用cmd切换到\*\*\*.obj目录下   
命令 armar -create \*\*\*.lib \*\*\*.obj   
其中\*\*\*.obj是由\*\*\*.c文件编译后生成的。   
使用的话在\*\*\*.mak里面加上   
COMPOBJS += 对应的路径   
记得将lis文件里的.c删除

===========================================================================

方法0：  
MTK平台如何生存LIB库文件：  
1.       make 文件夹建立需要生成库的mylib，放入c和h文件，单独编译mylib模块  
2.       make文件中COMPLIST加入mylib  
3.       在REL\_MMI\_GPRS.mak中CUS\_REL\_MTK\_COMP加入mylib  
4.       在mtklib文件夹里面放mylib，就ok啦。

方法一：  
MTK生成库(ARM) 过程  
第一步，生成真机的obj文件  
第二步，找到相关的obj文件 D:\\LongCheer\\build\\LC6225\_GEMINI\\gprs\\MT6225o\\mmi\_app  
第三步，把相关的obj文件拷贝到一个目录下面 D:\\lib\_temp  
第四步，打开命令窗口，输入命令 armar -r temp.lib \*.obj 生成 temp.lib 库  
第五步，输入命令armar -tv temp.lib 可以看到加入的库  
也可以这样armar -r xxx.lib x1.obj x2.obj。

方法二：  
在option.mak 中单独添加一个模块  
COMPLIST += tmptmp  
在make中添加tmptmp目录及其那是个文件。  
会在gprs\\MT6225o\\lib目录中有tmptmp.lib

本文来自：我爱研发网(52RD.com) - R&D大本营  
详细出处：<http://www.52rd.com/Blog/Detail_RD.Blog_linweishang_22952.html>

=========================================================================

例：编译完之后--

1、在。。\build\XXX\gprs\MT6223Po下新建一个a文件夹

2、把需要的obj文件放到lexue目录下

3、修改..\make\XXX\_MMI\_GPRS.mak

在下面这三行后面

ifeq ($(findstring FMT\_NOT\_PRESENT,$(CUSTOM\_OPTION)),)

  CUS\_REL\_SRC\_COMP += fmt

endif

添加

CUS\_REL\_SRC\_COMP += a

4、执行命令

make custom=XXX gprs remake

5、找到。。\build\XXX\gprs\MT6223Po\lib

这时a.lib就已经生成好了

6、注意：生成完之后，把第3步聚的东西修改回来，同时把a.lib放到根目录中的a目录下

7、在确定lib生成没问题了，把第一步的a文件夹删除掉

----------------------------------------------------------

MTK生成lib库(Win32)

1.      首先要确定你的电脑上安装了visual c++ 6.0（其他版本没试过），假设我们将vc安装在了C:\Program Files\Microsoft Visual Studio

2.      在桌面上右键点击“我的电脑”---属性---高级---环境变量

3.      在系统变量里找到“path”，双击编辑

4.      在变量值的最后添加“;C:\Program Files\Microsoft Visual Studio\VC98\Bin;C:\Program Files\Microsoft Visual Studio\Common\MSDev98\Bin”

5.      点击确定

6.      新建一个环境变量，变量名：“INCLUDE”，变量值：“C:\Program Files\Microsoft Visual Studio\VC98\INCLUDE”

7.      再新建一个环境变量，变量名：“LIB”，变量值：“C:\Program Files\Microsoft Visual Studio\VC98\LIB”

8.      确定保存

然后就可以使用VC的link命令来打win32 的lib包了

举个例子

将使用vc编译出来的obj文件(注意不是arm编译出来的obj)，例如：..\MoDIS\mmi\_app\Debug中你需要的obj拷贝到D:\obj下用命令行模式进入该目录，执行 link /lib /out:mylib.lib \*.obj就可以生成一个mylib.lib的目标文件

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

另外一个关于编lib库的方法

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

若我们需要加入zlib这么一个模块(zlib包含了很多程序都用到的压缩和解压函数)，我们可以按如下步骤进行。

1. 把zlib的源代码包拷贝到你的MTK软件系统根目录下，这样所有的zlib代码都在zlib目录或zlib-1.2.3目录下(用哪个目录名由你的喜好来确定)，在此为简单起见就放在zlib目录。

2. 在make目录下新增一个zlib的目录(最好和根目录下的目录名一样)，增加四个文件，分别是zlib.def、zlib.inc、zlib.lis、zlib.pth。zlib.def文件只要加上APCS\_INTWORK就可以了，其他三个文件中加上源文件及其目录、头文件目录即可。

3. 在REL\_CR\_MMI\_<project>.mak这个文件的合适位置加上如下语句。

CUS\_REL\_SRC\_COMP += zlib

4. 把这个工程remake一下，若没有错误，zlib模块就成功加上去了。其他模块就可以调用zlib提供的压缩和解压缩函数了。

若我们需要加入一个没有源代码的第三方库，譬如加入wifi，我们可以按如下步骤进行。

1. 把wifi的所有第三方库文件拷贝到一个新建的wifi目录下。

2. 在Option.mak这个文件的合适位置加上如下语句。

ifeq ($(strip $(WIFI\_SUPPORT)),WIFI\_LIB)

COMPOBJS += wifi\sslplus.lib

CUS\_REL\_OBJ\_LIST += wifi\sslplus.lib

COMPOBJS += wifi\sb.lib

CUS\_REL\_OBJ\_LIST += wifi\sb.lib

endif

3. 把这个工程remake一下，若没有错误，wifi库就成功加上去了。其他模块就可以调用wifi提供的功能了。