嵌入式产品是需要进行资源管理的，它不像是开发安卓APP那种，官方IDE会帮助将各种各样自动打包成一体，开发者只需要去关注基础的业务需求开发，那么嵌入式Gui也同样的需要各种各样的打包工具，用以支持项目和业务需求开发所用

**图片资源打包工具（必须）:**

通常ui设计会为项目制造好丰富多彩的ui图片包，这些图片资源多是bmp, png, jpg格式的，我们需要对其进行加工处理，最终生产出一个bin格式的二进制文件，这个文件需要包含所用到的资源的所有有效信息，并且能优秀的按设备平台打包成设备快速使用的数据格式，这个也需要注意。

通常Gui内部只会关系像素以及像素流，形如bmp格式的数据，我们需要提取出里面的bitmap, width, height, format等信息，例如: 设备平台使用到的是RGB565格式的像素数据，但是一般PC上的bmp都是RGB888的格式，那么打包工具在处理该类型图片时，就需要将RGB888的格式一个个像素点计算然后转换成RGB565的格式，然后再按顺序一个个像素点拼装起来，此外在这里大小端序也是需要考虑的事情，因为我们的像素点通常都不是1字节数据，那么就会存在翻转的可能性，那么也需要一并考虑进去，这样子以后，我们绘制一张不带透明度的图片到屏幕上最终就不需要一个个像素点处理了，甚至我们都不需要解析像素点，而是一行一行的像素点直接进行memcpy就行了，到了这一步，这部分就适配好了，至此你可以使用DMA或者如果设备支持，使用DMA-2D进行硬件加速，加速你图片的copy和blend的过程，，，扩充升级，当你完整的处理好了整个像素点流格式，并且得到了一个像素点数组的时候，你可以考虑使用压缩算法，去降低对外部的消耗，这里推荐lz4压缩，目前来说是比较通用的压缩图片bitmap的选择，那么对应的你需要设备上实现一个缓存管理器，去维护加载的图片，及时的释放，因为如果你使用了压缩，那么从外存读入内存的工程中，你就需要对其进行解压缩还原出以前的bitmap。那么那些形如format，width，height等资源，你同样的也需要进行一次翻译工作，通常这部分要生成C的代码参与工程编译到ROM中，因为这部分内容的空间占用不多，但是可能涉及到频繁的访问，和bitmap一并放到bin文件中会导致设备的性能损失

其他数据格式，如png和jpg格式，这些格式都是通用的图片压缩格式，它里面是根据对应的压缩算法将原来的BMP经过压缩得到的数据，我们一般需要在PC上先解压缩取到bitmap，然后取像素点翻译成设备直接可以使用的RGB565和ARGB8565的格式，当然如果你给Gui移植了PNG的解码器也行，那么这个翻译的工作就放在设备上去了，如果你有了解码器，无论是软件解码器还是硬件解码器，此时你只需要将整个图片以文件的形式完整的打包好即可，无需对里面做任何的修改，这两种模式各有好坏，前者降低设备的运算，加大设备的外部存储（你转成bitmap再进行lz4压缩，通常压缩效率低于PNG或者JPG原生文件压缩很多），后者的话就加大CPU的运算，因为你需要本地解码和翻译的工作，如果你有硬解码器，那就完美优化了不足，因为硬解解码器通常会帮你解码顺便翻译到当前平台的像素点格式，你得到的数据是可以直接使用的

图片解码器的实现就看个人的实际储备和喜好，你可以搭建一个PC上的C工程，主要用到的是stdio去一个个处理文件和解析文件，最后一步步打包到新的bin之中，你可能需要用到一些库，最终的工程可以在PC上生成一个exe，如果完成度足够高，还可以携带界面给用户选择参数，这种一般就推荐Qt去做前端了  
 此外个人推荐的就是py脚本处理了，一般几百行就能完成这个打包工具，给定脚本的一个复杂的图片路径包，自动遍历所有图片文件一个个取出，然后解析翻译最终生成bytes流写入到bin文件，参考：scui\_image\_parser.py

**文字资源打包工具（可选）：**

这个部分个人还没深入研究，我选取的是一套简单的方案，使用lv\_font\_conv工具去进行的文字资源打包，然后移植改造lv\_font\_loader.c就可以了，再额外复杂的就是考虑移植FreeType解析器，让设备平台直接从FreeType中去字形信息和相关的内容，这种会加大设备平台的CPU开销，有选择性的考虑

**布局信息打包工具（可选）：**

这部分就是看Gui自己的需求了，比如说lvgl就不需要这个，因为实现是代码，如果你有类似与json这种格式的文件去描述ui的布局信息这类似的，且Gui能简要的去处理静态布局信息并且生成的话，你可以使用这个，通常是将ui文件伴随的json解析打包成一个大的布局信息的C代码表，将其和工程一起编译这样子可以减少ui界面的代码量和编码量，这个看Gui框架自行考虑，不是非必要的，可以节省一部分代码空间，只用于静态布局较多的ui产品需求中

**多国语资源打包工具（可选）：**

这个就比较简单了，输入一个xlsx的多国语表格，生成出一个c语言的二维数组即可，写法比较多样了，参考：scui\_multi\_lang.py