MTK 智能穿戴入门篇

——疯壳·线下课程系列

Fengke-Team

2017/08/02

目录

官网地址：https://www.fengke.club/GeekMart/views/mall/goodsDetails.html?productId%3D33
配套书籍：https://www.fengke.club/GeekMart/views/mall/goodsDetails.html?productId%3D73
配套视频：http://www.fengke.club/GeekMart/su_fihsGbMhu.jsp
淘宝地址：http://shop115904315.taobao.com/
QQ 群：457586268

MTK 编程——新增 APP

在 MTK 系统中，所有的功能都是模块化管理，并且大多数功能都有自己的功能宏，和对应的源代码文件，我们把这种功能模块称之为应用，简称 APP（Application），比如闹钟、计算器、日历等都是一个 APP。在之前的例子中我们把自己的代码写在了 IdleCommon.c 文件中，这个文件属于待机功能模块。很显然我们把代码写在这个文件里面是不科学的，接下来我们就添加自己的功能模块，并建立自己的源文件和资源文件。

1、首先我们对平台做一些修改，在 make\FengKe2502C_WT_11C_GPRS.mak 文件中定义一个平台代码公共宏 __PLATFORM_PUBLIC__，把我们对平台修改的代码全部用这个宏包含。代码如下：

```
# ********************************************************************
# Consistent Feature Options
# ********************************************************************
COM_DEFS_FOR_MT6261RF_CUSTOM = MT6261RF_RF MT6261RF_CUSTOM
COM_DEFS_FOR_FENGKE2502C_WT_11C_LCM = COLOR_LCD FENGKE2502C_WT_11C_LCM TFT_MAINLCD
# ********************************************************************
# Include MODEM.mak
# ********************************************************************
include make\MODEM.mak

#-----------------------------------------------------------##
#---------- customer add begin-----------------------------##
CUSTOM_OPTION     +=   __PLATFORM_PUBLIC__     #平台公共宏

#---------- custom add end---------------------------------##
#-----------------------------------------------------------##

# ********************************************************************
# Project specified preprocessor definitions
# ********************************************************************
CUSTOM_OPTION     +=   __MSDC_NO_WRITE_PROTECT__
CUSTOM_OPTION     +=   __BT_SHRINK_SIZE__ __MMS_MEMORY_CARD_STORAGE_SUPPORT__
```

2、在 plutommi\mmi 文件夹下新建一个文件夹，命名为 CustomerApp，后面我们把自己开发的所有功能模块都放在这个文件夹下面。并创建 CustomerApp\HelloMTK\Src、CustomerApp\HelloMTK\Inc、CustomerApp\HelloMTK\Res 目录。在以上三个目录中分别新建 HelloMTK.c、HelloMTK.h、HelloMTK.res 三个文件。如下图所示：

3、 在 plutommi\mmi\Inc 目录下新建 MMI_features_switch_custom_app.h 文件。在后面的开发中，这个文件专门用于添加我们自己的 APP 功能宏，把这个文件包含到 plutommi\Customer\CustResource\FengKe2502C_WT_11C_MMI\MMI_features_switchFengKe2502C_WT_11C.h 文件中，MMI_features_switchFengKe2502C_WT_11C.h 代码如下：

```
*****************************************************************************/
#ifndef __MMI_FEATURES_SWITCH_H__
#define __MMI_FEATURES_SWITCH_H__

/*****************************************************************************
* Option Value Definition
*****************************************************************************/
#include "MMI_features_type.h"

#if defined(__PLATFORM_PUBLIC__)
#include "MMI_features_switch_custom_app.h"
#endif
```

4、 在 MMI_features_switch_custom_app.h 文件中定义宏开关，代码如下：

```
/*****************************************************************************
* 该文件用于放置自定义的功能宏
* herunping
*============================================================================
*****************************************************************************/
#ifndef __MMI_FEATURES_SWITCH_CUSTOM_APP_H__
#define __MMI_FEATURES_SWITCH_CUSTOM_APP_H__

/*
* hello MTK
*/
#define CFG_MMI_HELLO_MTK           (__ON__)

#endif /* __MMI_FEATURES_SWITCH_CUSTOM_APP_H__ */
```

在 MMI_features.h 文件末尾定义宏。代码如下：

```
#if (!defined(__MMI_FWUI_SLIM__))
    #ifndef WGUI_STATUS_ICON_SHOW_V_BAR
    #define WGUI_STATUS_ICON_SHOW_V_BAR
    #endif
#endif

#if defined(__PLATFORM_PUBLIC__)

/*hello MTK*/
#if defined(CFG_MMI_HELLO_MTK) && (CFG_MMI_HELLO_MTK != __OFF__)
    #define __MMI_HELLO_MTK__
#endif

#endif

#endif /* __MMI_FEATURES__ */
```

5、 在 make\plutommi\mmi_app\mmi_app.mak 文件末尾加载源文件和头文件，添加代码如下：

```
    SRC_LIST += plutommi\MMI\Idle\IdleSrc\IdleSwatch.c \
                plutommi\MMI\Swatch\SwatchSrc\Swatch.c
endif

# ------------------------------------------------------------------------------
#                  add custom app src and inc dir
# ------------------------------------------------------------------------------
ifneq ($(filter __MMI_HELLO_MTK__ , $(strip $(MODULE_DEFS))),)
   INC_DIR += plutommi\mmi\CustomerApp\HelloMTK\Inc
   SRC_LIST += plutommi\mmi\CustomerApp\HelloMTK\Src\HelloMTK.c
endif
```

6、 在 plutommi\mmi\Inc\Mmi_pluto_res_range_def.h 文件末尾加载资源文件，添加代码如下：

```
#if defined(__PLATFORM_PUBLIC__)
/*****************************************************************************************
    custom app
*****************************************************************************************/

#if defined(__MMI_HELLO_MTK__)
/*****************************************************************************
* hello MTK
*****************************************************************************/
MMI_RES_DECLARE(APP_HELLOMTK, 10, ".\\mmi\\CustomerApp\\HelloMTK\\Res\\")
#define HELLOMTK_BASE                           ((U16) GET_RESOURCE_BASE(APP_HELLOMTK))
#define HELLOMTK_BASE_MAX                       ((U16) GET_RESOURCE_MAX(APP_HELLOMTK))
#endif

#endif

#endif /* !defined(__COSMOS_MMI_PACKAGE__) */
```

7、 最后，把我们添加在 idle.res、MainMenuRes.res 中的资源移到 HelloMTK.res 中，把 idlecommon.c 中的源码移到 HelloMTK.c 文件中。代码如下：

HelloMTK.c 文件

```
#include "MMI_features.h"
#if defined(__MMI_HELLO_MTK__)
#include "HelloMTK.h"
#include "GlobalResDef.h"
#include "Mmi_frm_gprot.h"
#include "Gui_themes.h"

void mmi_my_mtk_func_exit(void)
{
}

void mmi_my_mtk_func(void)
{
    mmi_frm_scrn_enter(GRP_ID_ROOT, SCR_ID_MY_MTK_FUNC,
mmi_my_mtk_func_exit,mmi_my_mtk_func,MMI_FRM_FULL_SCRN);

    gui_set_text_color(UI_COLOR_WHITE);/*设置字符打印颜色*/

    gdi_image_draw_id(0, 0, IMG_ID_HELLO_MTK);/*显示图片*/

    gui_move_text_cursor(10, 15);/*设置字符打印坐标*/

    gui_set_font(&MMI_medium_font);/*设置字符显示的字体*/
```

```
        gui_print_text((UI_string_type)GetString(STR_ID_HELLO_MTK));/*打印字符*/

        /*刷新屏幕*/
        gui_BLT_double_buffer(0, 0, UI_DEVICE_WIDTH,UI_DEVICE_HEIGHT);

        /*注册右软键事件*/
        SetKeyHandler(mmi_frm_scrn_close_active_id, KEY_RSK, KEY_EVENT_UP);
    }

    void mmi_highlight_my_mtk(void)
    {
        SetLeftSoftkeyFunction(mmi_my_mtk_func, KEY_EVENT_UP);
        SetKeyHandler(mmi_frm_scrn_close_active_id, KEY_RSK, KEY_EVENT_UP);
    }
#endif
```

HelloMTK.h 文件

```
    #ifndef __HELLOMTK_H__
    #define __HELLOMTK_H__

    #include "MMI_features.h"

    #if defined(__MMI_HELLO_MTK__)
    #include "mmi_rp_app_hellomtk_def.h"

    extern void mmi_my_mtk_func_exit(void);
    extern void mmi_my_mtk_func(void);
    extern void mmi_highlight_my_mtk(void);

    #endif
#endif /*__HELLOMTK_H__*/
```

HelloMTK.res 文件

```
    #include "mmi_features.h"
    #include "custresdef.h"

    #if defined(__MMI_HELLO_MTK__)

    <?xml version="1.0" encoding="UTF-8"?>

    <APP id="APP_HELLOMTK">

        <!--Include Area-->
        <INCLUDE file="GlobalResDef.h,SettingResDef.h"/>
```

```
        <!----------String Resource Area---------->
        <STRING id="STR_ID_HELLO_MTK">"Hello MTK !"</STRING>

        <!----------Image Resource Area---------->
        <IMAGE id="IMG_ID_HELLO_MTK">CUST_IMG_PATH"\\\\MainLCD\\\\IdleScreen\\\\Wallpaper\\\\WALL01.jpg"</IMAGE>

        <!----------Menu Resource Area---------->
        /*添加菜单 ID*/
        <MENU id="MENU_MY_MTK_ID" type="APP_MAIN" str="STR_ID_HELLO_MTK" img="IMG_GLOBAL_OK"
highlight="mmi_highlight_my_mtk"/>

        <!----------Timer Resource Area---------->


        <!----------SCREEN Resource---------->
        <SCREEN id="SCR_ID_MY_MTK_FUNC"/>

    </APP>
#endif
```

最后依次执行 make new，make　gen_modis，重新运行模拟器，运行结果虽然没变，但我们把代码模块化了，方便以后的维护、移植。这只是一个很小的例子，如果是一个上万行代码的应用，这种方式就能体现出极大的优势。另外 audio.zip、images.zip 中的资源文件我们也可以建立文件夹单独列出来管理，请读者自己尝试，在后面的 app 实例开发中我们会采用这种方法。