

iTE 可視對講系統

室內機參考設計程式代碼介紹

V0.92

ITE TECH. INC.

修訂記錄

修訂日期	修訂說明	頁次
2014/10/2	初建版本 V0.9	
2014/12/4	新增 13.9 註冊至 SIP Server 一節	22
2014/12/12	新增 29 章: 設定檔	52
2015/3/24	SZ Review	10,21, 47,59, 60

目錄

1. 前言	1
1.1 編寫目的	1
1.2 適用範圍	1
1.3 適用人員	1
2. 室內機模組介紹	2
2.1 ADDRESSBOOK 模組	2
2.2 AUDIO 模組	2
2.3 AUDIORECORD 模組	2
2.4 CAPTURE 模組	2
2.5 CARDLIST 模組	2
2.6 CONFIG 模組	2
2.7 DEVICEINFO 模組	2
2.8 EVENT 模組	2
2.9 GUARD 模組	3
2.10 HOUSE 模組	3
2.11 LINPHONE 模組	3
2.12 MESSAGE 模組	3
2.13 NETWORK 模組	3
2.14 PERIPHERAL 模組	3
2.15 PYIME 模組	3
2.16 REMOTE 模組	3
2.17 SCENE 模組	3
2.18 SCREEN 模組	4
2.19 STRING 模組	4
2.20 UPGRADE 模組	4
2.21 VIDEORECORD 模組	4
2.22 VIDEOMEMO 模組	4
2.23 VOICEMEMO 模組	4
2.24 WEATHER 模組	4
2.25 WEBSERVER 模組	4
3. ADDRESSBOOK 模組	5
3.1 版本資訊	5
3.2 網路設定	5
3.3 其他裝置地址資訊	5
4. AUDIO 模組	7

4.1	播放檔案.....	7
4.2	設定音量.....	7
4.3	控制靜音.....	7
5.	AUDIORECORD 模組.....	9
5.1	錄音.....	9
5.2	播放.....	9
5.3	取得錄音檔案時間長度.....	9
6.	CAPTURE 模組.....	10
6.1	檔案管理.....	10
6.2	監視錄影.....	10
6.3	播放監視錄影.....	11
6.4	拍照.....	11
7.	CARDLIST 模組.....	12
7.1	版本資訊.....	12
7.2	檔案格式.....	12
8.	CONFIG 模組.....	13
8.1	讀取.....	13
8.2	儲存.....	13
9.	DEVICEINFO 模組.....	14
9.1	裝置資訊.....	14
9.2	工作模式.....	14
10.	EVENT 模組.....	15
10.1	註冊室內機.....	15
10.2	上傳事件.....	15
11.	GUARD 模組.....	17
11.1	模式設定.....	17
11.2	安防檢查.....	17
11.3	記錄管理.....	18
11.4	探頭狀態.....	18
12.	HOUSE 模組.....	19
12.1	模式管理.....	19
12.2	取得模組資訊.....	19
12.3	下達模組命令.....	19
12.4	模組介面.....	20
13.	LINPHONE 模組.....	21
13.1	撥接電話.....	21
13.2	監視大廳機或門鈴機.....	21

13.3	播放遠端留影留言	22
13.4	通訊錄管理	22
13.5	通話記錄管理	22
13.6	音量設定	24
13.7	播放音效	24
13.8	勿擾模式設定	25
13.9	註冊至 SIP SERVER	26
14.	MESSAGE 模組	27
14.1	下載留影留言清單	27
14.2	刪除留影留言	27
14.3	下載信息服務清單	28
14.4	刪除信息服務	28
14.5	下載信息服務內容	29
15.	NETWORK 模組	30
15.1	網路狀態	30
15.2	網路重置	30
16.	PERIPHERAL 模組	31
16.1	讀卡	31
16.2	開門	31
17.	PY IME 模組	32
17.1	拼音輸入	32
18.	REMOTE 模組	33
18.1	取得資訊	33
18.2	下達命令	33
18.3	同步勿擾模式設定	34
19.	SCENE 模組	35
19.1	接收輸入裝置	35
19.2	檢查事件	37
19.3	LINPHONE 回呼	38
20.	SCREEN 模組	39
20.1	亮度調整	39
20.2	螢幕保護	39
21.	STRING 模組	40
21.1	範例	40
21.2	UTF-8 字碼查詢方法	40
22.	UPGRADE 模組	41
22.1	更新程序	41

22.2	更新檔案 CRC	42
23.	VIDEORECORD 模組.....	43
23.1	錄影或錄音	43
23.2	播放	43
23.3	取得錄影或錄音檔案時間長度	43
24.	VIDEOMEMO 模組.....	44
24.1	檔案管理	44
24.2	錄影	44
24.3	播放	44
25.	VOICEMEMO 模組	46
25.1	檔案管理	46
25.2	錄音	46
25.3	播放	46
26.	WEATHER 模組.....	48
26.1	解析內容	48
26.2	取得資訊	48
27.	WEBSERVER 模組	50
27.1	取得新訊息通知	50
28.	語言選擇	51
28.1	更換 UI	51
28.2	字型設定	51
29.	設定檔.....	52
29.1	呼叫時間	52
29.2	通話時間	52

插圖目錄
找不到圖表目錄。

1. 前言

1.1 編寫目的

介紹室內機各模組與功能。

1.2 適用範圍

可視對講軟體系統的室內機。

1.3 適用人員

軟體設計人員。

2. 室內機模組介紹

2.1 AddressBook 模組

提供讀取與解析地址簿 XML 的功能。檔案在/project/doorbell_indoor/addressbook.c。

2.2 Audio 模組

提供音效控制相關函數，包括播放、音量與靜音功能。檔案在/project/doorbell_indoor/audio.c。

2.3 AudioRecord 模組

提供錄音相關函數，供 VoiceMemo 模組使用。檔案在/project/doorbell_indoor/audio_record.c。

2.4 Capture 模組

提供錄影回放的檔案管理，以及監視錄影與拍照的功能。檔案在/project/doorbell_indoor/capture.c。

2.5 CardList 模組

提供門禁清單管理與刷卡開門的功能。只有在無門鈴機，由室內機提供門禁管理功能時使用。檔案在/project/doorbell_indoor/cardlist.c。

2.6 Config 模組

提供組態的讀取/修改/儲存功能。檔案在/project/doorbell_indoor/config.c。

2.7 DeviceInfo 模組

提供記錄相關裝置地址資訊的功能，與本地工作模式。檔案在/project/doorbell_indoor/device_info.c。

2.8 Event 模組

提供上傳事件與註冊室內機到控制伺服器的功能。檔案在/project/doorbell_indoor/event.c。

2.9 Guard 模組

提供安防報警相關功能，包括模式控制與記錄管理。檔案在 /project/doorbell_indoor/guard.c。

2.10 House 模組

提供智能家居相關功能。檔案在 /project/doorbell_indoor/house.c 與 /project/doorbell_indoor/house_*.c。

2.11 Linphone 模組

提供 Linphone 相關功能，包括撥接電話，通訊錄管理，通話記錄管理，音量設定，勿擾模式設定等。檔案在 /project/doorbell_indoor/linphone.c。

2.12 Message 模組

提供訊息服務相關功能，從控制伺服器讀取訊息。檔案在 /project/doorbell_indoor/message.c。

2.13 Network 模組

提供網路相關功能，包括網路初始化、網路校時與斷線偵測等。檔案在 /project/doorbell_indoor/network.c。

2.14 Peripheral 模組

提供週邊介面相關功能，包括讀卡、開門等。只有在無門鈴機，由室內機提供門禁管理功能時使用。檔案在 /project/doorbell_indoor/peripheral.c。

2.15 PY IME 模組

提供簡體中文的拼音輸入法相關功能。檔案在 /project/doorbell_indoor/py_ime.c。

2.16 Remote 模組

提供傳遞命令到遠端裝置的功能。檔案在 /project/doorbell_indoor/remote.c。

2.17 Scene 模組

GUI 控制模組，提供讀取/繪製 GUI、接收輸入裝置輸入等功能。檔案在 /project/doorbell_indoor/scene.c。

2.18 Screen 模組

提供螢幕相關功能，包括亮度調整、螢幕保護功能等。檔案在 /project/doorbell_indoor/screen.c。

2.19 String 模組

提供多國語字串的功能。檔案在 /project/doorbell_indoor/string.c。

2.20 Upgrade 模組

提供更新 Firmware、檔案 CRC 檢查與 Recovery 功能。檔案在 /project/doorbell_indoor/upgrade.c。

2.21 VideoRecord 模組

提供視訊通訊時的錄影、錄音與播放功能，供 VideoMemo 與 Capture 模組使用。檔案在 /project/doorbell_indoor/video_record.c。

2.22 VideoMemo 模組

提供本地端視訊錄影錄音檔案的管理功能。檔案在 /project/doorbell_indoor/videomemo.c。

2.23 VoiceMemo 模組

提供留影留言檔案的管理功能。檔案在 /project/doorbell_indoor/voicememo.c。

2.24 Weather 模組

提供解析天氣 XML 資料的功能。檔案在 /project/doorbell_indoor/weather.c。

2.25 WebServer 模組

提供網頁伺服器的功能，包括內部設定網頁與接收遠端命令。檔案在 /project/doorbell_indoor/webserver.c。

3. AddressBook模組

從地址簿中可以得到包括版本、網路設定與各裝置地址的資訊等。

3.1 版本資訊

版本資訊可用來讓控制伺服器決定是否要升級地址簿。

```
char* ptr = AddressBookGetVersion();

// HERE TO USE ptr as version information string
...
// remember to free after used
free(ptr);
```

3.2 網路設定

室內機可由地址簿中統一設定 netmask 與 gateway 的 IP address。

```
char* ptr = AddressBookGetGateway();

// HERE TO USE ptr as gateway ip like 192.168.1.254
...
// remember to free after used
free(ptr);

ptr = AddressBookGetNetMask();

// HERE TO USE ptr as netmask ip like 255.255.255.0
...
// remember to free after used
free(ptr);
```

3.3 其他裝置地址資訊

室內機由地址簿中得知如控制伺服器、大廳機、門鈴機與其他室內機等的位址資訊。

```
#define MAX_COUNT 10
DeviceInfo info;
DeviceInfo infos[MAX_COUNT];
int len;
char* ptr;
```

```

// get device information by ip
AddressBookGetDeviceInfo(&info, "192.168.1.20");
printf(buf, "%s-%s-%s-%s-%s", info.area, info.building, info.unit, info.floor, info.room);

// get all extension devices' information of address "01-02-03-04-05"
len = AddressBookGetDeviceInfoArray(infos, MAX_COUNT, "01", "02", "03", "04", "05");

// get all manager devices' information
len = AddressBookGetManagerInfoArray(infos, MAX_COUNT);

// get all devices' information with camera
len = AddressBookGetCameraInfoArray(infos, MAX_COUNT);

// get device's ip from address
ptr = AddressBookGetDeviceIP("01", "02", "03", "04", "05", "06");

// get center's device infomation
AddressBookGetCenterInfo(&info);

// get entrance's device infomation
AddressBookGetEntranceInfo(&info, &theDeviceInfo);

// get outdoor's device infomation
AddressBookGetOutdoorInfo(&info, &theDeviceInfo);

```

4. Audio模組

由此模組可播放 mp3 與 wav 檔案，以及控制音量與靜音功能。要注意的是目前室內機使用 Linphone 模組提供的播放與音量控制功能，並未使用到此模組的播放與音量控制功能。

4.1 播放檔案

```
// play mp3 file
AudioPlay("C:/test.mp3");

// play wav file
AudioPlay("C:/test.wav");

// play key sound
AudioPlayKeySound();

// play warning sound
AudioPlayWarnSound();

// stop playing
AudioStop();
```

4.2 設定音量

```
// set key volume to 50%
AudioSetKeyLevel(50);
```

4.3 控制靜音

```
bool muted;

// mute audio
AudioMute();

// unmute audio
AudioUnMute();

// get mute status
muted = AudioIsMuted();
// muted should be false right now
```


5. AudioRecord模組

此模組提供本地錄音的功能，檔案格式是私有壓縮格式，附檔名是.svm。另外也提供播放該錄音檔的功能。

5.1 錄音

```
// start record
int ret = AudioRecordStartRecord("C:/test.svm");
if (ret == 0)
{
    // success recording
}

// stop record
ret = AudioRecordStopRecord();
if (ret == 0)
{
    // success stop recording
}
```

5.2 播放

```
// start play
int ret = AudioRecordStartPlay("C:/test.svm");
if (ret == 0)
{
    // success playing
}

// stop playing
AudioRecordStopPlay();
```

5.3 取得錄音檔案時間長度

```
// get time length of record file, in seconds.
int sec = AudioRecordGetTimeLength("C:/test.svm");
```

6. Capture模組

此模組在錄影回放的檔案管理部分，在初始化時會創造一個 **task** 讀取錄影回放全部的檔名，並依日期排序。另外包裝了監視錄影與拍照的功能。

6.1 檔案管理

```
CaptureEntry* entry;
int count;

// get total captured file count
count = CaptureGetCount();

// get unplayed captured 2file count
count = CaptureGetUnplayCount();

// get the first captured file's information
entry = CaptureGetEntry(0);

// delete the first captured file
CaptureDeleteEntry(0);
```

6.2 監視錄影

由於需要記錄是與誰在通話，開始錄影時需要傳入對方的 IP address。

```
// start to capture incoming video
int ret = CaptureStartRecord(MEDIA_VIDEO, "192.168.1.20")
if (ret == 0)
{
    // success to start capturing video
}

// stop capturing video
ret = CaptureStopRecord();
if (ret == 0)
{
    // success to stop capturing video
}

// start to capture incoming audio only phone call
ret = CaptureStartRecord(MEDIA_AUDIO, "192.168.1.21")
if (ret == 0)
```

```
{  
// success to start capturing audio  
}
```

6.3 播放監視錄影

```
// start to play the first captured file  
int ret = CaptureStartPlay(0);  
if (ret == 0)  
{  
// success to start playing captured file  
}  
  
// stop playing captured file  
CaptureStopPlay();
```

6.4 拍照

拍照時可指定拍照時間，輸入 0 表示使用現在時間。

```
char* filename;  
int ret;  
  
// snapshot current video image  
ret = CaptureSnapshot("192.168.191.20", 0);  
if (ret == 0)  
{  
// success to snapshot  
}  
  
// get snapshot file name by snapshot's time  
filename = CaptureGetSnapshotFileNameByTime("01/01/2014 12:00:00");
```

7. CardList模組

此模組初始化時會讀入卡片清單，並另創造一個 task 做刷卡開門的工作。

7.1 版本資訊

版本資訊可用來讓控制伺服器決定是否要升級卡片清單。

```
// get version of card list  
int ver = CardListGetVersion();
```

7.2 檔案格式

卡片清單用 XML 格式儲存，格式如下：

```
<?xmlversion="1.0"encoding="UTF-8"?>  
<CardListver="00000">  
<devro="01-02-01-01-01-01">  
<card>123456789012345678901234567890123456789012345678901234</card>  
<card>123456789012345678901234567890123456789012345678901235</card>  
<card>123456789012345678901234567890123456789012345678901236</card>  
</dev>  
</CardList>
```

其中 /CardList/@ver 是卡片清單的版本號碼，/CardList/dev/@ro 是房門地址，
/CardList/dev/card 是卡片號碼，一共 64 個字元。

8. Config模組

組態分成儲存在 private 分割區與 public 分割區二種。地址的設定會儲存在 private 分割區，其它的設定會儲存在 public 分割區。在 public 分割區的組態提供 CRC 的檢查，毀損時可由 private 分割區的出廠備份組態回復。

8.1 讀取

```
// load configurations
ConfigLoad();

// printf brightness setting
printf("%d", theConfig.brightness);
```

8.2 儲存

```
// setting brightness
theConfig.brightness = 9;

// save public configurations
ConfigSave();

// setting address
strcpy(theConfig.room, "02");

// save public and private configurations
ConfigSaveAll();
```

9. DeviceInfo模組

此模組在初始化時預先從地址簿讀取常用裝置的資訊，包括本機、控制伺服器、大廳機與門鈴機。資訊中包括裝置型別、別名、地址與 IP address。另外也可由本模組取得本機的工作模式。

9.1 裝置資訊

```
// printf local address infomation
printf("%s-%s-%s-%s-%s-%s\n",
    theDeviceInfo.area,
    theDeviceInfo.building,
    theDeviceInfo.unit,
    theDeviceInfo.floor,
    theDeviceInfo.room,
    theDeviceInfo.ext);

// printf center's ip
printf("%s\n", theCenterInfo.ip);

// printf entrance's alias
printf("%s\n", theEntranceInfo.alias);

// printf outdoor's ip
printf("%s\n", theOutdoorInfo.ip);
```

9.2 工作模式

當室內機未設定地址時，此時即在工作模式，位址資訊由目前設定的 IP address 從地址簿中取得。反之則是在一般模式。

```
bool value = DeviceInfoIsInEngineerMode();
if (value)
{
    // in engineer mode
}
else
{
    // in normal mode
}
```

10. Event模組

此模組在初始化時會創造一個 **task**，不斷檢查是否有新的事件，然後上傳至控制伺服器。

10.1 註冊室內機

在開機時會把本機的 **IP address**、**MAC address** 與地址等資訊傳給控制伺服器註冊，以提供控制伺服器判斷 **IP** 是否衝突等資訊用。

```
ITPEthernetInfo info;

// get infomation of the first ethernet card
info.index = 0;
ioctl(ITP_DEVICE_ETHERNET, ITP_IOCTL_GET_INFO, &info);

// send register information to center
EventWriteRegisterDevice(info.hardwareAddress);
```

10.2 上傳事件

上傳的事件種類包括呼叫記錄、安防記錄、社區服務、開門記錄、開門密碼與簡訊服務等。

```
struct timeval tv;

// upload callout start event
EventWriteCalloutLog("192.168.1.20", EVENT_CALLOUT_START);

// upload callout end event
EventWriteCalloutLog("192.168.1.20", callConnected ? EVENT_CALLOUT_END : EVENT_CALLOUT_NOACK);

// get current time
gettimeofday(&tv, NULL);

// upload guard emergency enabled event
EventWriteGuardLog(GUARD_EMERGENCY, GUARD_ENABLE, tv.tv_sec);

// upload guard emergency disabled event
EventWriteGuardLog(GUARD_EMERGENCY, GUARD_DISABLE, tv.tv_sec);

// upload help event
EventWriteHelpInfo("I need help about my pipe", " pipe broken");
```

```
// upload success to open door event
EventWriteOpenDoorLog(EVENT_OPENDOOR_CARD, NULL, true, "1234567890");

// upload fail to open door event
EventWriteOpenDoorLog(EVENT_OPENDOOR_CARD, NULL, false, "1234567890");

// upload password of opening door event
EventWriteOpenDoorPassword(theConfig.door_password);

// upload sms event
EventWriteSmsLog("192.168.1.20");
```


11. Guard模組

此模組負責初始化各探頭的 **GPIO**，依照模式設定檢查各探頭的狀態與回報檢查結果。設定與安防事件的發生會循環性地記錄起來。

11.1 模式設定

包含在家模式、外出模式、就寢模式與撤防模式四種。

```
// set to indoor mode
GuardSetMode(GUARD_INDOOR);

// set to outdoor mode
GuardSetMode(GUARD_OUTDOOR);

// set to sleep mode
GuardSetMode(GUARD_SLEEP);

// set to none mode
GuardSetMode(GUARD_NONE);
```

11.2 安防檢查

檢查結果有三種：檢查正常、檢查異常但無需發出警報聲與檢查異常需發出警報聲三種。

```
// check guard status
int ret = GuardCheck();
if (ret == 0)
{
    // no guard event occur
}
elseif (ret > 0)
{
    // guard event occurred, but need to be quiet
}
else// if (ret < 0)
{
    // guard event occurred, to play alarm
}
```

11.3 記錄管理

記錄目前只保留在記憶體，最大 999 筆，當記錄滿了之後會把最舊的覆蓋掉。

```
// get total guard log count
int count = GuardGetLogCount();

// get the first guard log
GuardLog* log = GuardGetLog(0);
printf("sensor=%d status=%d\n", log->sensor, log->status);

// delete the first guard log
GuardDeleteLog(0);

// clear all guard logs
GuardClearLog();
```

11.4 探頭狀態

可以取得目前探頭狀態，包括是否致能與是否正在警報中。另可取得上次狀態變化時的時間。

```
// get current status of emergency sensor
GuardStatus status = GuardGetSensorStatus(GUARD_EMERGENCY);

time_t rawtime;
struct tm* timeinfo;
char buf[80];

// get last time of emergency sensor which changing status
rawtime = GuardGetSensorTime(GUARD_EMERGENCY);
timeinfo = localtime(&rawtime);
strftime(buf, sizeof(buf), "%Y.%m.%d %H:%M:%S", timeinfo);

printf("%s", buf);
```

12. House模組

此模組統一管理各個智能家居模組，每種模組需實作統一的介面。目前有 AirConditioner、Curtain、Light、Plug 與 TV 模組。另外提供模式管理、提供各模組資訊與下達命令的功能。

12.1 模式管理

包含在家模式、外出模式、就寢模式與無四種。

```
// set to indoor mode
HouseSetMode(HOUSE_INDOOR);

// set to outdoor mode
HouseSetMode(HOUSE_OUTDOOR);

// set to sleep mode
HouseSetMode(HOUSE_SLEEP);

// set to none
HouseSetMode(HOUSE_NONE);
```

12.2 取得模組資訊

各模組狀態包括型別、名稱與開關狀態。

```
#define MAX_COUNT 5

HouseInfo infos[MAX_COUNT];

// get all informations and count of air conditioner module
int count = HouseGetInfoArray(HOUSE_AIRCONDITIONER, infos, MAX_COUNT);
```

12.3 下達模組命令

目前包含開、關、停止、自定設定與取得自定設定。

```
// turn on #0 light
HouseExecAction(HOUSE_LIGHT, 0, HOUSE_OPEN, NULL);

// turn off #1 light
HouseExecAction(HOUSE_LIGHT, 1, HOUSE_CLOSE, NULL);
```

12.4 模組介面

需要實作四個函數，包括初始化 `Init()`、結束 `Exit()`、取得全部資訊 `GetInfoArray()`與執行命令 `ExecAction()`。

13. Linphone模組

此模組透過 Message Queue 的方式下達相關的命令給 Linphone Task。

13.1 撥接電話

```
#define MAX_COUNT 5

DeviceInfo infos[MAX_COUNT];
int len;

// get video window widget
callingRemoteBackground = ituSceneFindWidget(&theScene, "callingRemoteBackground");

// set this widget as video background
LinphoneSetWindowID(callingRemoteBackground);

// answer current incoming call
LinphoneAnswer(0);

// terminate current call
LinphoneTerminate(0);

// terminate all calls except #1
LinphoneTerminateOthers(1);

// call to 192.168.1.20 with video, without preview
LinphoneCall("192.168.1.20", true, false);

len = AddressBookGetDeviceInfoArray(infos, MAX_COUNT, "01", "02", "03", "04", "05");

// call to all extension devices on "01-02-03-04-05"
LinphoneCallDevices(infos, len, false, false);
```

13.2 監視大廳機或門鈴機

```
// watch camera of outdoor or entrance
LinphoneWatchCamera("192.168.1.21");

// stop watch camera
LinphoneStopCamera();
```

13.3 播放遠端留影留言

播放儲存在控制伺服器的留影留言也是使用 SIP 協議，因此使用 Linphone 播放。

```
// play video message #1 on center
LinphonePlayVideoMessage("192.168.1.1", 1);
```

13.4 通訊錄管理

通訊錄使用 Linphone 內建的通訊錄功能做管理。

```
DeviceInfo info;
LinphoneCastor3Friend* fr;

// get device information by IP address
AddressBookGetDeviceInfo(&info, "192.168.1.20");

// add this device to contacts
LinphoneAddFriend(&info, false, "My best neighborhood.");

if (linphoneCastor3.friend_count > 0)
{
    // linphoneCastor3.friends is the contacts array
    fr = &linphoneCastor3.friends[0];

    // delete the first member on the contacts
    LinphoneDeleteFriend(fr->refkey);
}

// clear contacts
LinphoneClearFriends();
```

13.5 通話記錄管理

通話記錄使用 Linphone 內建的通話記錄功能做管理。

```
int i, count;
LinphoneCastor3CallLog* log;
struct tm tm;
char buf[32];

// print all missed logs infomation
count = linphoneCastor3.missed_log_count;
```

```

for (i = 0; i < count; i++)
{
    log = &linphoneCastor3.missed_logs[i];

    strptime(log->time, "%c", &tm);
    strftime(buf, sizeof(buf), "%Y.%m.%d %H:%M", &tm);

    printf("missed log uri=%s time=%s\n", log->uri, buf);
}

// delete the last missed log
LinphoneDeleteCallLog(log->refkey);

// clear all missed logs
LinphoneClearMissedLogs();

// print all received logs infomation
count = linphoneCastor3.received_log_count;
for (i = 0; i < count; i++)
{
    log = &linphoneCastor3.received_logs[i];

    strptime(log->time, "%c", &tm);
    strftime(buf, sizeof(buf), "%Y.%m.%d %H:%M", &tm);

    printf("received log uri=%s time=%s\n", log->uri, buf);
}

// delete the last received log
LinphoneDeleteCallLog(log->refkey);

// clear all received logs
LinphoneClearReceivedLogs();

// print all sent logs infomation
count = linphoneCastor3.sent_log_count;
for (i = 0; i < count; i++)
{
    log = &linphoneCastor3.sent_logs[i];

    strptime(log->time, "%c", &tm);
    strftime(buf, sizeof(buf), "%Y.%m.%d %H:%M", &tm);

    printf("sent log uri=%s time=%s\n", log->uri, buf);
}

```

```
// delete the last sent log
LinphoneDeleteCallLog(log->refkey);

// clear all sent logs
LinphoneClearSentLogs();
```

13.6 音量設定

包括人聲、鈴聲、按鍵聲與麥克風音量的設定。

```
int level;

// get current voice volume
level = LinphoneGetVoiceLevel();

// increase 10% voice volume
LinphoneSetVoiceLevel(level + 10);

// get current ring volume
level = LinphoneGetRingLevel();

// increase 10% ring volume
LinphoneSetRingLevel(level + 10);

// get current key volume
level = LinphoneGetKeyLevel();

// increase 10% key volume
LinphoneSetKeyLevel(level + 10);

// get current mic volume
level = LinphoneGetMicLevel();

// increase 10% mic volume
LinphoneSetMicLevel(level + 10);
// mute mic
LinphoneMuteMic();

// unmute mic
LinphoneUnMuteMic();
```

13.7 播放音效

可以使用 **Linphone** 播放 WAV 檔案，目前有鈴聲、按鍵聲與警報聲。


```

// set ring sound file
LinphoneSetRingSound("ring1.wav");

// play ring sound
LinphonePlayRingSound();

// set key sound file
strcpy(theConfig.keysound, "key1.wav");

// play key sound
LinphonePlayKeySound();

// play warning sound
LinphonePlayWarnSound();

```

13.8 勿擾模式設定

程式需要在 Main Loop 呼叫 `LinphoneCheckDoNotDisturb()` 來檢查勿擾模式狀態，來決定是否要變化 UI。

```

int ret;

// enable do not disturb mode
LinphoneEnableDoNotDisturb();

// disable do not disturb mode
LinphoneDisableDoNotDisturb();

ret = LinphoneCheckDoNotDisturb();
if (ret == 1)
{
    // time to enter do not disturb mode
}
elseif (ret == -1)
{
    // time to leave do not disturb mode
}

if (LinphoneIsInDoNotDisturb())
{
    // in do not disturb mode right now
}

```

13.9 註冊至 SIP Server

程式提供了註冊本機到 SIP Server 的功能：

```
// register to sip server on control server  
LinphoneRegister();
```

以上範例將註冊本機到控制伺服器上的 SIP Server 上，ID 是本機地址，例如 01-01-01-02-01-01。分機為 01 以外的裝置不會做註冊的動作。

14. Message模組

此模組使用 libcurl 的 APIs，透過 HTTP 協定抓取控制伺服器的各種訊息清單與內容。各命令皆透過非同步的方式呼叫，需傳入 callback 函式來通知結果，以避免卡住呼叫端的 Task。

14.1 下載留影留言清單

```
// Callback when the list is download finished
static void GetVideoListCallback(Message* msgs, int count)
{
    int i;

    printf("got %d messages.\n", count);

    for (i = 0; i < count; i++)
    {
        Message* msg = &msgs[i];

        printf("msg %d: topic=%s time=%s\n", i, msg->topic, msg->time);

        // remember to free after used it
        free(msg->topic);
        free(msg->time);
    }
    free(msgs);
}

...

// Download video message list
MessageVideoListGet(GetVideoListCallback);
```

14.2 刪除留影留言

```
// Callback when the message has been deleted
static void DeleteComplete(void)
{
    printf("deleted.\n");
}

// Delete the #1 video message
```

```
MessageVideoDelete(1, DeleteComplete);
```

14.3 下載信息服務清單

```
// Callback when the list is download finished
staticvoid GetTextListCallback(Message* msgs, int count)
{
    int i;

    printf("got %d messages.\n", count);

    for (i = 0; i < count; i++)
    {
        Message* msg = &msgs[i];

        printf("msg %d: topic=%s time=%s imageUrl=%s\n", i, msg->topic, msg->time, msg->image);

        // remember to free after used it
        free(msg->topic);
        free(msg->time);
        free(msg->image);
    }
    free(msgs);
}

...

// Download text message list
MessageTextListGet(GetTextListCallback);
```

14.4 刪除信息服務

```
// Callback when the message has been deleted
staticvoid DeleteComplete(void)
{
    printf("deleted.\n");
}

// Delete the #1 video message
MessageTextDelete(1, DeleteComplete);
```

14.5 下載信息服務內容

內容是一張 JPEG 影像，由控制伺服器負責排版繪製出。

```
// Callback when the image is download finished
staticvoid GetContent(uint8_t* image, int imageSize)
{
    printf("image=0x%X size=%d\n", image, imageSize);

    // remember to free after used it
    free(image);
}

// download text content
MessageTextContentGet(msg->image, GetContent);
```

15. Network模組

此模組在初始化時會創造一個 **task** 初始化網路設定，之後便不斷做著斷線與重置的偵測，以及確認與控制伺服器連線的工作。

15.1 網路狀態

可以偵測網路是否斷線與控制伺服器是否正常。

```
if (NetworkIsReady())
{
    printf("network is ready.\n");
}
else
{
    printf("network is disconnected.\n");
}

if (NetworkServerIsReady())
{
    printf("center is alive.\n");
}
else
{
    printf("center is disconnected.\n");
}
```

15.2 網路重置

當改變網路設定時，需要重置網路。

```
// change network gateway setting
strcpy(theConfig.gw, "192.168.2.254");
ConfigSave();

// reset network setting
NetworkReset();
```

16. Peripheral模組

此模組包含週邊相關函數。

16.1 讀卡

```
char* cardnum = PeripheralReadCard();  
printf("\card number=%s\n", cardnum);
```

16.2 開門

若無法實作偵測門是否在開啟的狀態，請直接實作一個回傳 **false** 的函數。

```
// if door is not opened  
if (!PeripheralDoorIsOpened())  
{  
    // open the door  
    PeripheralOpenDoor();  
}
```

17. PY IME模組

此模組提供簡體中文的拼音輸入法功能，輸出字串的編碼是 UTF-16，每個字佔 2 個 bytes，需要再轉成 UTF-8 才能設給 ITU。

17.1 拼音輸入

```
// declaration
extern unsigned char* py_ime(unsigned char input_py_val[], int* len);

int i, count, len;
char buf[512];

// get all words of "ying" on UTF-16
char* imestr = py_ime("ying", &count);

for (i = 0; i < count; i += 2)
{
    wchar_t wc = imestr[i] | ((wchar_t)imestr[i + 1] << 8);

    // convert to UTF-8
    len = wctomb(buf, wc);
    if (len > 0)
    {
        // the buf will filled with UTF-8 character
    }
}
```


18. Remote模組

此模組使用 libcurl 的 APIs，透過 HTTP 協定傳遞命令到遠端裝置。必要時命令會透過非同步的方式呼叫，需傳入 callback 函式來通知結果，以避免卡住呼叫端的 Task。

18.1 取得資訊

```
// get center's information
ServerInfo* serverInfo = RemoteGetServerInfo();

printf("upgrade fw url=%s\n", serverInfo->firmwareUrl);

// Callback when information is download finished
staticvoid GetOutdoorInfoCallback(ServerInfo* info)
{
    printf("outdoor's cardlist version=%d\n", info->cardListVersion);
}

// get outdoor device's information
RemoteGetOutdoorInfo(GetOutdoorInfoCallback);

// callback when xml is download finished
staticvoid GetWeather(char* xml, int size)
{
    // call weather module to parse
    WeatherParse(xml, size);
}

// get weather's xml
RemoteGetWeather(GetWeather);
```

18.2 下達命令

```
// call elevator down
RemoteCallElevator(false);

// call elevator up
RemoteCallElevator(up);

// check whether card number is valid or not from center
if (RemoteCheckCard("1234567890"))
{
```

```

    // card number is valid
}

// send open door command to entrance or outdoor device
RemoteOpenDoor("192.168.1.20");

// set outdoor settings
strcpy(theConfig.outdoor_ipaddr, "01");
strcpy(theConfig.outdoor_area, "02");
strcpy(theConfig.outdoor_building, "03");
strcpy(theConfig.outdoor_unit, "04");
strcpy(theConfig.outdoor_floor, "05");
strcpy(theConfig.outdoor_room, "06");
strcpy(theConfig.outdoor_ext, "07");

// send outdoor settings to outdoor devices
RemoteSetOutdoorSetting();

```

18.3 同步勿擾模式設定

當本機分機號碼為"01"時，把勿擾模式設定送到其他分機去，否則從分機號碼為"01"的室內機把設定傳送回來。

```

// sync do not disturb configuration with extension devices
if (strcmp(theDeviceInfo.ext, "01"))
    RemoteGetDontDisturbConfig();
else
    RemoteSetDontDisturbConfig();

```

19. Scene模組

在初始階段會初始化 **SDL** 與 **ITU Library**，並讀取 **GUI** 檔案與字型。之後 **Main Task** 在此模組中不斷地接收輸入裝置的輸入，傳給 **ITU Library** 處理，依結果決定是否要重畫畫面。也就是所謂的 **Main Loop**。另外也由這邊檢查各種事件，如安防事件、螢幕保護等。**Linphone** 的事件也由這裡接收再送到 **ITU Library** 來更新畫面。

19.1 接收輸入裝置

在 **Main Loop** 中不斷讀取 **SDL** 的輸入事件，包括實體鍵的按下與放開、觸控屏的按下與放開與手勢等。

```
int SceneRun(void)
{
    SDL_Event ev;

    // main loop
    for (;;)
    {
        bool result = false;
        ...

        // probe SDL input event
        while (SDL_PollEvent(&ev))
        {
            switch (ev.type)
            {
            case SDL_KEYDOWN:
                // send key down event to ITU
                result = ituSceneUpdate(&theScene, ITU_EVENT_KEYDOWN, ev.key.keysym.sym, 0, 0);
                break;

            case SDL_KEYUP:
                // send key up event to ITU
                result = ituSceneUpdate(&theScene, ITU_EVENT_KEYUP, ev.key.keysym.sym, 0, 0);
                break;

            case SDL_FINGERDOWN:
                // send touch down event to ITU
                result = ituSceneUpdate(&theScene, ITU_EVENT_MOUSEDOWN, 1, ev.tfinger.x, ev.tfinger.y);
                break;

            case SDL_FINGERUP:
                // send touch up event to ITU
```

```

        result = ituSceneUpdate(&theScene, ITU_EVENT_MOUSEUP, 1, ev.tfinger.x, ev.tfinger.y);
break;

        // send gesture event to ITU
        case SDL_SLIDEGESTURE:
switch (ev.dgesture.gestureId)
{
case SDL_TG_LEFT:
        result = ituSceneUpdate(&theScene, ITU_EVENT_TOUCHSLIDELEFT, 1, ev.tfinger.x,
ev.tfinger.y);
break;

case SDL_TG_UP:
        result = ituSceneUpdate(&theScene, ITU_EVENT_TOUCHSLIDEUP, 1, ev.tfinger.x,
ev.tfinger.y);
break;

case SDL_TG_RIGHT:
        result = ituSceneUpdate(&theScene, ITU_EVENT_TOUCHSLIDERIGHT, 1, ev.tfinger.x,
ev.tfinger.y);
break;

case SDL_TG_DOWN:
        result = ituSceneUpdate(&theScene, ITU_EVENT_TOUCHSLIDEDOWN, 1, ev.tfinger.x,
ev.tfinger.y);
break;
}
break;
}

        // send timer event to ITU
        result |= ituSceneUpdate(&theScene, ITU_EVENT_TIMER, 0, 0, 0);
if (result)
{
        // need to update screen, call ITU to redraw
        ituSceneDraw(&theScene, screenSurf);

        // flip screen buffer
        ituFlip(screenSurf);
}
}
}

```

19.2 檢查事件

在 Main Loop 會隨時檢查事件來更新 UI。其中包括離開事件、其它 Task 的事件、未接來電、新訊息通知、安防事件、網路事件、勿擾模式與螢幕保護等。

```
int SceneRun(void)
{
    for (;;)
    {
        // check quit event
        if (CheckQuitValue())
            break;

        // process event sent from other tasks
        if (commandQueueSize > 0)
            ProcessCommand();

        // check missed calls to update UI
        CheckMissedCalls();

        // check new messages to update UI
        CheckNewMessages();

        // check guard event to update UI
        CheckGuard();

        // check network event to update UI
        CheckNetwork();

        // check do not disturb mode to update UI
        CheckDoNotDisturb();

        while (SDL_PollEvent(&ev))
        {
            ...
        }

        // check screen saver
        if (theConfig.screensaver_type != SCREENSAVER_NONE &&
            ScreenSaverCheck())
        {
            ...
        }
    }
}
```

19.3 Linphone 回呼

Linphone 主動通知的回呼函數有：

1. `CallEnd()`: 通話結束時通知。
2. `CallReceived()`: 來電通知。
3. `CallConnected()`: 應答通知。
4. `CallError()`: 通話發生錯誤時通知。
5. `SceneCallOnDoNotDisturb()`: 勿擾模式時的來電通知。正常情況下回傳 0。若要無視勿擾模式，在此回傳 1。要進行本地端留言則回傳 -1。轉接至控制伺服器則回傳 2。
6. `SceneCallAllowCallWaiting()`: 插撥通知。允話插撥回傳 1，則否回傳 0。
7. `SceneCameralViewing()`: 當來電時，判斷目前是否在監視門口機或播放控制伺服器端留言。是回傳 1，否回傳 0。
8. `UpdateIndicators()`: 當 Linphone 初始化完成時，更新 UI。
9. `SceneNewLog()`: 當 Linphone 有新的通話記錄產生時通知。

20. Screen模組

當省電模式開啟時，進入螢幕保護會降低 CPU 的頻率以降低功耗。

20.1 亮度調整

亮度的階數是由 Kconfig 中 BACKLIGHT_DUTY_CYCLES 的數量來決定。

```
// set brightness to level 5
ScreenSetBrightness(5);
```

20.2 螢幕保護

當有任何輸入裝置輸入時呼叫 **ScreenSaverRefresh()**來重置進入螢幕保護的時間或離開螢幕保護模式。在 Main Loop 不斷呼叫 **ScreenSaverCheck()**來檢查是否到達進入螢幕保護模式。

```
...
case SDL_FINGERDOWN:
    // refresh sceen saver timer or turn on screen
    ScreenSaverRefresh();
...

// check screen saver timer
if (theConfig.screensaver_type != SCREENSAVER_NONE &&
    ScreenSaverCheck())
{
    // turn off screen
    ScreenOff();
}
```

21. String模組

當需要動態生成多國語的字串時，統一在此模組提供。內部字串統一使用 UTF-8 編碼，依不同語言分成不同陣列宣告。

21.1 範例

```
// get address description in current language
char* desc = StringGetRoomAddress("01", "02", "03", "04", "05", NULL);

// will be "Area 01 Building 02 Unit 03 Floor 04 Room 05" in English
printf("%s", desc);

// need to free string
free(desc);
```

21.2 UTF-8 字碼查詢方法

1. 使用支援 UTF-8 編碼的文字編輯器(例如 UltraEdit)，把欲轉換的字串存檔成 UTF-8 格式的文字檔。
2. 使用支援檢視 16 進位檔案的工具(例如 UltraEdit)檢視該文字檔，即可查到該字串的字碼。
3. 注意某些 UTF-8 文字檔會含有 EF BB BF 的檔頭。
4. 以“中文”兩字為例，用 UltraEdit 存檔成 UTF-8 格式的文字檔後，再使用 16 進位模式檢視，會發現內容為 EF BB BF E4 B8 AD E6 96 87。其中 E4 B8 AD E6 96 87 即為“中文”的 UTF-8 字碼。
5. 在宣告成 C 裡的靜態文字時，要使用“\x”前綴字。以上述“中文”為例，範例如下：

```
// string in UTF-8
static const char* stringChineseCht = "\xE4\xB8\xAD\xE6\x96\x87";
```


22. Upgrade模組

此模組在初始化時會檢查 **public** 分割區與 **temp** 分割區中的 **CRC** 記錄，若 **temp** 分割區損毀會將之格式化；若 **public** 分割區損毀會將之回復至出廠設定。

22.1 更新程序

更新程序包括回復出廠設定模式、重置 **temp** 分割區、更新韌體、更新資源檔、更新地址簿、更新門禁卡片清單與由 **WebServer** 模組進行更新動作等。更新韌體時若有呼叫 `UpgradeSetUrl()` 設定更新韌體的 **URL** 時，更新程序會進行遠端更新，否則進行從 **USB** 磁碟機更新。

```
ServerInfo* serverInfo;
int ret;

// reset to default factory settings
ret = UpgradeProcess(QUIT_RESET_FACTORY);
if (ret)
{
    // upgrade fail
}

// reset temp partition
ret = UpgradeProcess(QUIT_RESET_TEMP_DRIVE);
if (ret)
{
    // upgrade fail
}

// upgrade firmware from center
serverInfo = RemoteGetServerInfo();
UpgradeSetUrl(serverInfo->firmwareUrl);

ret = UpgradeProcess(QUIT_UPGRADE_FIRMWARE);
if (ret)
{
    // upgrade fail
}

// upgrade resource
ret = UpgradeProcess(QUIT_UPGRADE_RESOURCE);
if (ret)
{
    // upgrade fail
}
```

```

}

// upgrade addressbook
ret = UpgradeProcess(QUIT_UPGRADE_ADDRESSBOOK);
if (ret)
{
    // upgrade fail
}

// upgrade cardlist
ret = UpgradeProcess(QUIT_UPGRADE_CARDLIST);
if (ret)
{
    // upgrade fail
}

// upgrade by web
ret = UpgradeProcess(QUIT_UPGRADE_WEB);
if (ret)
{
    // upgrade fail
}

```

22.2 更新檔案 CRC

當被 CRC 檢查機制保護的檔案有更新時，需要更新該檔案的 CRC 值。

```

// update crc value of doorbell_indoor.ini on public partition
UpgradeSetFileCrc(CFG_PUBLIC_DRIVE "":"/doorbell_indoor.ini");

```

23. VideoRecord模組

在通話與監視門口機時，若使用者按下錄影鍵，會觸發錄影功能。在提供本地留言功能的室內機，當進行留影留言時也會觸發該功能。

23.1 錄影或錄音

```
// start record
int ret = VideoRecordStartRecord("C:/test.mov");
if (ret == 0)
{
    // success recording
}

// stop recording
ret = VideoRecordStopRecord();
if (ret == 0)
{
    // success stop recording
}
```

23.2 播放

```
// start play
int ret = VideoRecordStartPlay("C:/test.mov");
if (ret == 0)
{
    // success playing
}

// stop playing
VideoRecordStopPlay();
```

23.3 取得錄影或錄音檔案時間長度

```
// get time length of record file, in seconds.
int sec = VideoRecordGetTimeLength("C:/test.mov");
```

24. VideoMemo模組

此模組在提供本地留影留言的室內機中，提供留影留言的檔案管理部分，在初始化時會創造一個 **task** 讀取留影留言全部的檔名，並依日期排序。另外包裝了錄影與播放的功能。

24.1 檔案管理

```
VideoMemoEntry* entry;
int count;

// get total video memo file count
count = VideoMemoGetCount();

// get unplayed video memo file count
count = VideoMemoGetUnplayCount();

// get the first video memo file's information
entry = VideoMemoGetEntry(0);

// delete the first video memo file
VideoMemoDeleteEntry(0);
```

24.2 錄影

```
// start to record incoming video
ret = VideoMemoStartRecord(MEDIA_VIDEO, "192.168.1.21");
if (ret == 0)
{
    // success to start recording video
}

// stop recording video
ret = VideoMemoStopRecord();
if (ret == 0)
{
    // success to stop recording video
}
```

24.3 播放

```
// start to play the first recorded file
int ret = VideoMemoStartPlay(0);
if (ret == 0)
{
    // success to start playing recorded file
}

// stop playing recorded file
VideoMemoStopPlay();
```

25. VoiceMemo模組

此模組提供語音留言的檔案管理部分。在初始化時會創造一個 **task** 讀取語音留言全部的檔名，並依日期排序。另外包裝了錄音與播放的功能。

25.1 檔案管理

```
VoiceMemoEntry* entry;
int count;

// get total voice memo file count
count = VoiceMemoGetCount();

// get unplayed voice memo file count
count = VoiceMemoGetUnplayCount();

// get the first voice memo file's information
entry = VoiceMemoGetEntry(0);

// delete the first voice memo file
VoiceMemoDeleteEntry(0);
```

25.2 錄音

```
// start to record audio
ret = VoiceMemoStartRecord();
if (ret == 0)
{
    // success to start recording audio
}

// stop recording audio
ret = VoiceMemoStopRecord();
if (ret == 0)
{
    // success to stop recording audio
}
```

25.3 播放

```
// start to play the first recorded file
```

```
int ret = VoiceMemoStartPlay(0);
if (ret == 0)
{
    // success to start playing recorded file
}

// stop playing recorded file
VoiceMemoStopPlay();
```

26. Weather模組

本模組需與 **Remote** 模組配合，先從遠端取得天氣資料的 **XML** 後，再交由本模組解析，取出解析後的資訊。

26.1 解析內容

```
// callback when xml is download finished
static void GetWeather(char* xml, int size)
{
    // call weather module to parse
    WeatherParse(xml, size);
}

// get weather's xml
RemoteGetWeather(GetWeather);
```

26.2 取得資訊

```
        // if weather xml is parsed
if (WeatherIsAvailable())
{
char* ptr;
        int index;
        float value;

// get today date
    ptr = WeatherGetDate(1);
    printf("day 1 date: %s\n", ptr);
    free(ptr);

        // get day 1 symbol picture index
    index = WeatherGetSymbolIndex(1);
    printf("day 1 symbol index: %d\n", index);

        // get today minimum temperature
    value = WeatherGetTemperatureMin(1);
    printf("day 1 min temperature: %f\n", value);

        // get today maximum temperature
    value = WeatherGetTemperatureMax(1);
    printf("day 1 max temperature: %f\n", value);
```



```
    // get today symbol name
    ptr = WeatherGetSymbolName(1);
    printf("day 1 symbol name: %s\n", ptr);
    free(ptr);

    // get today wind speed name
    ptr = WeatherGetWindSpeedName(1);
    printf("day 1 wind speed: %s\n", ptr);
    free(ptr);
}
```

27. WebServer模組

本模組封裝了 Libmicrohttpd 網頁伺服器的功能，提供了組態設定網頁，與接收遠端命令與通知的功能。

27.1 取得新訊息通知

留影留言訊息通知只有在非本地端留影留言的室內機機型才有效。

```
// get new text message count
int textCount = WebServerGetNewTextMsgCount();

// get new video message count
int videoCount = WebServerGetNewVideoMsgCount();
```

28. 語言選擇

系統提供動態切換語言的功能。除了在 **String** 模組中提供不同語言的字串，另外不同的語言也提供不同的 **UI(.itu)** 檔案。字型檔因為包含亞洲語系全部字碼，不用再另行切換。

28.1 更換 UI

在參數設置中更換語言時，會依語系而重新讀取 **doorbell_indoor.itu** (英文)、**doorbell_indoor_chs.itu** (簡體中文)或 **doorbell_indoor_cht.itu** (繁體中文)。相關的程式碼可以參考 `\project\doorbell_indoor\layer_setting_display.c`

28.2 字型設定

在系統的 **Kconfig** 裡的 **System** 選項裡，可以設定系統字型。選項是 **Font file name**，預定值是 **WenQuanYiMicroHeiMono.ttf**，包含了絕大部份亞洲字。網址：
<http://wenq.org/wqy2/index.cgi?MicroHei>

在系統中，是在 **Scene** 模組裡將該字型讀取入 **ITU** 模組，程式碼如下：

```
ituFtInit();  
ituFtLoadFont(0, CFG_PRIVATE_DRIVE ":/font/" CFG_FONT_FILENAME, ITU_GLYPH_8BPP);
```

29. 設定檔

設定檔儲存在 `public` 分割區下的 `/doorbell_indoor.ini`。其中未能在 UI 中設定的選項，需要手動修改該檔案的部分，說明如下：

29.1 呼叫時間

指的是呼叫對方到對方接起來的時間。當時間內對方未接起來時，裝置將主動掛斷通話。項目是 `dial_time`，單位是秒。

29.2 通話時間

指的是對方接起來後到一方掛斷的時間。當時間內雙方未掛斷時，裝置將主動掛斷通話。項目是 `calling_time`，單位是秒。