

江苏东大集成电路 .Net

开发指南

Revision: v1.1

此开发指南仅对开发人员使用该 SDK 作指导，具体技术细节请参阅类库。

同时本指南认为阅读者已具备一定的.Net 开发基础，对于.Net 中特有的名词、技术不作具体解释。

1. 开发者知识基础要求

在开发中，要求开发者对方法、属性、委托、事件、集合的概念有一定的了解。对单件设计模式有一定了解。电话、短消息的类实例初始化建议通过 `GetInstance()` 方法进行，以保证内存全局唯一性。

可参考文献资料：

- 1、<http://msdn.microsoft.com/zh-cn/library/ms173171.aspx>
- 2、[http://msdn.microsoft.com/zh-cn/library/ms173171\(v=vs.80\).aspx](http://msdn.microsoft.com/zh-cn/library/ms173171(v=vs.80).aspx)
- 3、<http://msdn.microsoft.com/zh-cn/library/awbftdfh.aspx>
- 4、[http://msdn.microsoft.com/zh-cn/library/17sde2xt\(v=vs.80\).aspx](http://msdn.microsoft.com/zh-cn/library/17sde2xt(v=vs.80).aspx)
- 5、<http://msdn.microsoft.com/zh-cn/library/system.collections.ilist.aspx>

2. 系统结构



3. 类关系图

暂无

4. 设计思想概要

在本 SDK 中,涉及硬件的操作方法,统一使用 static 修饰符,不需要实例化类对象,应用程序接受 SDK 层状态改变通过.Net 事件机制。

电话本、通话记录、短消息的持久化通过数据库保存,应用访问此类数据通过 Collection 方式,做了隔离,并对于通话记录信息,其特性决定了其可读取、删除,但不允许添加,通过 Collection,对于此类操作做限制。

5. 编程建议

5.1 需使用相关功能的情况下

- 1) 初始化模块
 - a) 调用 SEUIC.Phone.Initialize.Init()或其重载指定加载特定功能模块
 - b) 确认该函数是否调用成功,若未成功,重新调用或复位机器
- 2) 反初始化
 - a) 程序结束时请确保调用 SEUIC.Phone.Initialize.UnInit(),特别俗注意程序异常退出时,该方法能够执行到。
备注:该方法执行不到的风险为程序结束后会根据初始化模块的不同有线程继续执行,此为调用的 Win32API处于临界区,当程序异常退出时,不能正常清理资源导致。
- 3) 各模块的功能调用
 - a) 电话
 - b) 短信
 - c) 拨号上网
 - d) 模块及网络信息

5.2 不需使用电话短信模块时

请调用 SEUIC.Phone.Initialize.Init()方法的重载指定需加载的模块。

备注:若需禁用模块或复位模块,请同样调用 SEUIC.Phone.Initialize.Init()方法初始化

5.3 开发注意事项

- I 系统处于电话状态时,请勿发送短信。可在发送短信前查询模块状态,详见 [6.4.6](#)
- I 对于发送短信等可能需时较长的同步操作,请尽量通过线程方式异步调用,避免造成界面不响应。
- I 发送 AT 命令,为避免错误调用造成不良影响,2011 年 8 月 2 日之后版本不再对外提供,详见 [6.4.16](#)
- I 处于非电话状态时 (ModuleStatus.STATE_CALL),接听电话和挂断电话方法调用无效,系

统不再接受该命令 ([6.1.3](#) [6.1.4](#))。

6. 各模块开发指南

6.1 电话

6.1.1 拨打电话

为静态方法，使用方法如下：

`Call.MakeCall(sCallNumber);`

调用此方法的返回值为 `CallResult` 枚举，

成功返回 `CallResult.RIL_RESULT_OK`

失败返回 `CallResult.RIL_RESULT_ERROR`

枚举的其他状态暂时未使用

拨号成功后，模块会开始拨号过程，在此过程中会上报状态并触发相应事件，详细请见类库中各事件说明。

6.1.2 拨打分机号 (DTMF)

为静态方法，使用方法如下：

`Call.SendDTMF(string sNumber);`

返回值为 `bool`，指示方法调用是否成功，如成功，则发送成功 DTMF。

6.1.3 应答

为静态方法，使用方法如下：

`Call.Answer();`

返回值为 `bool`，指示方法调用是否成功，如成功，则应答接通电话。

6.1.4 挂断

为静态方法，使用方法如下：

`Call.HangUp();`

返回值为 `bool`，指示方法调用是否成功，如成功，则挂断电话。

6.1.5 打开/关闭扬声器

为静态方法，使用方法如下：

打开扬声器

```
Call.Speaker(true);
```

关闭扬声器

```
Call.Speaker(false);
```

6.1.6 电话各状态事件

电话的时间同样功能的事件提供两个版本，一个为无来电显示的，一个为包含来电显示信息的，请根据自己的需要选择相应的事件绑定处理。

6.2 短消息

6.2.1 发送短消息

使用类似如下代码发送邮件

```
if(SMS.Sms.GetInstance().SMSSend(sDescAddress, sSMSContent))
{
    //do things when send success
}
else
{
    //do things when send fail
}
```

6.2.2 事件

1) 发送成功事件

绑定事件：

```
m8ms.OnSMSSendSuccessEvent += new
SEUIC.Phone.SMS.Sms.SMSSendNotifyEvent(m8ms_OnSMSSendSuccessEvent);
```

2) 短消息发送失败事件

绑定事件：

```
m8ms.OnSMSSendFailEvent += new
SEUIC.Phone.SMS.Sms.SMSSendNotifyEvent(m8ms_OnSMSSendFailEvent);
```

3) 短消息接收事件

绑定事件：

```
mSms.OnSMSReceivedEvent += new  
SEUIC.Phone.SMS.Sms.SMSReceivedNotifyEvent (mSms_OnSMSReceivedEvent);
```

6.3 拨号上网

6.3.1 拨号

使用类似如下代码拨号：

```
if (mRas.DialUp("", "", "*99***1#"))  
{  
    //do things when ras connect success  
}  
else  
{  
    //do things when ras connect fail  
}
```

6.3.2 挂断

```
if (mRas.HangUp("", "", "*99***1#"))  
{  
    //do things when ras hangup success  
}  
else  
{  
    //do things when ras hangup fail  
}
```

6.3.3 获取连接状态

获取的连接状态为枚举类型，示例如下：

```
RAS.RASConnState rasConnState = mRas.GetStatus();
```

6.3.4 事件

主要提供连接成功事件以及连接失败事件。
可根据自己逻辑需要绑定处理。

6.4 模块

模块相关主要提供了模块的控制，模块状态、信息的获取，网络信息的获取等方法，均为 static 类型的方法，不需要实例化对象。

6.4.1 启用模块

```
if (Module.Module.ModuleEnable())
{
    //do things when ModuleEnable success
}
else
{
    //do things when ModuleEnable fail
}
```

6.4.2 禁用模块

```
if (Module.Module.ModuleDisable())
{
    //do things when ModuleDisable success
}
else
{
    //do things when ModuleDisable fail
}
```

6.4.3 复位模块

```
if (Module.Module.ResetModule())
{
    //do things when ResetModule success
}
```

```

}
else
{
    //do things when ResetModule fail
}

```

6.4.4 启用模块异常检测

```

if (Module.Module.ModuleAbnormalDetectEnable())
{
    //do things when ModuleAbnormalDetectEnable success
}
else
{
    //do things when ModuleAbnormalDetectEnable fail
}

```

6.4.5 禁用模块异常检测

```

if (Module.Module.ModuleAbnormalDetectDisable())
{
    //do things when ModuleAbnormalDetectDisable success
}
else
{
    //do things when ModuleAbnormalDetectDisable fail
}

```

6.4.6 获取模块状态

返回的模块状态为枚举类型：

```
Module.ModuleStatus moduleStatus = Module.Module.GetModuleStatus();
```

6.4.7 获取模块名称、版本等原始信息

返回的为模块原始信息：


```
string sModuleInfo = Module.Module.GetModuleInfo();
```

6.4.8 获取无线网络类型

返回的为枚举类型：

```
Module.NetworkType networkType= Module.Module.GetNetworkType();
```

6.4.9 获取电子序列号 ESN

返回的为 string 类型：

```
string sESN = Module.Module.GetESN();
```

此方法暂未实现

6.4.10 获取国际移动用户识别码 IMSI

返回 string 类型的 IMSI：

```
string sIMSI = Module.Module.GetIMSI();
```

6.4.11 获取移动设备识别号 MEID

返回 string 类型的 MEID：

```
string sMEID = Module.Module.GetMEID();
```

6.4.12 获取信号强度 RSSI

返回的为 0-7 的 int 类型的信号表示：

```
int iRSSI = Module.Module.GetRSSI();
```

注意：此处返回的信号值表述范围为 0-7，请注意，如需其他范围，表示方式，请自行转换、

6.4.13 获取 EVDO 网络信号强度 HDRSSI

返回 int 类型的 HDRSSI：

```
int iRSSI = Module.Module.GetHDRSSI();
```

6.4.14 获取小区识别码 CI

返回值为 string 类型：

```
string sCellID = Module.Module.GetCellID();
```

6.4.15 获取位置区码 LAC

返回值为 string 类型：

```
string sLAC = Module.Module.GetLAC();
```

6.4.16 直接向模块发送 AT 命令

返回值为 string 类型：

```
string sATCMDResponse = Module.Module.SendATCommand(sATCMD);
```

6.4.17 获取 EVDO 工作模式

返回值为枚举类型：

```
EVDOMode evdMode = Module.EVDO.GetNetWorkMode();
```

6.4.18 设置 EVDO 工作模式

返回是否设置成功，传入参数为枚举：

```
if(Module.EVDO.SetNetWorkMode())
{
    //do things when set evdo mode success
}
else
{
    //do things when set evdo mode fail
}
```

此方法暂未实现

6.4.19 获取 EVDO 基站定位信息

返回值为 EVDOLocationInfo类实例：

```
EVDOLocationInfo evdoLocationInfo = Module.EVDO.GetEVDOLocationInfo();
```

6.4.20 获取扬声器音量

返回当前扬声器音量

```
int iSpeakerVolume = GetSpeakerVolume();
```

6.4.21 设置扬声器音量

设置当前扬声器音量，参数为音量值，返回值为 bool 类型，指示是否设置成功

```
bool SetSpeakerVolume(int iSpeakerVolume);
```

6.4.22 获取麦克风音量

返回当前麦克风音量

```
int iSpeakerVolume= GetMicVolume();
```

6.4.23 设置麦克风音量

设置当前麦克风音量，参数为音量值，返回值为 bool 类型，指示是否设置成功

```
bool SetMicVolume(int iSpeakerVolume);
```

6.4.24 获得SIM卡状态

获得模块当前状态是否有SIM卡，返回值bool类型，指示是否有SIM卡

```
bool IsSIMCardIn();
```