

# 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](http://www.fengke.club/GeekMart/su_fihsGbMhu.jsp)

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

QQ 群: 457586268

## 编译指令

在开始进入编码之前, 我们简要介绍一下常用的代码编译指令, 在接下来的编码过程中, 会经常用到。对于刚接触 MTK 开发的人, 可以把这一节的内容当作词典来用, 不记得就查一查, 没必要死记。在后面的开发中, 我也会提示采用什么样的指令编译代码。

编译指令通用格式:

```
make [-debug] project project_attr command
```

指令格式说明:

- 1、**make**: 这是一个批处理的文件名称, 在工程源码的根目录下, 有一个 **make.bat** 的文件。在 **make** 后面的字段实际上都是给批处理传的参数, 这个名称一般不允许更改, 我们把它当做一个固定字段使用。
- 2、**-debug**: 这个参数只用于编译模拟器。
- 3、**project**: 这是我们要编译的项目名称。在工程源码的根目录下, 有一个 **make** 文件夹, 在这个文件夹里面有一个 **Verno\_XXXXXX.bld** 文件, 比如 **Verno\_FengKe2502C\_11C.bld**。这个文件是项目的版本控制文件, 在 **Verno\_** 后面的 **FengKe2502C\_11C** 就是项目名称。在 **make** 文件夹下面, 有可能存在多个这样的文件, 那就说明在这个工程源码中, 包含多个项目。
- 4、**project\_attr**: 这个字段为项目的属性, 常用取值有 **gprs**、**gsm**、**none**。其中 **gprs** 代码项目支持 sim 卡, 可以打电话、发短信, 还能上网; **gsm** 表示该项目也支持 sim 卡, 可以打电话, 可以发短信, 但无法上网, 在这样的项目中, 我们无法进行 **socket** 编程; **none** 表示项目不支持 sim 卡。那么如何确定项目的属性呢? 在上一条 **project** 的介绍中, 我们提到了 **make** 目录下的项目版本控制文件——**Verno\_FengKe2502C\_11C.bld**。与之对应的还有一个项目配置文件——**FengKe2502C\_11C\_XXXX.mak**, 比如 **FengKe2502C\_11C\_GPRS.mak**。那么这个项目的属性就是 **gprs**。每一个项目在 **make** 文件夹中都必须包含这两个文件。
- 5、**command**: 这是我们要用到的编译命令, 采用什么样的方式生成二进制 **bin** 文件。取值及说明见如下表格:

new	这个指令耗时最长, 它会重新编译所有模块, 每一个项目第一次编译都要使用这个指令, 在修改了项目配置, 或者新增文件时都要使用这个指令重新编译整个项目。
Remake (可简写为 r)	这个指令只是简单的重新编译链接有改动的部分。它不检查依赖关系, 不扫描资源, 只扫描代码的改变, 有改变的重编, 资源和无改变的代码不编。如果 <b>r</b> 后面没有其他字段, 则重新编译所有模块, 如果有其他模块字段, 则重新编译单个模块或多个模块。比如: <b>r mmi_app</b> 则只重新编译 <b>mmi_app</b> 模块, <b>r mmi_app mmi_framework</b> 则只重新编译两个模块。至于这些模块名称从哪里查找, 在后面会有介绍。
Updata (可以简写为)	这个指令会扫描工程中文件和库的依赖关系, 若依赖关系有变化

u)	会建立新的依赖关系，随后根据新的依赖关系重新编译链接有改动的部分。它跟 remake 的用法类似。可以更新所有模块，也可以更新指定的模块，比如 u mmi_app, u mmi_app mmi_framework。但是 r mmiresource 不会重新生成资源，u mmiresource 会重新生成资源，相当于 resgen 和 r mmiresource 的合并。
Resgen	编译资源。只要修改了.res 文件，都要用这个指令编译
Gen_modis	生成模拟器工程文件，可用 Visual Studio 2008 生成模拟器，调试 mmi 部分的代码。

编译指令举例：

```

make FengKe2502C_11C GPRS new
make FengKe2502C_11C GPRS resgen
make FengKe2502C_11C GPRS r
make FengKe2502C_11C GPRS r mmiresource
make FengKe2502C_11C GPRS u mmiresource (等同于 make resgen 和 make r mmiresource)
make FengKe2502C_11C GPRS r mmi_app mmi_framework
make FengKe2502C_11C GPRS u mmi_app mmi_framework

make -debug FengKe2502C_11C GPRS gen_modis
make -debug FengKe2502C_11C GPRS codegen_modis
make -debug FengKe2502C_11C GPRS new_modis (等同于 make gen_modis 和 make codegen_modis)
    
```

在后面的编码过程中，我会省略 project project\_attr 用 make command 的格式提示大家用什么样的命令来编译代码。比如 make new 就等同于 make FengKe2502C\_11C GPRS new，如果你的代码之前有使用该命令编译过，源码根目录中有 FengKe2502C\_11C\_gprs.log 文件，那直接使用 make new 也是可以编译的，同理直接使用 make r 或者 make r mmi\_app 都是可以直接编译。但是对于新手而言，不推荐使用这种简写的指令，防止忘记完整的指令格式。