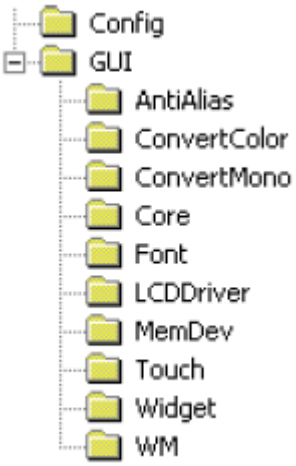
第2章 入门指南

这一章提供一个在你的目标系统上设置和配置μC/GUI的基本处理过程的概述。同时也包括了一个简单的范例程序。

请注意，大多数主题在后面的章节会有更详细的描述。在你开始更复杂的编程之前，你很有必要参阅本手册的其它部分。

2.1推荐的结构

使μC/GUI 和你的应用文件分离，这是我们推荐的。在工程文件的“root”目录的 GUI 子目录下保留所有的程序文件（包括头文件），这是一个好的习惯。目录结构应该和下图相似。这种习惯有一个好处，就是很容易升级更新版本的μC/GUI，只需要替换 GUI 目录就可以。



子目录

下表显示了 GUI 所有子目录的内容

目　录	内　　容
Config	配置文件
GUI/AntiAlias	抗锯齿支持 *
GUI/ConvertMono	用于 B/W（黑白两色）及灰度显示的色彩转换程序
GUI/ConvertColor	用于彩色显示的色彩转换的程序
GUI/Core	μC/GUI 内核文件
GUI/Font	字体文件
GUI/LCDDriver	LCD 驱动
GUI/Mendev	存储器件支持 *
GUI/Touch	触摸屏支持 *
GUI/Widget	视窗控件库 *
GUI/WM	视窗管理器 *

（带“*”标志的为可选项）

“Include”目录

确认你的 Include 路径包括有以下目录（包括的先后顺序并不重要）：

- Config
- GUI/Core
- GUI/Widget（如果使用视窗控件库）
- GUI/WM（如果使用视窗管理器）

警告：你必须确认你在每个文件中只使用了一个版本的μC/GUI

2.2 向目标程序加入μC/GUI

你主要是在这两者之间做一个选择，一是将你要在你的工程中使用的源文件包括进来，然后进行编译和连接；或者建立一个库并连接这个库文件。如果你的链接工具支持“智能化”连接（仅仅连接那些使用到的模块而不是那些没有使用到的模块），那么就完全没有必要建立建立一个库，因为只是要求将函数和数据结构进行连接。如果你的工具链接不支持“智能化”连接，建立一个库就很有意义了，否则如果将每样东西都要进行连接的话，程序会变得非常大。对于一些 CPU 来说，我们能提供有效的范例工程帮助你开始使用。

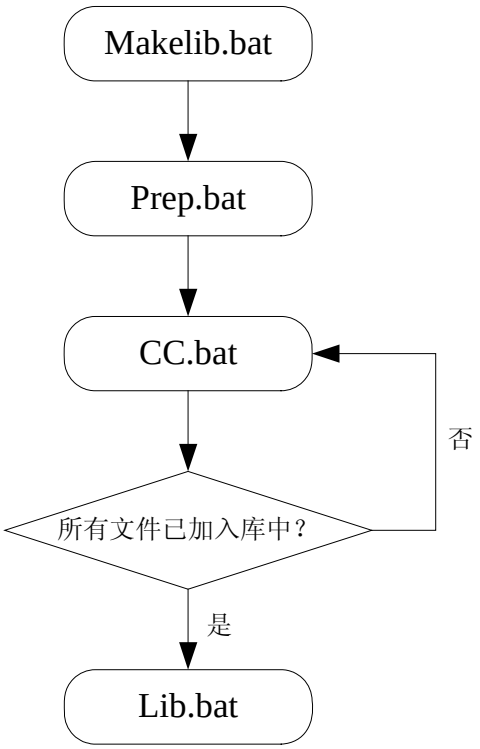
2.3 建立一个库

从源程序建立一个库是一个简单的流程。第一步是拷贝批处理文件（位于“Sample\Makelib”目录下面）到根目录下。然后，做一些必要的修改。总共有四个批处理文件需要拷贝，如下表描述的那样。主文件“Makelib.bat”在所有的系统中都是一样的，所以无需修改。在你的目标系统上建立一个库，正常情况下你需要对其它三个比较小的文件做一些微小的改动。最后执行“Makelib.bat”文件建立库。批处理文件假定你的 GUI 和配置子目录已经如前面所推荐的那样建立起来了。

文　件	说　　明
Makelib.bat	主批处理文件，不需要修改
Prep.bat	由 Makelib.bat 调用，建立用于链接工具的工作环境
CC.bat	由 Makelib.bat 调用，对库所用到的文件进行处理，为这些目标文件建立一个列表，该列表在下一步中由 lib.bat 中的
lib.bat	由 Makelib.bat 调用，将列表中的目标文件置入一个库当中

建立库的流程如下图所示。Makelib.bat 文件首先调用 Prep.bat 准备用于链接工具的环境。然后调用 CC.bat 处理库当中所包括的每一个文件，做完这些工作需要一些时间。CC.bat

将这些目录文件加入一个列表，这个列表是 lib.bat 要使用的。当所有加入到库当中的文件已经写入列表后，Makelib.bat6 调用 lib.bat，使用一个库管理工具将列表中的目标文件置入一个活动的库当中。



我们假设一个微软编译器已经安装到它的默认位置。如果所有的批处理文件都拷贝到根目录（GUI 的上一级目录），并且不作任何修改，将会产生一个用于μC/GUI 仿真的仿真库。无论如何，要建立一个目标库的话，必须要对 Prep.bat、CC.bat 和 lib.bat 三个文件进行修改。

2.4 将μC/GUI的“C”文件加入工程中

通常说法，你需要加入μC/GUI 的核心“C”文件，LCD 驱动，你显示屏所使用的字体文件及其它你定制可选择的模块：

- 目录 GUI\Core、GUI\ConvertColor 及 GUI\ConvertMono 下的所有“C”文件
- 你的显示屏用到的字体（位于目录 GUI\Font 下）

附加的软件包

如果你的显示屏使用附加的可选的模块，你必须也要包括相关的“C”文件

2.5 配置μC/GUI

配置目录应该包含与你的要求相匹配的配置文件。文件 LCDConf.h 通常包含所有的需要的定义，使你能够为你的 LCD 使用μC/GUI，这是开始配置μC/GUI 的主要任务。了解更多的细节，请参阅第 20 章“底层配置”。

如果因为你没有选择正确的显示方案或选择了错误的 LCD 控制器而导致μC/GUI 没有正确配置，LCD 可能不会显示任何东西，或者显示些不是你所期望的内容。因此，要注意修改你所需要的 LCDConf.h。

配置宏的类型

下面是一些配置宏的类型：

二进制开关“B”

这个开关的数值是“0”或“1”，“0”表示不激活，而“1”表示激活（除了“0”以外的数值都可以激活，但是使用“1” 使配置文件更易于阅读）。这些开关能够启用或禁止某一个功能或行为。开关是配置宏中最简单的格式。

数值“N”

数值有代码中某些地方使用，以替代数值常量。在 LCD 配置方案中有一个典型的例子。

选择开关“S”

选择开关用于从多个选项中选择一项（只能选中一项）。典型的例子是用于所使用的 LCD 控制器的选择，选择的数值指示调用相应源代码（相应的 LCD 驱动）产生目标代码。

别名“A”

一个类似于简单的文本替代这样操作的宏。一个典型例子是定义 U8，预处理程序会用“unsigned char”代替“U8”。

函数替换“F”

该宏基本被视为一个正常的函数，尽管有某些应用上的限制，宏依旧被放入代码当中，就象文本代换的例子一样。函数替换主要用于给一个高度依赖硬件的模块增加一些特殊的函

数（例如 LCD 的访问），这类宏通常使用括弧（与可选择参数一起）来声明。

2.6 初始化µC/GUI

程序 GUI_Init()初始化 LCD 和µC/GUI 的内部数据结构，在其它µC/GUI 函数运行之前必须被调用。这通过将下面一行放入你的程序序列的开始来做到：

```
GUI_Init();
```

如果忽略了这个调用，整个图形系统将不会得到初始化，从而无法准备下一步的动作。

2.7 在目标硬件上使用µC/GUI

下面所陈述是只是我们使用µC/GUI 进行编程的一些基本的步骤要点。这些步骤更详细的解释在以后的章节介绍。

第一步：定制µC/GUI

通常第一步是通过修改头文件 LcdConf.h 来定制µC/GUI。你必须定义一些基本数据类型（U8，U16 等），及有关显示方案和所使用的 LCD 控制器的开关配置。

第二步：定义访问地址和访问规则

对于使用存储器映象的 LCD，仅仅需要在 LcdConf.h 中定义访问地址。

对于端口/缓冲的 LCD，必须定义接口程序，在 Samples\Lcd_x 目录下，或是在我们 Web 站点的下载区的中，有一些所需的接口程序的范例代码可供参考。

第三步：编译、连接和测试范例程序

µC/GUI 带有一些单任务和多任务环境下的范例程序，编译、连接和测试这些范例程序，直到你感觉已经熟悉它们了。

第四步：修改范例程序

对范例程序进行简单的修改，增加些额外的命令，诸如在显示时显示不同尺寸的文字，显示一条线等等。

第五步：多任务应用：适应你的操作系统（如果需要的话）

如果多任务允许同时访问显示器，则宏 GUI_MAXTASK 和 GUI_OS 与文件 GUITask.cg 一道开始运行。更详细的内容及范例程序的修改请参考第 21 章：高层次配置。

第六步：使用μC/GUI 编写你的应用程序

到现在，你应该对如何使用μC/GUI 应该有一个清楚的了解。考虑如何去构建你的应用要求的程序，通过调用适当的程序来使用μC/GUI。参考本手册后面相关的章节，这些章节讨论特殊的μC/GUI 函数和配置有效的宏。

2.8 “Hello World”范例程序

在早些时候，一个“Hello World”程序被做为 C 语言编程的入门程序，因为它本质上是一个能写出的最简单的程序。μC/GUI 的“Hello World”程序的名称是 Hello.c，如下所示。在μC/GUI 所带的范例中的它的名称为 Basic_HelloWorld.c。

该程序的目的是在显示器的左上角写“Hello World”，为了能实现这个功能，应用硬件，LCD 和 GUI 必须首先要初始化。μC/GUI 的初始化通过在程序开始调用 GUI_Init()来实现，就象先前所描述的那样。在本程序中，我们假设应用硬件的初始化已经完成。

```
/*----------------------------------------------------------------------
文件:       BASIC_HelloWorld.c
目的:       绘制“Hello world”的简单范例
----------------------------------------------------------------------*/

#include "GUI.H"

/*************************************************************************
*                                主函数                                  *
*************************************************************************/

void main(void)
{
/* 要做的事：确认硬件首先初始化了！ */
    GUI_Init();
    GUI_DispString("Hello world!");
    while(1);
```

```
}
```

给“Hello Word”程序增加功能

我们的小程序能做的工作实在太少，现在我们对它扩展一点功能：在显示“Hello World”后，我们希望程序开始计数以估计能够获得多快的输出速度（至LCD）。我们在主程序末尾的仅仅增加一点点代码进行循环，本质上是调用一个显示十进制形态数值的函数。

```
/*-----------------------------------------------------------------------
文件:       BASIC_HelloWorld1.c
目的:       绘制“Hello world”的简单范例
-----------------------------------------------------------------------*/

#include "GUI.H"

/*************************************************************************
*                                  主函数                                *
*************************************************************************/

void main(void)
{
    int i=0;
/* 要做的事：确认硬件首先初始化了！ */
    GUI_Init();
    GUI_DispString("Hello world!");
    while(1)
    {
        GUI_DispDecAt( i++, 20,20,4);
        if(i>9999) i=0;
    }
}
```