# STM32&FreeRTOS学习笔记

## 编程风格

### 数据类型

|  |  |  |
| --- | --- | --- |
| 新定义的数据类型 | 实际的数据类型（C标准类型） | |
| portCHAR | char | |
| portSHORT | short | |
| portLONG | long | |
| portTickType | unsigned short int | 用于定义系统时基计数器的值和阻塞时间的值，  当 FreeRTOSConfig.h 头文件中的宏configUSE\_16\_BIT\_TICKS 为 1 时则为 16位 |
| unsigned int | 用于定义系统时基计数器的值和阻塞时间的 值 。 FreeRTOSConfig.h 头 文 件 中 的宏configUSE\_16\_BIT\_TICKS 为 0 时则为 32位。 |
| PortBASE\_TYPE | long | 根据处理器的架构来决定是多少位的， 如果是 32/16/8bit 的处理器则是 32/16/8bit 的数据类型。一般用于定义函数的返回值或者布尔类型 |

### 变量名

在 FreeRTOS 中，定义变量的时候往往会把变量的类型当作前缀加在变量上， 这样的

好处是让用户一看到这个变量就知道该变量的类型。 比如 char 型变量的前缀是 c， short 型变量的前缀是 s， long 型变量的前缀是 l， portBASE\_TYPE 类型变量的前缀是 x。 还有其他的数据类型，比如数据结构，任务句柄， 队列句柄等定义的变量名的前缀也是 x。

如果一个变量是无符号型的那么会有一个前缀 u， 如果是一个指针变量则会有一个前缀 p。因此，当我们定义一个无符号的 char 型变量的时候会加一个 uc 前缀， 当定义一个char 型的指针变量的时候会有一个 pc 前缀

### 函数名

函数名包含了函数返回值的类型、 函数所在的文件名和函数的功能，如果是私有的函

数则会加一个 prv（private） 的前缀。特别的， 在函数名中加入了函数所在的文件名， 这大大的帮助了用户提高寻找函数定义的效率和了解函数作用的目的， 具体的举例如下：

1. vTaskPrioritySet()函数的返回值为 void 型， 在task.c 这个文件中定义。

2. xQueueReceive()函数的返回值为 portBASE\_TYPE 型， 在 queue.c 这个文件中定义。

3. vSemaphoreCreateBinary()函数的返回值为 void 型， 在 semphr.h 这个文件中定义。

### 宏

|  |  |
| --- | --- |
| 前缀 | 宏定义的文件 |
| port（举例，portMAX\_DELAY） | portable.h |
| task(举例，taskENTER\_CRITICAL()) | task.h |
| pd(举例，pdTRUE) | projdefs.h |
| config(举例，configUSE\_PREEMPTION) | FreeRTOSConfig.h |
| err(举例，errQUEUE\_FULL) | projdes.h |

|  |  |
| --- | --- |
| 宏 | 实际的值 |
| pdTRUE | 1 |
| pdFALSE | 0 |
| pdPASS | 1 |
| pdFAIL | 0 |

## 列表