

MTK 智能穿戴入门篇

——疯壳·线下课程系列

Fengke-Team

2017/08/02

目录

一、MTK 开发环境搭建.....	3
二、MTK 平台框架.....	错误! 未定义书签。
三、MTK 编译指令.....	错误! 未定义书签。
四、MTK 编程入门.....	错误! 未定义书签。
五、资源.....	错误! 未定义书签。
六、新增 APP.....	错误! 未定义书签。

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>

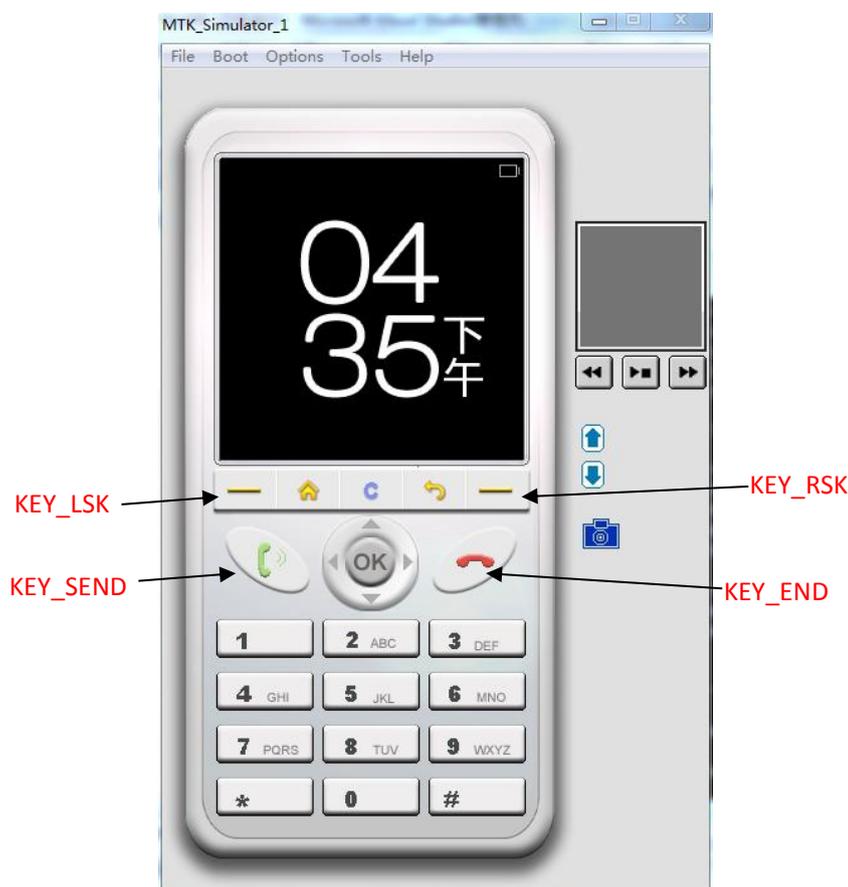
配套视频: http://www.fengke.club/GeekMart/su_fihsGbMhu.jsp

淘宝地址: <http://shop115904315.taobao.com/>

QQ 群: 457586268

第一个程序

首先, 建立 source insight 工程 (使用方法, 请自己查阅资料)。然后在 CMD 命令行窗口切换到工程源码根目录, 执行 `make FengKe2502C_11C GPRS new` 生成二进制 bin 文件, 然后再执行 `make -debug FengKe2502C_11C GPRS gen_modis` 生成 VS2008 工程文件。再用 VS2008 打开 `MoDIS_VC9\MoDIS.sln` 文件, 编译生成模拟器。模拟器运行界面如下图:



图中红色字符标注了几个常用的按键值, 在后面的编码过程中会经常提到。其中 `KEY_LSK` 叫确认键, `KEY_RSK` 叫返回键。 `KEY_SEND` 系统默认用于拨号按键, `KEY_END` 称之为退出按键, 在任何界面都可以返回 `idle` 界面。 `idle` 界面也叫待机界面, 即上图中看到的界面。

按键功能注册

每一个按键在不同的屏幕中都有不同的功能，这些功能都可以自己定义。常见的按键定义函数有如下几个：

1、void SetKeyHandler (FuncPtr funcPtr,U16 keyCode,U16 keyType)

funcPtr : 按键执行的功能函数指针。

keyCode: 按键值。取值范围见 GlobalConstants.h 文件中的枚举 mmi_keypads_enum

keyType: 按键方式，取值范围见 GlobalConstants.h 文件中的枚举 mmi_key_types_enum

比如：SetKeyHandler (mmi_scr_locker_launch, KEY_RSK, KEY_EVENT_UP); 含义为按住 KEY_RSK (右软键)，弹起的时候执行 mmi_scr_locker_launch 函数。如果 KEY_EVENT_UP 改为 KEY_EVENT_DOWN，则表示按下 KEY_RSK 键立即执行 mmi_scr_locker_launch 函数。

SetKeyHandler 实际上是对 mmi_frm_set_key_handler 的封装，另外还有一些按键注册函数，比如 SetKeyDownHandler (注册按下事件)、SetKeyUpHandler (注册弹起事件) 都是对 mmi_frm_set_key_handler 函数的封装，最终执行按键注册都会在 mmi_frm_set_key_handler 函数中进行。

2、void SetLeftSoftkeyFunction(void (*f) (void), MMI_key_event_type k)

f: 按键执行的功能函数指针。

k: 按键方式，取值范围见 GlobalConstants.h 文件中的枚举 mmi_key_types_enum

这个函数只能用于注册左软键按键功能，实际上等同于 SetKeyHandler (FuncPtr funcPtr, KEY_LSK, U16 keyType)。另外还有一个专用于注册右软键功能的函数——SetRightSoftkeyFunction。已经一个注册 KEY_CSK 的函数——SetCenterSoftkeyFunction。

3、void SetGroupKeyHandler(FuncPtr funcPtr, PU16 keyCodes, U8 len, U16 keyType)

funcPtr : 按键执行的功能函数指针。

keyCodes: 按键值数组。取值范围见 GlobalConstants.h 文件中的枚举 mmi_keypads_enum

keyType: 按键方式，取值范围见 GlobalConstants.h 文件中的枚举 mmi_key_types_enum

这个函数用于注册多个按键执行同一个函数时使用，比如在 idle 界面按 0~9 个数字时，都会进入拨号界面。这个函数使用的比较少，在穿戴设备开发中，基本不会用到。

接下来，我们在代码中实现按键注册功能。在 source insight 工程中打开 IdleCommon.c 文件，在 mmi_idle_set_handler 函数上面添加一个我们自己的函数，函数名为 mmi_my_mtk_func，简单实现一个显示 “hello MTK !” 字符的功能，代码如下：

```

void mmi_my_mtk_func(void)
{
    gui_set_text_color(UI_COLOR_WHITE); /*设置字符打印颜色*/
    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);
}

/*****
 * FUNCTION
 * mmi_idle_set_handler
 * DESCRIPTION
 * This function sets the default handler according to the capability of the
 * idle object.
 * PARAMETERS
 * obj          : [IN]      Idle object
 * RETURNS
 * void
 *****/
void mmi_idle_set_handler(mmi_idle_obj_struct *obj)
{

```

然后在 `mmi_idle_set_handler` 函数的最后一行添加代码，注册 `KEY_LSK` 按键的弹起功能执行函数 `mmi_my_mtk_func`。

```

    SetKeyHandler(mmi_my_mtk_func, KEY_LSK, KEY_EVENT_UP);
} ? end mmi_idle_set_handler ?

```

运行模拟器，鼠标单击 左软键 (`KEY_LSK`，位置如图红色框中)，是否发现屏幕上显示有“hello MTK!”



这是一个很简单的函数，代码中都有详细的注释。MTK 的编码规范完全遵循 C 语言的编码规范，本书的读者也必须有一定的 C 语言基础。下面简单介绍几个 MTK 的函数接口。

1、gui_set_text_color

设置字符显示的颜色。可以调用系统定义的一些颜色常量，比如 UI_COLOR_RED、UI_COLOR_BLACK 等。也可以使用 gui_color(U8 r, U8 g, U8 b)自己生成，比如 gui_color(255,0,0)为红色。

2、gui_move_text_cursor

设置字符显示的坐标。屏幕上有一个坐标系，以左上角为原点(0, 0)，右下角坐标为屏幕尺寸(LCD_WIDTH, LCD_HEIGHT)。宽为 X 轴，高为 Y 轴。

3、gui_set_font

设置字符打印的字体。MTK 的字体跟字库有关，此处我们不讲解斜体、下划线等特效，只讲解字体大小。系统中定义有一些字体全局变量，常用的有 MMI_small_font、MMI_medium_font、MMI_large_font。

4、gui_print_text

在屏幕上显示字符。是否发现字符常量"Hello MTK !"前面有一个大写的 L? 因为 MTK 的手机屏幕上能够显示的所有字符，不管是英文、中文还是日文，都必须是 UCS2 编码，而 L 的作用就是把字符常量强制转换成 UCS2 编码。特别注意，L 只能转换字符常量，不能转换变量。有兴趣的读者可以试一下不加 L 打印出来的效果。

5、gui_BLT_double_buffer

屏幕上显示的内容只要有变更，都必须调用这个函数刷新屏幕，否则看不到任何效果。屏幕中的四个参数为要刷新的矩形区域，这个矩形区域必须包含你绘画的内容，此处为全屏刷新。当然也可以不使用全屏刷新，此函数中字符显示的坐标为(100, 150)，假设字符串的宽度为 50，高度为 20，则改 gui_BLT_double_buffer(100, 150, 100+50, 15+20) 也是同样的效果。有兴趣的读者可以改为 gui_BLT_double_buffer(0, 0, 50,50)，试试看能否显示出"Hello MTK !"。

特别说明：

定义屏幕宽度的常量或全局变量有：UI_DEVICE_WIDTH、LCD_WIDTH、UI_device_width

定义屏幕高度的常量或全局变量有：UI_DEVICE_HEIGHT、LCD_HEIGHT、UI_device_height

