

嵌入式C语言之- 全局变量的作用域和生命周期

讲师：叶大鹏

助力你成为优秀的电子工程师！



变量的作用域和生命周期

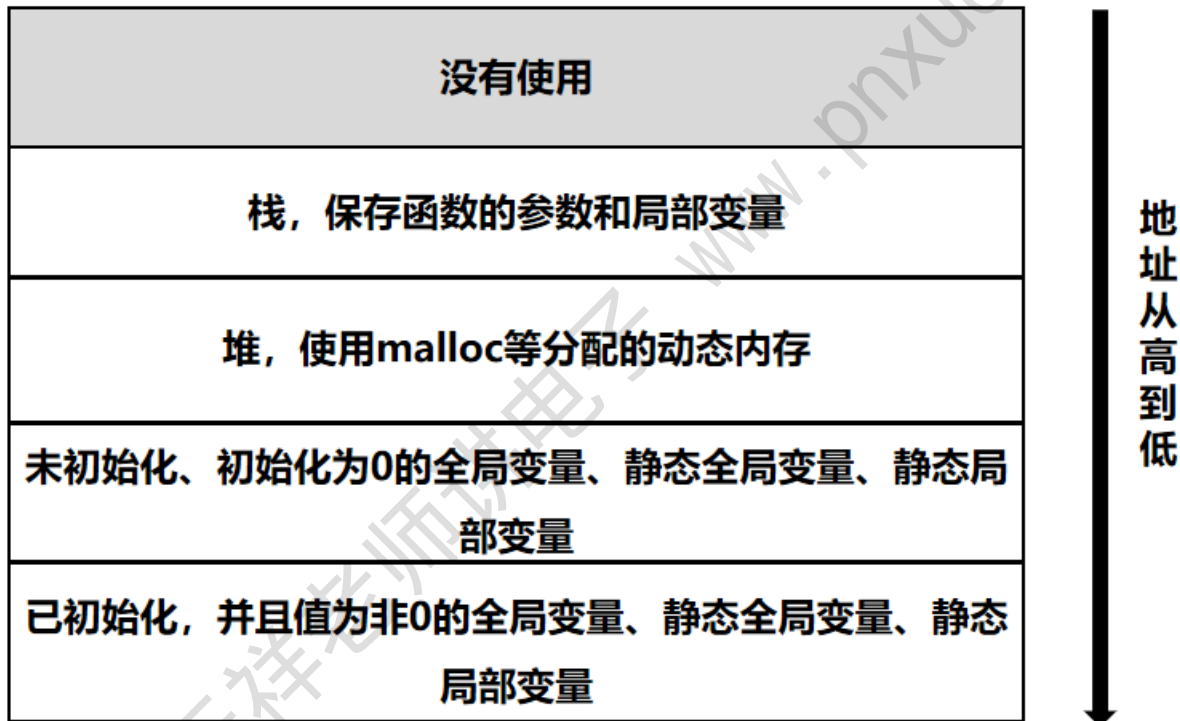
- 变量的作用域，指的是变量能够被使用的范围；针对的是程序编译链接阶段；
- 变量的生命周期，指的是变量创建（分配存储空间）到变量销毁（释放存储空间）之间的时间段（即变量的存在时间）；针对的是程序的执行阶段；
- 按照作用域和生命周期的范围，变量可以分为：
局部变量、静态局部变量、全局变量、静态全局变量

全局变量的作用域

- 定义在函数外部的变量称为全局变量（Global Variable），它的作用域默认是整个程序，也就是所有的源文件。

变量的存储空间

- 内存根据变量的生命周期和用途会划分为多个部分：



全局变量的生命周期

- 全局变量保存在内存的第1和第2部分，生命周期是整个程序的生命周期。

| | |
|---|--------------------------------|
| | 没有使用 |
| | 栈，保存函数的参数和局部变量 |
| | 堆，使用malloc等分配的动态内存 |
| 2 | 未初始化、初始化为0的全局变量、静态全局变量、静态局部变量 |
| 1 | 已初始化，并且值为非0的全局变量、静态全局变量、静态局部变量 |

全局变量的应用案例

- 在main函数中从外部（HMI或者网络）得到温度较准系数，在获取温度数据时，要根据这个系数进行运算。

全局变量的优缺点

- 全局变量的优点

- 1) 对于初学者友好，定义和使用起来都较为简单；
- 2) 全局变量内存地址固定，读写效率比较高。

- 全局变量的缺点

- 1) 长期占用内存

全局变量的生命周期是整个程序运行的生命周期，始终占据内存；

- 2) 增加程序之间的耦合性，降低了模块的可移植性和维护性

函数类似于一个黑匣子，一般是通过函数参数和返回值进行输入输出，函数内部实现相对独立。但函数中如果使用了全局变量，破坏了函数的独立性，使函数对全局变量产生依赖，不符合模块化设计思想，也不利于团队的分工协作。

嵌入式C语言之- 静态全局变量的作用域和生命周期

讲师：叶大鹏

助力你成为优秀的电子工程师！

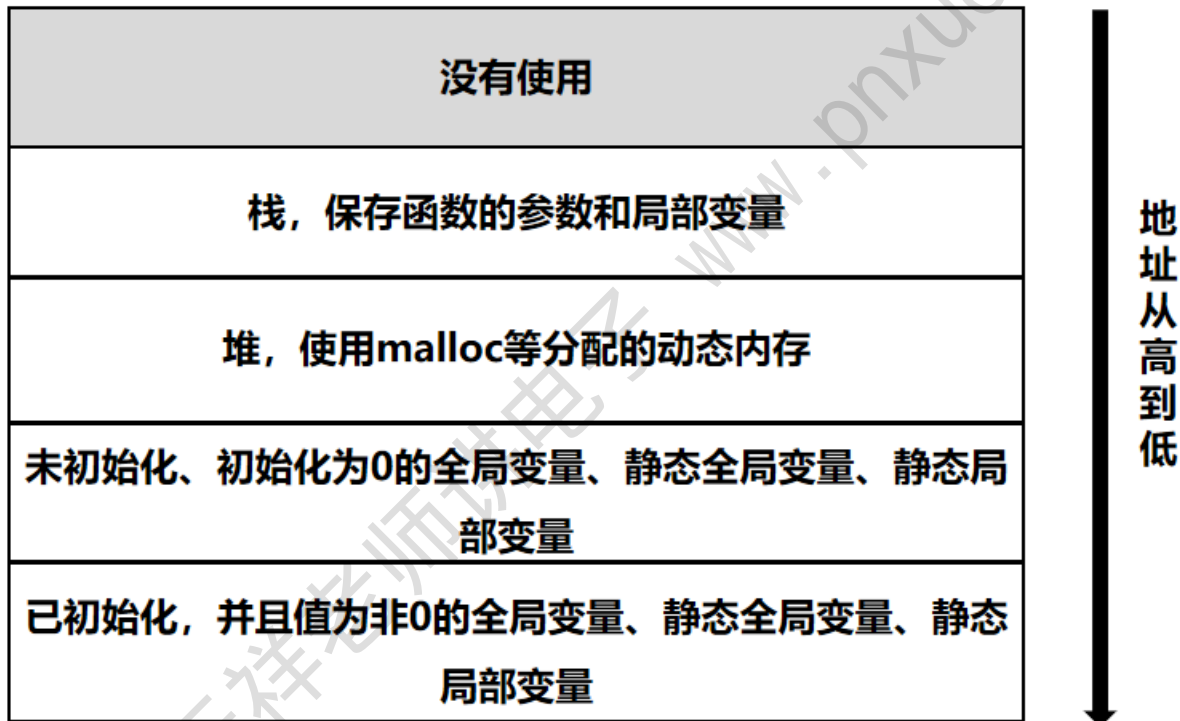


静态全局变量的作用域

- 静态全局变量是指在全局变量前加个关键字static，它的作用域是定义该变量的源文件内，而不是所有的源文件。

变量的存储空间

- 内存根据变量的生命周期和用途会划分为多个部分：



静态全局变量的生命周期

- 静态全局变量保存在内存的第1和第2部分，生命周期是整个程序的生命周期。

| | |
|---|--------------------------------|
| | 没有使用 |
| | 栈，保存函数的参数和局部变量 |
| | 堆，使用malloc等分配的动态内存 |
| 2 | 未初始化、初始化为0的全局变量、静态全局变量、静态局部变量 |
| 1 | 已初始化，并且值为非0的全局变量、静态全局变量、静态局部变量 |

嵌入式C语言之- 静态局部变量的作用域和生命周期

讲师：叶大鹏

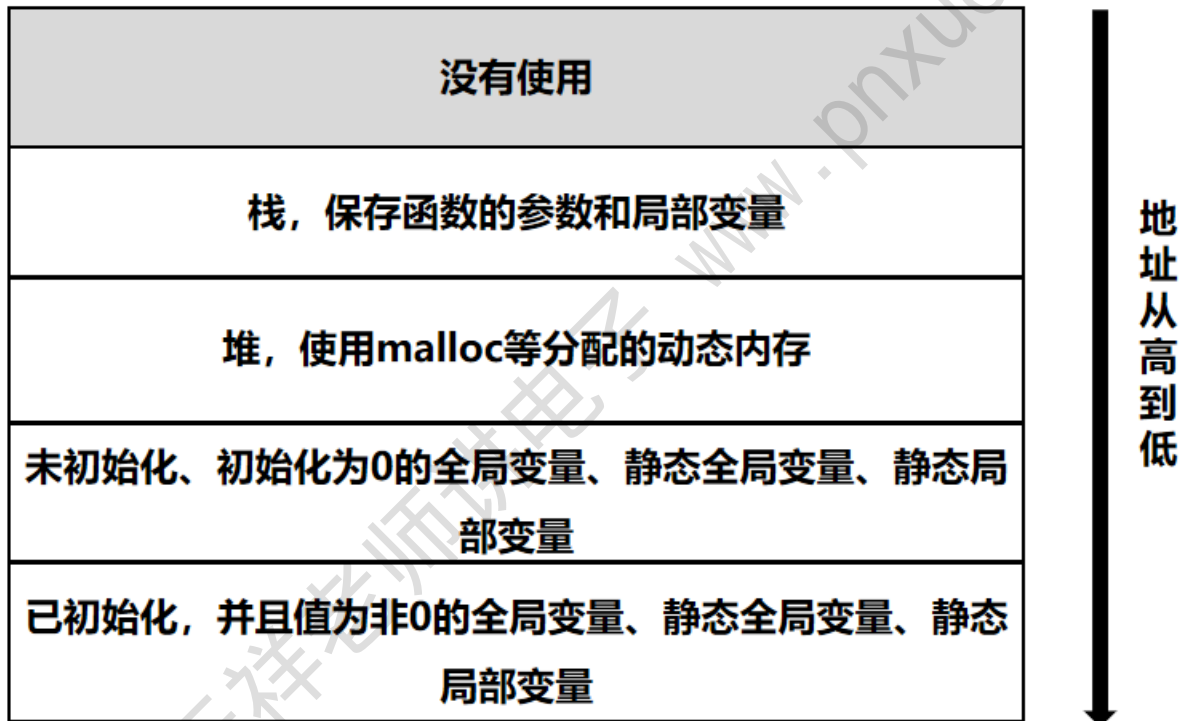
助力你成为优秀的电子工程师！

静态局部变量的作用域

- 静态局部变量是指在局部变量前加个关键字static，它的作用域是函数和代码块{ }之内。

变量的存储空间

- 内存根据变量的生命周期和用途会划分为多个部分：



静态局部变量的生命周期

- 静态局部变量保存在内存的第1和第2部分，它的生命周期是整个程序的生命周期，所以在函数退出时，静态局部变量不会被销毁，也就不会被释放存储空间。

| 没有使用 | |
|------|--------------------------------|
| | 栈，保存函数的参数和局部变量 |
| | 堆，使用malloc等分配的动态内存 |
| 2 | 未初始化、初始化为0的全局变量、静态全局变量、静态局部变量 |
| 1 | 已初始化，并且值为非0的全局变量、静态全局变量、静态局部变量 |

静态局部变量的应用案例

- 在tem.c的代码中，模拟实现每次调用static float GetRawData(void)能够获取到变化的温度原始数据。