FAW HS7

IVI System Design

TTB-API User Guide

Released on the: 2016.07.14

Version Number: 0.100

Neusoft Automotives ChangChun

**Version History**

| No. | Version | Section | Brief Description | Date | Responsible |
| --- | --- | --- | --- | --- | --- |
| 1 | 0.100 | All | Initial Create | 2016-08-31 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Content

[1 概述 4](#_Toc23428)

[1.1 术语&缩略词 4](#_Toc5883)

[1.2 Block 4](#_Toc32001)

[2 相关文件 4](#_Toc28120)

[3 接口说明 4](#_Toc11789)

[3.1 TTB-API使用说明 4](#_Toc758)

[3.2 函数说明 5](#_Toc6801)

[3.2.1 通用接口 5](#_Toc19440)

[3.2.2 初始化接口 5](#_Toc23984)

[3.2.3 获取当前系统时间 6](#_Toc19336)

[3.2.4 获取当前状态栏具体状态 6](#_Toc6826)

[3.2.5 注册 6](#_Toc7545)

[3.2.6 状态栏回调 6](#_Toc21977)

[3.2.7 快捷菜单栏回调 7](#_Toc30204)

[3.2.8 通知栏回调 7](#_Toc22205)

[3.2.9 蓝牙电话状态回调 7](#_Toc23086)

[3.2.10 Titlebar快捷菜单通用接口（set、get） 7](#_Toc25385)

[3.2.11 Titlebar短信播报控制接口 8](#_Toc16642)

[4 Refrences 9](#_Toc25868)

# 概述

本文档主要描述TTB-API。

## 术语&缩略词

本文中涉及的术语及缩略词如下：

|  |  |  |
| --- | --- | --- |
| NO. | 缩略语 | 含义 |
| 1 | USB | Universal Serial Bus 通用串行总线 |
| 2 | BT | Bluetooth 蓝牙 |
| 3 | WLAN | Wireless Local Area Networks 无线局域网络 |
| 4 | T-Box | Telematics Box |

## Block

应用程序使用libappTBAR.so 和 HMI\USB\BT等进行通信，并提供头文件和接口说明。

ama-app-tbar.service 用于启动tbar

libappTBAR.pc 配置文件，用于findpkg

app-tbar-0.1.1-Linux.rpm 系统安装tbar

mount -o remount rw /

rpm -ivh app-tbar-0.1.1-Linux.rpm --force --nodeps

# 相关文件

无

# 接口说明

## TTB-API使用说明

约束条件：

* 系统中AppManager进程正常运行；
* 使用TTB-API前需要首先初始化AppSDK；
* 使用TTB-API前需要其次初始化tbar\_preinit；

## 函数说明

typedef int(\* ST\_Callback\_t)(int key, int keyValue);//status bar of titlebar E\_STATUS\_BAR

typedef int(\* STime\_Callback\_t)(std::string keyValue);//systime status of titlebar

typedef int(\* TSC\_Callback\_t)(int key, int keyValue);//shortcut status of titlebar E\_SHORTCUT\_BAR

typedef int(\* NT\_Callback\_t)(int key, int keyValue);//notification status of titlebar E\_NOTI\_BAR

typedef int(\* BTPH\_Callback\_t)(BTPHstatus call\_status);//bt\_phone status

bool isNum(std::string str);// check is num or not for stoi

int tbar\_init(void);//tbar\_init

int tbar\_GetCurTime(std::string &curTime);// get curent sys time

int tbar\_GetCurStatus(std::map<int, int> &tBARStatusBar);// get current status of title

int tbar\_Register(void);// titlebar register

int tbar\_STCallback(ST\_Callback\_t STcallback,STime\_Callback\_t Stimecallback);//callback of status of title

int tbar\_SCCallback(TSC\_Callback\_t SCcallback);//callback of shortcut

int tbar\_NTCallback(NT\_Callback\_t NTcallback);//callback of notification status

int tbar\_BTPHCallback(BTPH\_Callback\_t BTPHcallback);//bt\_phone status with bt\_phone callback closely

bool tbar\_Ctrl(tbarCtrl &\_tbarCtrl);/\* set or get SC & etc. \*/

int tbar\_BTPH\_SMSPlay(smsCtrl \_smsCtrl);//sms play ctrl

### 通用接口

|  |  |  |
| --- | --- | --- |
| *declare* | bool isNum(std::string str) | |
| *description* | 数据类型为string的数据有转换为int型的需求，转换前需要做判断，否则会抛异常导致应用启动失败 | |
| *parameter* | std::string str | 需要转换int的string数据 |
| *return* | true 可以转换int false 不能转int | |

### 初始化接口

|  |  |  |
| --- | --- | --- |
| *declare* | int tbar\_init(void); | |
| *description* | 初始化titlebar相关的bt、wlan模块及其回调函数注册（仅前屏） | |
| *parameter* | none |  |
| *return* | 0 成功 | |

### 获取当前系统时间

|  |  |  |
| --- | --- | --- |
| *declare* | int tbar\_GetCurTime(std::string &curTime); | |
| *description* | 获取当前时间，用于状态栏显示 | |
| *parameter* | std::string &curTime | 当前时间 格式xx：xx：xx（包含秒） |
| *return* | 0 成功 -1 失败 | |

### 获取当前状态栏具体状态

|  |  |  |
| --- | --- | --- |
| *declare* | int tbar\_GetCurStatus(std::map<int, int> &tBARStatusBar); | |
| *description* | 获取当前状态栏的状态 | |
| *parameter* | std::map<int, int> &tBARStatusBar | 包括以下项目的状态 E\_STATUS\_BAR （在tbar\_types.h中定义）  状态对应E\_STATUS\_USB\_IVI、E\_STATUS\_USB\_RSE、E\_STATUS\_SWI、E\_STATUS\_BTPH\_BAT、E\_STATUS\_BTPH\_RSSI、E\_STATUS\_WLAN\_RSSI、E\_STATUS\_HOT\_SPOTS、E\_STATUS\_TBOX\_COMS、E\_STATUS\_TBOX\_RSSI、E\_STATUS\_TBOX\_NETT等（在tbar\_types.h中定义） |
| *return* | 0 成功 -1 失败 | |

### 注册

|  |  |  |
| --- | --- | --- |
| *declare* | int tbar\_Register(void); | |
| *description* | 注册 | |
| *parameter* | none |  |
| *return* | 0 成功 -1 失败 | |

### 状态栏回调

|  |  |  |
| --- | --- | --- |
| *declare* | int tbar\_STCallback(ST\_Callback\_t STcallback,STime\_Callback\_t Stimecallback); | |
| *description* | 注册状态栏回调函数 | |
| *parameter* | ST\_Callback\_t STcallback | 状态栏回调typedef int(\* ST\_Callback\_t)(int key, int keyValue);  Key对应E\_STATUS\_BAR （在tbar\_types.h中定义）  keyValue对应E\_STATUS\_USB\_IVI、E\_STATUS\_USB\_RSE、E\_STATUS\_SWI、E\_STATUS\_BTPH\_BAT、E\_STATUS\_BTPH\_RSSI、E\_STATUS\_WLAN\_RSSI、E\_STATUS\_HOT\_SPOTS、E\_STATUS\_TBOX\_COMS、E\_STATUS\_TBOX\_RSSI、E\_STATUS\_TBOX\_NETT（在tbar\_types.h中定义）等 |
|  | STime\_Callback\_t Stimecallback | 系统时间回调typedef int(\* STime\_Callback\_t)(std::string keyValue);  系统时间发生改变时通过3.2.2获取系统时间更新状态栏 |
| *return* | 0 成功 -1 失败 | |

### 快捷菜单栏回调

|  |  |  |
| --- | --- | --- |
| *declare* | int tbar\_SCCallback(TSC\_Callback\_t SCcallback); | |
| *description* | 注册快捷菜单回调函数 | |
| *parameter* | TSC\_Callback\_t SCcallback | typedef int(\* TSC\_Callback\_t)(int key, int keyValue);  Key对应E\_SHORTCUT\_BAR （在tbar\_types.h中定义） |
| *return* | 0 成功 -1 失败 | |

### 通知栏回调

|  |  |  |
| --- | --- | --- |
| *declare* | int tbar\_NTCallback(NT\_Callback\_t NTcallback); | |
| *description* | 注册通知栏回调函数 | |
| *parameter* | NT\_Callback\_t NTcallback | typedef int(\* NT\_Callback\_t)(int key, int keyValue);  Key对应E\_NOTI\_BAR （在tbar\_types.h中定义） |
| *return* | 0 成功 -1 失败 | |

### 蓝牙电话状态回调

|  |  |  |
| --- | --- | --- |
| *declare* | int tbar\_BTPHCallback(BTPH\_Callback\_t BTPHcallback); | |
| *description* | 注册蓝牙电话回调函数 | |
| *parameter* | BTPH\_Callback\_t BTPHcallback | //support 6 calls max  struct BTPHstatus  {  int call\_total; //total call at current time  int call\_ID[6]; //index of call from basis  int call\_status[6]; //call status E\_STATUS\_BTPH\_CALL  std::string tel\_name[6]; //name of call  std::string tel\_num[6]; //Tel num of call  };  typedef int(\* BTPH\_Callback\_t)(BTPHstatus call\_status); |
| *return* | 0 成功 -1 失败 | |

### Titlebar快捷菜单通用接口（set、get）

|  |  |  |
| --- | --- | --- |
| *declare* | bool tbar\_Ctrl(tbarCtrl &\_tbarCtrl); | |
| *description* | 目前支持快捷菜单栏设置和获取 | |
| *parameter* | tbarCtrl &\_tbarCtrl | struct tbarCtrl  {  int seat\_no; //E\_SEAT\_NO\_TBAR  int tbarID; //E\_TBAR\_ID  int tbarType; //E\_SHORTCUT\_BAR  int param[6]; //E\_SC\_GET--OUT E\_SC\_SET--IN  std::string sparam[6]; //E\_SC\_GET--OUT E\_SC\_SET--IN  bool result; //true--success false--failure  }; |
|  | int seat\_no; | 座位号 E\_SEAT\_NO\_TBAR （在tbar\_types.h中定义） |
|  | int tbarID; | 功能ID E\_TBAR\_ID（在tbar\_types.h中定义） |
|  | int tbarType; | 快捷菜单栏的内容 E\_SHORTCUT\_BAR （在tbar\_types.h中定义） |
|  | int param[6]; | E\_SC\_GET时为出参 E\_SC\_SET时为入参 协商使用 |
|  | std::string sparam[6]; | E\_SC\_GET时为出参 E\_SC\_SET时为入参 协商使用 |
|  | bool result; | 接口状态，true表示操作成功 false表示操作失败 |
| *return* | True 成功 false 失败 | |

### Titlebar短信播报控制接口

|  |  |  |
| --- | --- | --- |
| *declare* | int tbar\_BTPH\_SMSPlay(smsCtrl \_smsCtrl); | |
| *description* | 给kanzi提供短信播报控制接口，包括play、stop、clear | |
| *parameter* | smsCtrl \_smsCtrl | struct smsCtrl  {  int operate; //E\_SMS\_OP  int identity; //E\_SMS\_ID  int clear\_range; //E\_SMS\_CR  int param[6]; //Extended  std::string sparam[6]; //Extended  bool result; //true--success false--failure  }; |
|  | int operate; | 操作码 E\_SMS\_OP （在tbar\_types.h中定义） |
|  | int identity; | Id 标识E\_SMS\_ID（在tbar\_types.h中定义） |
|  | int clear\_range; | 清除消息范围 E\_SMS\_CR （在tbar\_types.h中定义） |
|  | int param[6]; | 协商使用 |
|  | std::string sparam[6]; | 协商使用 |
|  | bool result; | 接口状态，true表示操作成功 false表示操作失败 |
| *return* | 0成功 -1失败 | |

# Refrences

<http://10.19.118.120/svn/RAD/SYS-Workspace/02.RequirementDevelopment(RD)/01.功能式样/HS7_IVI_FunctionSpec_Home&Title.docx>