prj.json(project description)

Default

Configure...

HTML...

Component...

Shell...

Open Source Program...

Resource File...

Linux Executable Program...

Shared Library...

Kernel Module...

SkinOS

Sline(CLI)

Remote Control

Local Control

SkinOS

Other Project

SkinOS

Initialization

Shutdown

SkinOS

Web Server

Config

method

prj.json

Default Config file

HTML

**Skin Layer**

Component

ffff

**Linux System and Application**

Executable Program

Shared Library

Shell

API

Kernel Module

Resource file

Register for startup run

HTML Page

Register for shutdown run

Register for network online event

\_shut()

Stop the service

\_take()

Reset the service

\_setup()

Register the service according to the component configuration

\_status()

Query status

\_set()

Set config

\_get()

Get configure

sreset()

spid()

sdelete()

sstart()

Query modify Query

Default Config file

\_service() call telecom\_cpe\_register()

1. Get machine infomation
2. Sending HTTP POST with device information
3. Reutn ttrue to exit

SkinSDK

Top SDK

project

config

rice

tools

$gPLATFORM

Openwrt SDK

Bottom SDK

Linux

project

FPK

/bin/

bin/

/lib/

prj.json:cmd

bin/

/etc/

lib/

lib/

Interval Flash

prj.json:lib

etc/

Configure Flash

etc/

int/

/prj/$PROJECT\_ID

int/

\*.cfg

\*.cfg

**Install**

**Compilation and packaging**

在项目中编写Linux可执行程序

在项目中编写共享库

在项目中编写内核模块

项目及组件开发 在项目中编写脚本程序 调用组件接口

高级开发者

管理组件配置

操作配置

系统服务

在开发中需要用到的API 获取项目信息

组件寄存器

日志

...会不断的添加

移植Linux程序到SkinSDK中-简单版

普通开发者

tiger7/readme.md 移植Linux程序到SkinSDK中-进阶版

移植程序到SkinSDK介绍

移植Openwrt项目到SkinSDK中-简单版

移植Openwrt项目到SkinSDK中-进阶版

认证及权限管理

日志管理

设备时间管理

普通用户

编程开发

远程管理

Telent服务器管理

4G/5G基带管理

系统常用项目组件接口 4G/5G网络管理

2.4G无线热点管理

5.8G无线热点管理

定位管理

路由表管理

UART管理

...跟据新开发的项目会不断的添加

注册为开机运行

注册为关机运行

Iftttserver可执行程序

execl

Ismartus可执行程序

execl

 \_ismartus\_service()

serv\_start

serv\_stop

\_iftt\_service()

\_setup()

\_shut()

PC中的管理程序的开发

upgrade

链接

管理程序

Gather项目

data组件

arch项目

logs

events

beacon

he命令行

libland库

land项目

调用

应用程序开发

remote

User

agent

tui

wui

Called by other components

More Project

…

method

Config

API

library

Component

Project

Web page

…

call other interface of component

executable program

other file

Config file

Register

Service

Linux library API

Syslog, Joint, Init

应用程序开发-附1管理组件配置API

application development

Component

Program

Component interaction structure

Component config( cget/cset )

obj\_t

talk\_t

attr\_t

Project C

\_get/\_set

…

More

Project

Project A

\_get/\_set

\_get/\_set

Project B

Config file

Config file

应用程序开发-与其它组件通信

application development

Component

Program

Component interaction structure

Component call( ccall/scall )

obj\_t

talk\_t

param\_t

Project C

More

Project

Method

…

Project A

Method

Method

Project B

Config file

Config file

组件开发

SkinOS startup

**boole\_t \_setup( obj\_t this, param\_t param ){}**

SkinOS shutdown

**boole\_t \_shut( obj\_t this, param\_t param ){}**

SkinOS event

**boole\_t \_take( obj\_t this, param\_t param ){}**

SkinOS

Service

**boole\_t \_service( obj\_t this, param\_t param ){}**

**boole\_t \_xxxx( obj\_t this, param\_t param ){}**

**...**

System web management

System sline CLI

**talk\_t \_status( obj\_t this, param\_t param ){}**

System agent for local control

**talk\_t \_get( obj\_t this, attr\_t path ){}**

Default Config file

**boole \_set( obj\_t this, talk\_t v, attr\_t path ){}**

System agent for remote control

Config file

应用程序开发-操作配置

终端指令

组件配置管理

可执行程序

组件

配置交互结构

path\_t

obj\_t

配置操作函数(config\_get/config\_set)

talk\_t

…

项目配置(JSON格式)

{JSON对象}

项目配置(JSON格式)

{JSON对象}

项目配置(JSON格式)

{JSON对象}

vpn@l2tpc

vpn@pptpc

openvpn@vpn

...

...

ifname@lan

ifname@wan

ifname@lte

...

client@station

client@dhcps

...

ifname@lte

ifname@lte2

...

uart@nmea

uart@modbus

uart@dtu

gnss@nmea

...

network@bridge

network@vlan

network@keeplive

...

modem@lte

modem@lte2

...

forward@route

forward@firewall

forward@nat

clock@date, clock@restart

modem@device

network@frame

uart@frame

Skinos frame compoent (land@service, land@joint, land@fpk, land@auth, land@reg...)

arch project (arch@data, arch@usb, arch@ethernet, wifi@nssid)

OpenVPN, PPTP, Proftpd, Samba

Busybox(shell, mtd, ip, tc, ...)

Skinos frame library( he, skin,daemon, libland, landnet, ... )

Library (libc, openssl,libevent...)

Kernel Module

Linux Kernel

**TCP(JSON)control protocol(22220)**

SkinOS

management tool

**Local search protocol(UDP-22222)**

**Local query protocol(UDP-22222)**

SkinOS

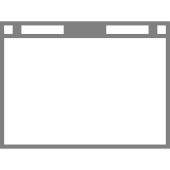
management tool

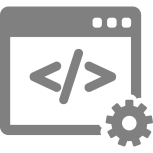
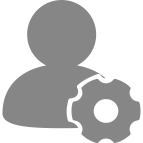
**Create TCP Connection**

Send JSON command

Reply

**Close TCP Connection**

 User Internet WUI/GUI Program



SkinOS API

Application

Linux or Other OS

...

...

API

Config

API

Config

...

project2

project

...

componentB

componentA

Get the componentA configure

project@componentA

Modify the componentA configure

project@componentA={}

client

forward

tui

wui

land

admin

auth

global

route

alg

ssh

syslog

clock

station

nat

firewall

telnet

restart

date

machine

gnss

io

lan

lan2

ifname

assid

assid2

asta

a

nsta

nmea

agent

ethernet

gpio

n

nssid2

nssid

wifi

wisp

wisp2

data

usb

firmware

pci

arch

wan2

wan

modem

lte

lte2

lte2

lte

serial

serial3

serial2

uart