
			log_debug("DI1 on");
		}
		else if(string_matched(event, "di1_off"))
		{
			log_debug("DI1 off");
		}
	}

END:
	safe_free(cmd);
	safe_free(event);
}

int main(int argc, char *argv[])
{

	int sd= -1;
	struct event event;

	daemon(0, 0);

	if(daemon_check_running("/var/run/sample_dido.pid"))
	{
		log_warn("sdk sample_dido already running!");
		return 0;
	}
	if(!uci_check_bool("devcfg.dido_support"))
	{
		log_error("Don't Support DIDO, sample_dido exit!");
		return -1;
```

```
    }

    log_notice("sdk sample_dido service started");

    sd = rmsg_init(argv[0]);
    event_init();

    event_set(&event, sd, EV_READ|EV_PERSIST, di_event_msg_proc, NULL);
    event_add(&event, NULL);

    event_subscribe("di1_on");
    event_subscribe("di1_off");
    event_dispatch();

    return 0;
}
```

10.5 How to Trigger DO Changes

RobustOS Pro uses the linux led subsystem for DO processing and control the relay using a script.

```
##### For EG5100/LG5100/EG5120 DO control
$ echo 0 > /sys/class/leds/do1/brightness
$ echo 1 > /sys/class/leds/do1/brightness

#### For EG5200/EV8100 Relay control
/usr/scripts/relay_control.sh type=relay    num=relay1 mode=on
/usr/scripts/relay_control.sh type=relay    num=relay1 mode=off
```

10.6 How to perform BLE scan

```
#####For EV8100
$ hciattach /dev/ttymxc1 any 115200 flow 2
$ hciconfig hci0 up
$ hciconfig
$ hcitool -i hci0 scan

#####For EG5100/EG5120
$ hciattach /dev/ttymxc3 any 115200 flow 2
$ hciconfig hci0 up
$ hciconfig
$ hcitool -i hci0 scan

#####For EG5200
$ echo 0 > /sys/class/gpio/gpio1/value      #power off BLE
$ echo 1 > /sys/class/gpio/gpio1/value      #power on BLE
```

```
$ hciattach /dev/ttymxc3 qca -t120 3000000 flow
$ hciconfig hci0 up
$ bluetooth scan on
```

10.7 How to Send Data via CAN Interface

```
#Check the operational state
$ cat /sys/class/net/can0/operstate

  #Bring the interface up and set the CAN bitrate
$ ip link set can0 up type can bitrate 800000

# Send a message with ID 138 and data bytes 0102030405060788
$ cansend can0 138#0102030405060788

# Monitor CAN Messages
$ candump can0
```

10.8 How to Mount SD Card Partition

```
#Create a Mount Point:
$ mkdir -p /mnt/sdcard

  #Mount the Partition
$ mount /dev/mmcblk0p1 /mnt/sdcard

# Verify the Mount
$ df -h | grep /mnt/sdcard

# Unmounting an SD Card Partition
$ umount /mnt/sdcard
```

10.9 How to Get AI Status

There are two ways to get AI status.

- Through AI device file

AI value can be read from AI device file /dev/AI1. It is used to get AI precision values.

- Subscribe AI events

If applications do not care about the precision values, but whether it reaches voltage/current threshold, then applications can subscribe AI events, listed below. Please take reference to how to get DI status.

10.10 How to Subscribe Event

10.10.1 Supported Events

- system_startup - system has been started
- system_reboot - system is going to reboot
- system_time_updated - system time has been updated
- system_login_success - someone has been logged in the system successfully
- system_login_fail - someone failed to login the system
- config_changed - configuration has been changed
- cellular_data_stats_clear - cellular data stats has been cleared
- cellular_data_traffic_overflow – cellular data traffic reach threshold
- cellular_network_type_changed – cellular network type has been changed.
- poor_signal_quality - signal quality is bad
- link_switch - system is going to switch link
- wan_up - data connection of WAN has been up
- wan_down - data connection of WAN has been down
- wlan_up - data connection of WLAN has been up
- wlan_down - data connection of WLAN has been down
- wwan_up - data connection of WWAN has been up
- wwan_down - data connection of WWAN has been down
- ipsec_up - IPSec tunnel up
- ipsec_down - IPSec tunnel down
- openvpn_up - OpenVPN tunnel up
- openvpn_down - OpenVPNtunnel down
- robustvpn_up - RobustVPN tunnel up
- robustvpn_down - RobustVPN tunnel down
- gps_up - GPS fixed
- gps_down - GPS fix fail
- lan_port_up - a certain LAN port link up
- lan_port_down - a certain LAN port link down
- usb_connected – a USB device was inserted
- usb_removed – usb device was removed
- ddns_update_success - DDNS has been updated successfully
- ddns_update_fail - failed to update DDNS
- sms_received – receive a new short message
- sms_sent – short message was sent
- sms_cmd_executed – command in short message has been executed
- di1_on – di1 has been enabled
- di1_off - di1 has been disable
- di1_counter_overflow – di1 counter reach threshold
- di2_on – di1 has been enabled
- di2_off - di1 has been disable
- di2_counter_overflow – di1 counter reach threshold
- di3_on – di1 has been enabled
- di3_off - di1 has been disable

- di3_counter_overflow – di1 counter reach threshold
- di4_on – di1 has been enabled
- di4_off - di1 has been disable
- di4_counter_overflow – di1 counter reach threshold
- ai1_voltage_low – voltage reaches minimum threshold
- ai1_voltage_high – voltage reaches maximum threshold
- ai1_current_low – current reaches minimum threshold
- ai1_current_high – current reaches maximum threshold
- ai1_recovery – voltage or current returns to normal

10.10.2 Code Example

```
#include “librouter.h”
#include “event.h”      //header file of libevent

void event_msg_proc(int sock, short ev, void *arg)
{
        char *msg;
        char *from;
        char *cmd = NULL;
        char *event = NULL;
        char *desc = NULL;
        char *date = NULL;

        if (OK != rmsg_recv(&msg, &from))
        {
                return;
        }

        cmd = rmsg_get("cmd");
        if (NULL == cmd
                || strcmp(cmd, “report”))
        {
                return;
        }
        event = rmsg_get("event");  //you should check the pointer for real project
        desc = rmsg_get("desc");
        date = rmsg_get("date");

        // add some code for event handling
        log_info(“receive event %s, desc = %s, date = %s”, event, desc, date);

        free(cmd);  //you should check the pointer for real project
        free(event);
        free(desc);
        free(date);
}
```

```
int main(int argc, char **argv)
{
        int sd;
        struct event event;

        daemon(0, 0);

        sd = rmsg_init(argv[0]);
        event_init();
        event_unsubscribe_all();

        event_set(&event, sd, EV_READ|EV_PERSIST, event_msg_proc, NULL);
        event_add(&event, NULL);

        event_subscribe("wwan_up");
        event_subscribe("config_changed");

        event_dispatch();

        return 0;
}
```

10.11 How to Get new SMS

When ROS pro received an SMS from modem, this SMS is transported to evend which will send this "sms_received" event with sms's sender and content to all its subscribers. At this point, if an APP needs to receive new SMS, it should subscribe the "sms_received" from eventd, like the example in section 10.8.2 with event_subscribe("sms_received") call.

10.12 How to Send SMS

```
// Send SMS "hello world" to phone number 123456789:
system("sms 123456789 \"hello world\"");
```

10.13 How to Send Email

User can use email for notification. Email function is implemented by emaild daemon process, and it can handle the send email request from other process. For user application, just send IPC message to emaild process with the target mail address and the content.

Please note that, user must enable and configure email function firstly (menu path: Service->Email).

```
// Send email to test@gmail.com
char email_addresses[]="test@gmail.com";
char buf[]="from router";
rmsg_send("emaild", "cmd", "send", "addresses", email_addresses, "content", buf);
```

10.14 How to Update DEB Packages

```
// download DEB packages, use curl tools or other methods
// add your code to DEB packages
// use sysupgrade command for DEB updating
system("sysupgrade /path/to/XXX.deb");
```

10.15 How to Import/Export XML Configuration File

```
// download XML configuration file to router
// add your code to download configuration file

// import configuration file, then save and apply
system("uci import /path/to/xmlfile");
system("uci save");
system("uci apply");

// export configuration to a XML file
system("uci export /path/to/xmlfile");

// upload the XML file, use curl/tftp command or other methods
// add your code to upload configuration file
```

10.16 Systemd

If developer do not need WEB GUI or CLI or SMS to manage the app, then you can develop app based on systemd. A sample systemd configuration file is as follows. The file should be copied to /lib/systemd/system/ path.

```
[Unit]
Description=hello
Documentation=man:hello(8)

[Service]
Type=simple
Restart=always
ExecStart=/usr/bin/hello
ExecStop=/bin/kill -SIGTERM $MAINPID

[Install]
WantedBy=multi-user.target
```

Then enable to start the service automatically when boot：

```
$ systemctl enable hello.service
```

Start the service manually:

```
$ systemctl start hello.service
```

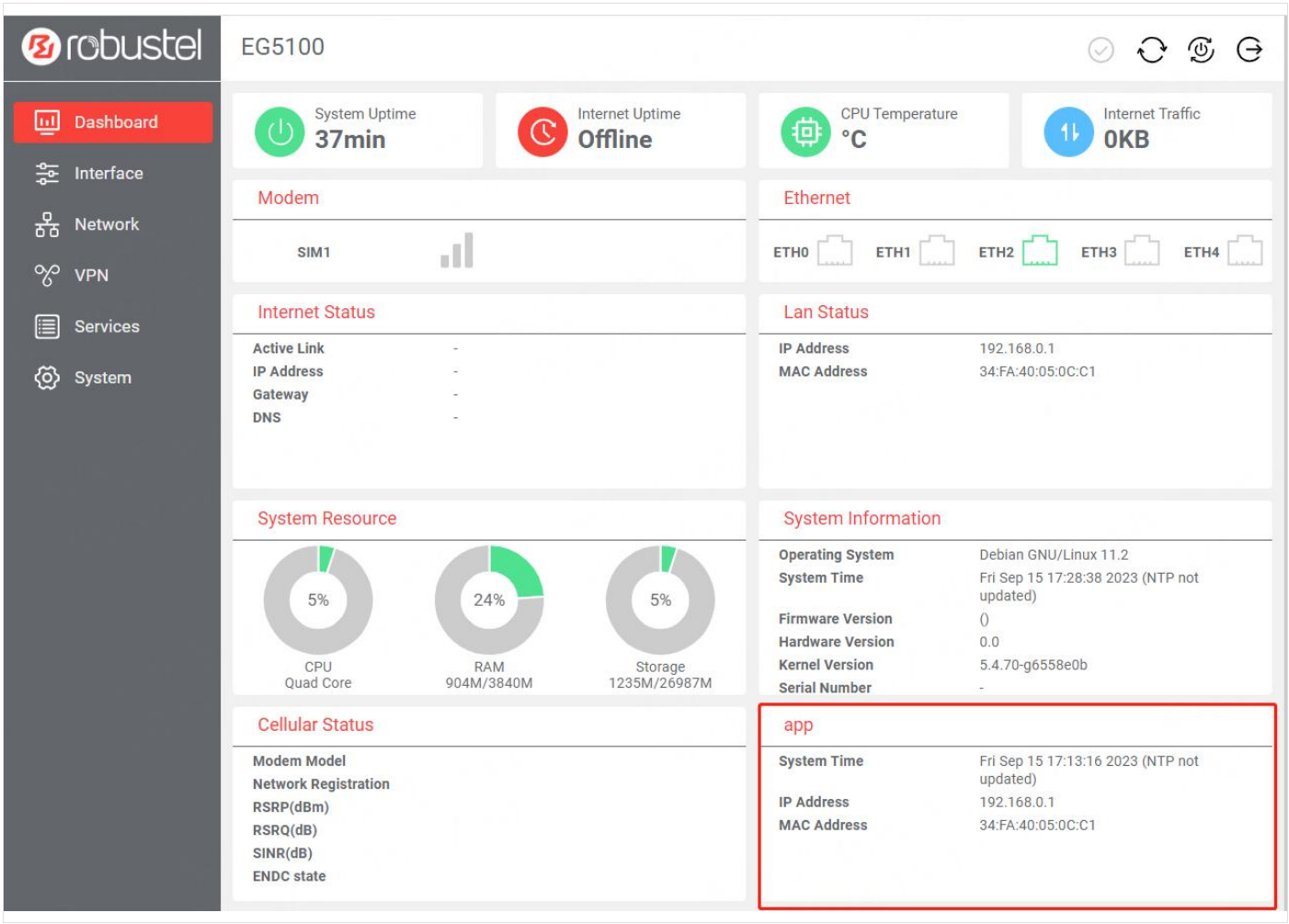
10.17 How to customize the dashboard area

Reserve application-defined content in the web dashboard area.
1) When the app is installed, copy the customized dashboard configuration file to /etc/router/dashboard/
2) The customized configurations were merged and processed during router restart
3) The web page displays custom dashboard content according to the configuration

File example：

```
<sdk_dashboard title="app" priority="100">
    <app desc="APP">
      <system_time desc="System Time"/>
      <link_status>
        <id>1</id>
        <ip4_addr desc="IP Address" />
        <mac desc="MAC Address" />
      </link_status>i
    </app>
</sdk_dashboard>
```

Content with a higher priority will be placed in a higher area. If there is no priority attribute or the dashboard area has been used up, the content will not be displayed. sdk_dashboard is a fixed node. The following sub-nodes map to the full system usi path to be obtained.



10.18 How to Query Installed Services

To query installed services on system, you can use several commands to get information about both active and inactive services. Here are the primary methods to do so.

10.18.1 Using systemctl

```
#List All Installed Services
systemctl list-unit-files --type=service

#List All Active (Running) Services
systemctl list-units --type=service --state=running

#List All Services (Active and Inactive)
systemctl list-units --type=service --all
```

Example Output

```
root@router:~# systemctl list-units --type=service --state=running
  UNIT                          LOAD   ACTIVE SUB     DESCRIPTION
  dbus.service                  loaded active running D-Bus System Message Bus
  exim4.service                 loaded active running LSB: exim Mail Transport Agent
  getty@tty1.service            loaded active running Getty on tty1
  NetworkManager.service        loaded active running Network Manager
  nginx.service                 loaded active running A high performance web server and a reverse proxy server
  rng-tools-debian.service      loaded active running LSB: rng-tools (Debian variant)
  router.service                loaded active running keeper for ros_pro
  serial-getty@ttymxc1.service loaded active running Serial Getty on ttymxc1
  snmpd.service                 loaded active running Simple Network Management Protocol (SNMP) Daemon.
  ssh.service                   loaded active running OpenBSD Secure Shell server
  systemd-journald.service      loaded active running Journal Service
  systemd-logind.service        loaded active running User Login Management
  systemd-udevd.service         loaded active running Rule-based Manager for Device Events and Files
  watchdog-ros.service          loaded active running "Hardware watchdog"
  wpa_supplicant.service        loaded active running WPA supplicant
  xl2tpd.service                loaded active running LSB: layer 2 tunelling protocol daemon
```

10.18.2 Using service

#To list all services managed by the init system

service --status-all

Example Output

[+]: The service is running.

[-]: The service is stopped.

[?]: The status of the service is unknown or could not be determined.

```
root@router:~# service --status-all
 [ - ]  bluetooth
 [ + ]  dbus
 [ - ]  dnsmasq
 [ + ]  exim4
 [ - ]  hostapd
 [ - ]  hwclock.sh
 [ + ]  inadyn
 [ + ]  kmod
 [ + ]  nginx
```

10.18.3 Using dpkg

#List All Installed Packages

dpkg -l

#Filter for Service-related Packages

dpkg -l | grep -E 'systemd|service|daemon'

Example Output

```
root@router:~# dpkg -l
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                                 Version                            Architecture Description
+++-====================================-==================================-============-====================================
ii  adduser                              3.118                              all          add and remove users and groups
ii  alsa-topology-conf                   1.2.4-1                            all          ALSA topology configuration files
ii  alsa-ucm-conf                        1.2.4-2                            all          ALSA Use Case Manager configuration
ii  apt                                  2.2.4                              arm64        commandline package manager
ii  autoconf                             2.69-14                            all          automatic configure script builder
ii  autoconf-archive                     20190106-2.1                       all          Autoconf Macro Archive
ii  automake                             1:1.16.3-2                         all          Tool for generating GNU Standards-co
ii  autopoint                            0.21-4                             all          tool for setting up gettext infrastr
ii  autotools-dev                        20180224.1+nmu1                    all          Update infrastructure for config.{gu
ii  base-files                           11.1+deb11u2                       arm64        Debian base system miscellaneous fil
ii  base-passwd                          3.5.51                             arm64        Debian base system master password a
ii  bash                                 5.1-2+b3                           arm64        GNU Bourne Again SHell
```

Chapter 11 Docker Support

Depending on user's needs, user can install Docker in different ways. Here we introduces one quick & easy method for setting up Docker environment.

Use the following commands to install docker Engine:

1.Update the apt package index.

$apt-get update

```
root@Router:~# apt update
Hit:1 http://deb.debian.org/debian bullseye InRelease
Hit:2 http://deb.debian.org/debian-security bullseye-security InRelease
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [39.4 kB]
Fetched 39.4 kB in 2s (25.9 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
9 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

2.Download the get-docker.sh script and execute it.

Note: Please keep the network connected, as it may take some minutes.

curl -fsSL get.docker.com -o get-docker.sh

sh get-docker.sh --mirror Aliyun

```
root@Router:~# curl -fsSL get.docker.com -o get-docker.sh
root@Router:~# ls get-docker.sh
get-docker.sh
root@Router:~# sh get-docker.sh --mirror Aliyun
# Executing docker install script, commit: b2e29ef7a9a89840d2333637f7d1900a83e7153f
+ sh -c 'apt-get update -qq >/dev/null'
+ sh -c 'DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificates curl >/dev/null'
debconf: delaying package configuration, since apt-utils is not installed
+ sh -c 'mkdir -p /etc/apt/keyrings && chmod -R 0755 /etc/apt/keyrings'
+ sh -c 'curl -fsSL "https://mirrors.aliyun.com/docker-ce/linux/debian/gpg" | gpg --dearmor --yes -o /etc/apt/keyrings/docker.g
pg'
+ sh -c 'chmod a+r /etc/apt/keyrings/docker.gpg'
+ sh -c 'echo "deb [arch=arm64 signed-by=/etc/apt/keyrings/docker.gpg] https://mirrors.aliyun.com/docker-ce/linux/debian bullse
ye stable" > /etc/apt/sources.list.d/docker.list'
+ sh -c 'apt-get update -qq >/dev/null'
+ sh -c 'DEBIAN_FRONTEND=noninteractive apt-get install -y -qq --no-install-recommends docker-ce docker-ce-cli containerd.io do
cker-compose-plugin >/dev/null'
debconf: delaying package configuration, since apt-utils is not installed
+ version_gte 20.10
+ '[' -z '' ']'
+ return 0
+ sh -c 'DEBIAN_FRONTEND=noninteractive apt-get install -y -qq docker-ce-rootless-extras >/dev/null'
debconf: delaying package configuration, since apt-utils is not installed
+ sh -c 'docker version'
Client: Docker Engine - Community
 Version:           20.10.17
```

3.Check the Docker version and verify that Docker Engine is installed correctly by running the hello-world image.

docker –version

docker run hello-world

```
root@Router:~# docker --version
Docker version 20.10.17, build 100c701
root@Router:~# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
7050e35b49f5: Pull complete
Digest: sha256:13e367d31ae85359f42d637adf6da428f76d75dc9afeb3c21faea0d976f5c651
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
```

4. When docker started running, it will create some iptables firewall rules. Since RobustOS Pro will clear firewall rules and create the rules based on WEB or CLI configurations. That means we need to restart the docker service and

freeze the firewall rules firstly.

```
$systemctl restart docker
$uci set svc_mgr.firewall 1
$uci commit
```

