

# 添加多国语言

## 字库

采用 unicode 编码，逻辑字体采用点阵字体，字体文件可由 FontMaker 生成，然后用 7zip 压缩。

MiniGUI 的配置文件 MiniGUI.cfg 中必须添加如下配置：

```
[sxf]
```

```
font_number=1
```

```
name0=sxf-arialuni-rrncnn-16-16-UTF-8
```

```
fontfile0=font/arialuni_U16.7z
```

其中 sxf-arialuni-rrncnn-16-16-UTF-8

sxf 表示字体名称为 sxf

arialuni 表示 family 为 arialuni

UTF-8 表示字符集为 UTF-8

16-16 表示点阵的大小为 16X16

字体文件的名称为 font/arialuni\_U16.7z

MiniGUI 在初始化的时候会加载并解压这个字体文件。

Unicode 字库制作的具体说明可以参考 2014-12-24 的补丁 unicode\_font\_patch\_sdkv1.1.rar

# 代码分析

## 1. 创建逻辑字体

```
logFont = CreateLogFont("sxf", "arialuni", "UTF-8",  
                        FONT_WEIGHT_REGULAR, FONT_SLANT_ROMAN,  
                        FONT_SETWIDTH_NORMAL,  
                        FONT_OTHER_AUTOSCALE, FONT_UNDERLINE_NONE,  
                        FONT_STRUCKOUT_NONE, 18, 0);
```

创建了一个名称为 sxf, family 为 arialuni , 字符集为 UTF-8 的逻辑字体 , 自动缩放 , 没有下划线 , 不不加粗 , 不倾斜 , 期望大小为 18 像素高。

## 2. 加载字符 string 文件

调用 initLabel(language);

需要指定当前的语言。

initLabel 根据 language 加载对应的语言文件 , 例如 language 为 LANG\_CN , 那么需要加载的文件为/res/lang/zh-CN.bin

然后将语言文件中的每一行存到一个向量容器 Vector 中。

## 3. 获取需要的字符串对应的 unicode 编码

调用 ResourceManager::getLabel(labelIndex);

cdrLang 中的 getLabel 会根据这个 labelIndex 到语言文件中的 Vector 去寻找对应的 unicode 编码。

## 4. 切换语言

切换语言会调用 ResourceManager::updateLangandFont(int langValue)

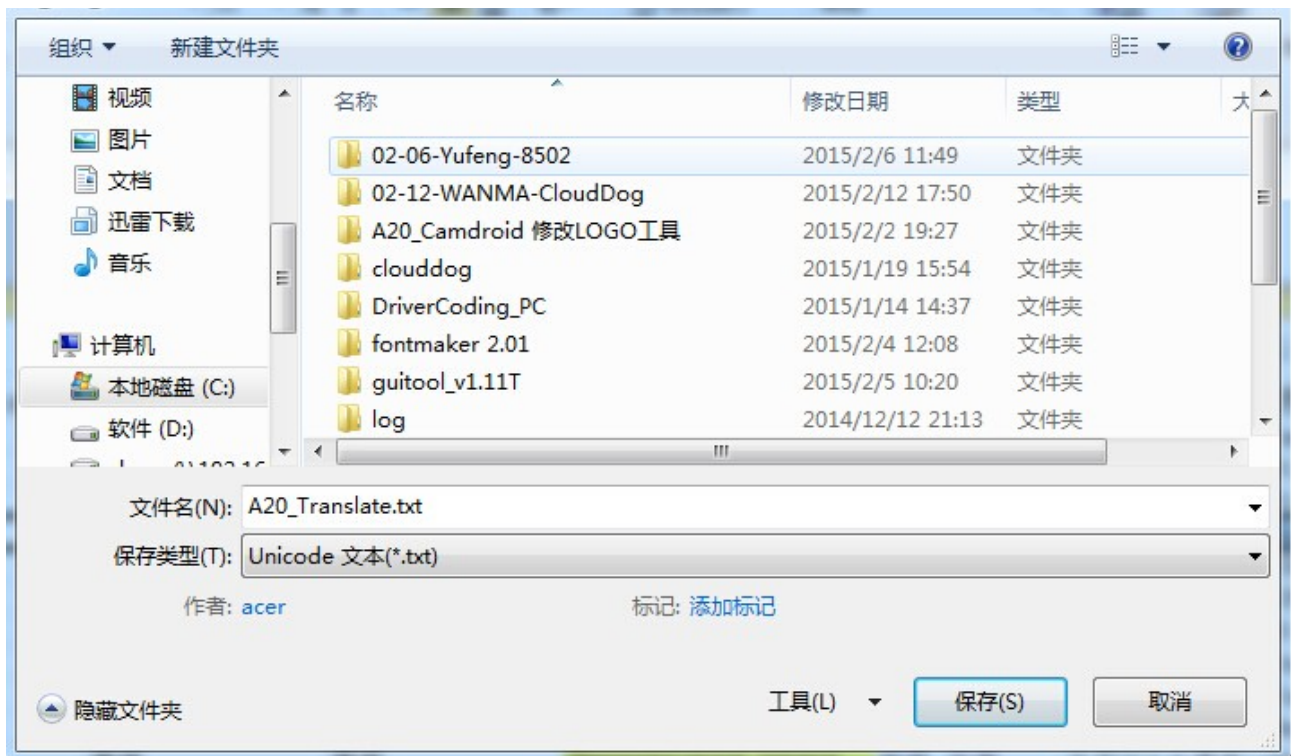
根据新的语言 , 重新调用 iniLabel 加载语言文件。

# 增加一种语言

## 一 添加语言字符串,生成需要的文件和定义

1. 修改 excel 表 A20\_Translate.xlsx , 添加对应的语言字符串

2. 将 A20\_Translate.xlsx 另存为 Unicode 文本 A20\_Translate.txt



3. 生成语言文件和语言字符串枚举定义

运行脚本 newcdr/tools/MultiLingual/language\_split.sh

例如：

```
$ ./language_split.sh A20_Translate.txt
select delimiter:
  1. table (Unicode txt convert from excel)
  2. comma (csv file convert from excel)
1
select Unicode txt
convert encoding from UTF-16 to UTF-8
remove ctrl lines
find 9 columns
total 7 languages
LANG_LABEL_ID in column 1
```

```
language 0: ----->   en.bin
language 1: ----->   zh-CN.bin
language 2: ----->   zh-TW.bin
language 3: ----->   russian.bin
language 4: ----->   korean.bin
language 5: ----->   jpn.bin
language 6: ----->   italian.bin
```

excel 生成的 unicode 文本是以制表符分隔的, 所以分隔符选择 1.

在 out 目录生成了对应的语言文件, 例如 en.bin

根据 excel 中第一列的 LANG\_LABEL\_ID 生成 newcdr/include/misc/cdrLang.h 中

enum LANG\_STRINGS 的定义.

## 二 修改相应的代码

### 1. 添加 LANGUAGE 枚举中的定义

include/misc/cdr.h 中定义了多国语言的枚举

```
typedef enum {
    LANG_CN      = 0,    // Chinese Simplified
    LANG_TW      = 1,    // Chinese Traditional
    LANG_EN      = 2,    // English
    LANG_JPN = 3,    // Japanese
    LANG_KR      = 4,    // Korean
    LANG_RS      = 5,    // Russian
    LANG_ITA = 6,    // Italian
    LANG_ERR
}LANGUAGE;
```

### 3. 修改 LANGUAGE 和 语言文件的对应关系

修改 newcdr/src/misc/cdrLang.cpp 中的构造函数 cdrLang

```
mLangFileTable.add(LANG_CN,      String8("/res/lang/zh-CN.bin") );
mLangFileTable.add(LANG_TW,      String8("/res/lang/zh-TW.bin") );
mLangFileTable.add(LANG_EN,      String8("/res/lang/en.bin") );
mLangFileTable.add(LANG_JPN,     String8("/res/lang/jpn.bin") );
mLangFileTable.add(LANG_KR,      String8("/res/lang/korean.bin") );
mLangFileTable.add(LANG_RS,      String8("/res/lang/russian.bin") );
mLangFileTable.add(LANG_ITA,     String8("/res/lang/italian.bin") );
```