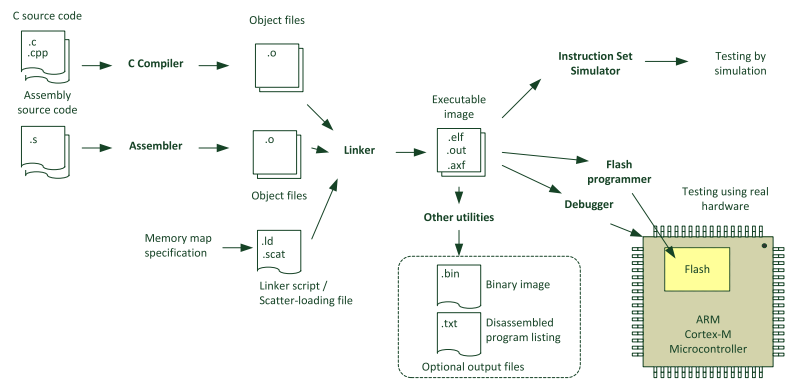
# Introduction to Embedded Software Development

## 准备工作：文档

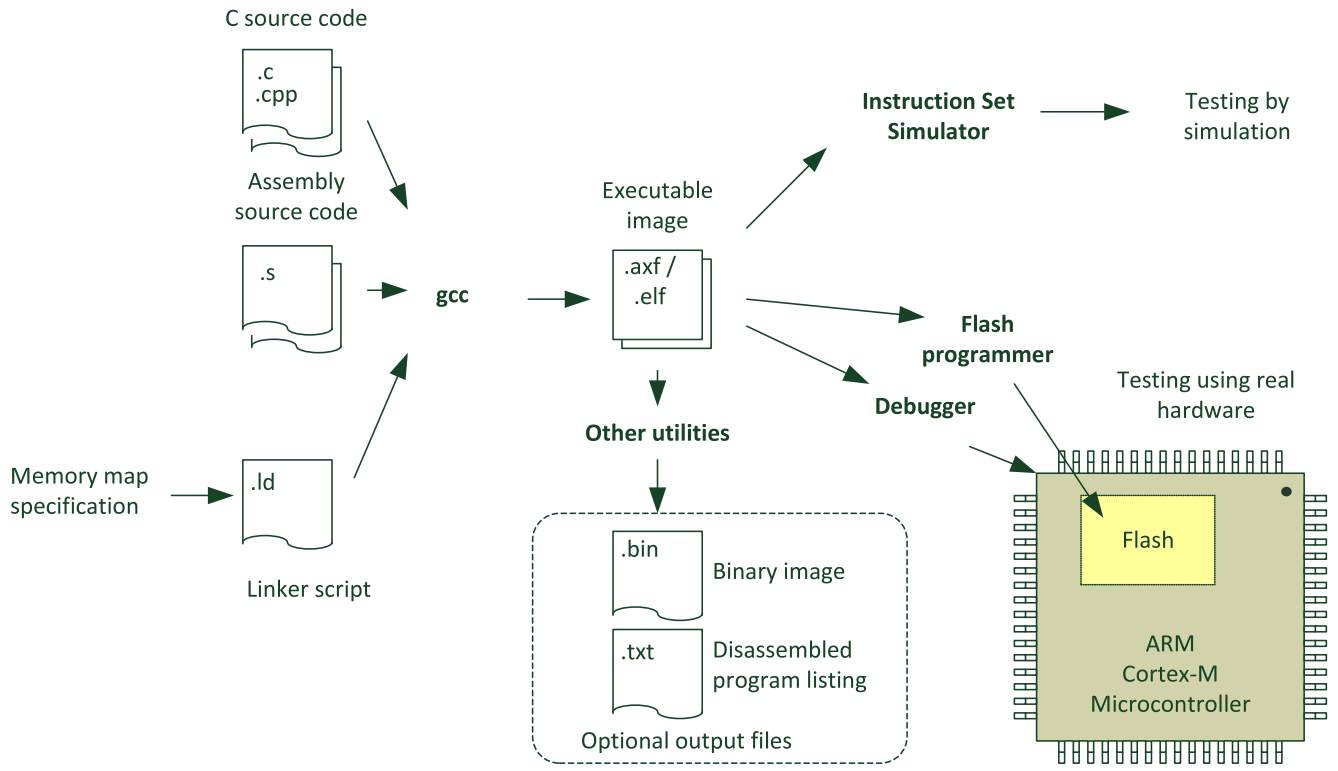
芯片厂商的**用户手册**是最重要的。用户手册介绍了芯片外设的编程模型，如寄存器等。

## 编译

### 一般的编译流程



### 采用GCC工具链的编译流程



## 控制流

### 轮询（Polling）

* 优点：简单
* 缺点：
  + 不易实现优先级 => 响应性低
  + 持续占用CPU，浪费计算资源
  + 浪费能源

### 中断驱动（Interrupt Driven）

* 强处：解决了轮询的缺点，节约计算资源和能源，易于实现优先级
* 弱点：不适合直接处理大型、复杂的任务。

中断驱动的弱点来自于两方面原因——首先，**为了保证实时性，中断例程必须在规定时间执行完**，否则将触发看门狗导致系统复位；这也就导致开发者必须人工实现任务的分片、调度，为开发引入了不必要的麻烦。

以下两张图是常见的两种中断驱动的控制流。

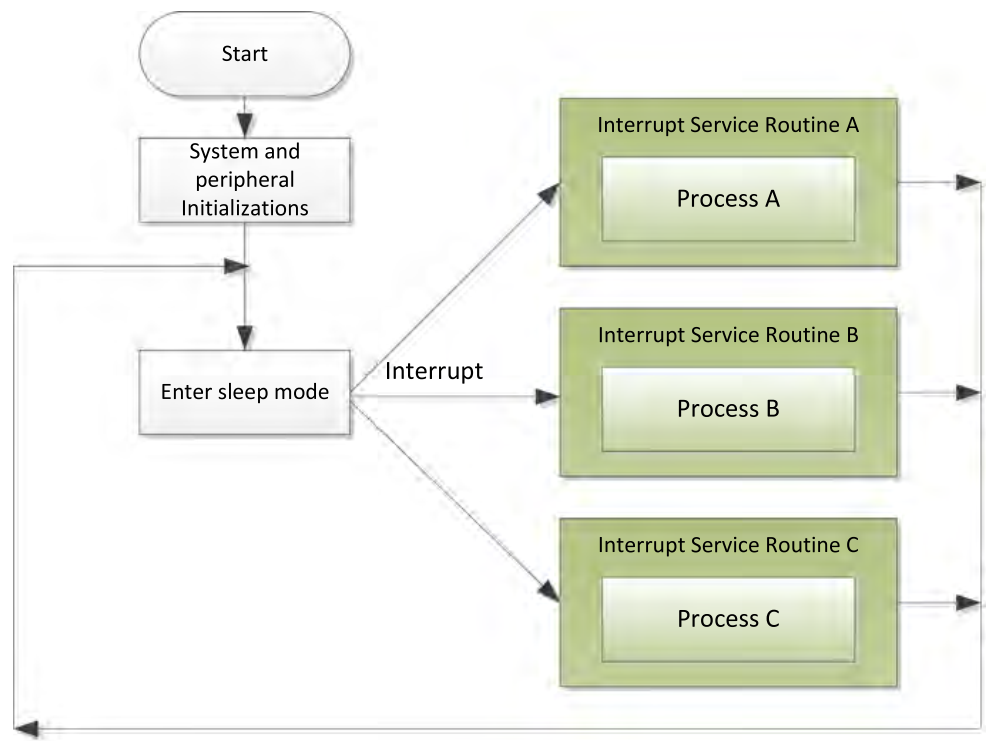


图 1 简单的中断驱动应用

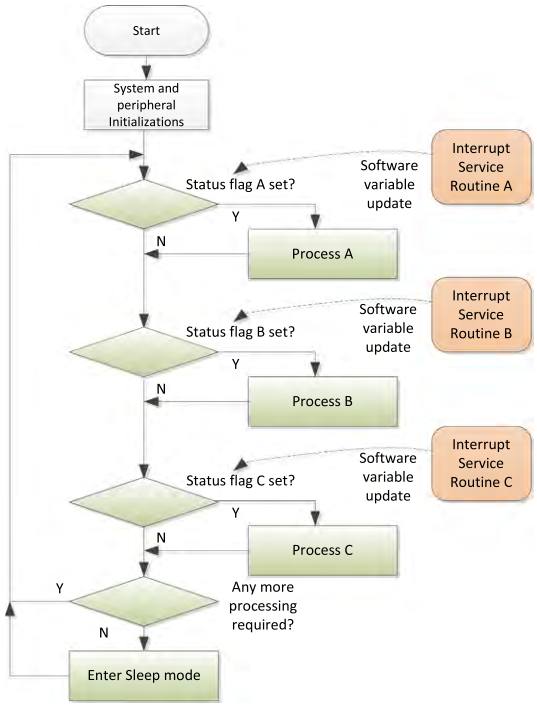


图 2 轮询+中断型应用

### RTOS

* 职责：把CPU时间分成时间片，并为进程分配时间片。
* 原理：采用计时器，每个时间片用完时产生计时器中断。这个计时器中断会触发操作系统的调度器，调度器接下来决定是否要切换上下文。

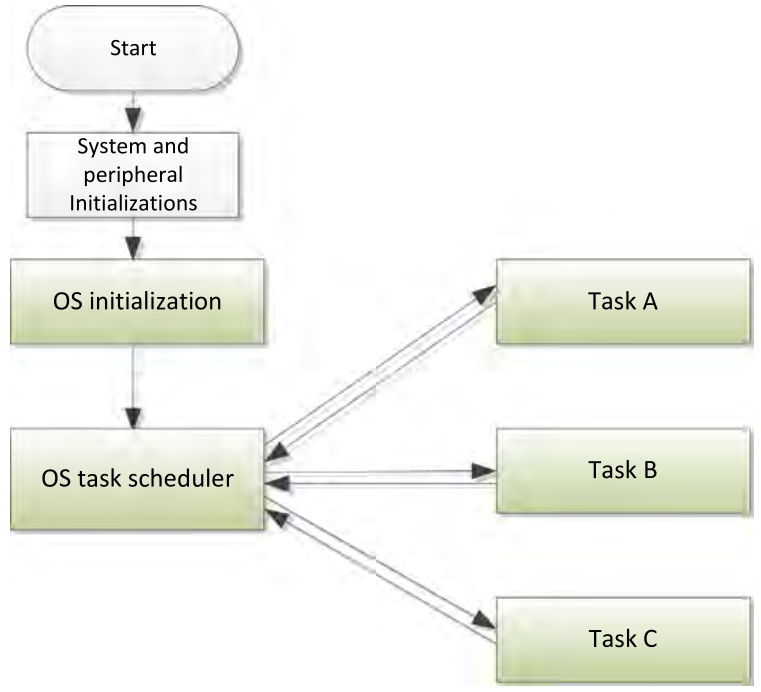


图 3 RTOS控制流

## I/O

Cortex M4芯片的外设是映射到系统内存上的，因此可以用指针访问外设的寄存器。

外设在使用之前需要初始化。初始化一般分为四步：

* 时钟初始化；
* I/O引脚的工作模式；比如，方向、复用方式、电气参数（如上拉下拉、推挽开漏）等。
* 外设配置；主要是通过外设与配置有关的寄存器来完成配置。
* 中断配置。