

queue.c文件中的重要函数

xQueueCreate

```
1 QueueHandle_t xQueueCreate(  
2                               UBaseType_t uxQueueLength,  
3                               UBaseType_t uxItemSize  
4                               );
```

创建一个队列。

xQueueCreateStatic

```
1 QueueHandle_t xQueueCreateStatic(  
2                               UBaseType_t uxQueueLength,  
3                               UBaseType_t uxItemSize,  
4                               uint8_t *pucQueueStorageBuffer,  
5                               StaticQueue_t *pxQueueBuffer  
6                               );
```

使用静态方式创建一个队列。

xQueueSendToToFront

```
1 BaseType_t xQueueSendToToFront(  
2                               QueueHandle_t xQueue,  
3                               const void *pvItemToQueue,  
4                               TickType_t xTicksToWait  
5                               );
```

将一个元素送到队首。队列中的元素通过复制而不是引用。

xQueueSendToBack

```
1 BaseType_t xQueueSendToBack(  
2                               QueueHandle_t xQueue,  
3                               const void *pvItemToQueue,  
4                               TickType_t xTicksToWait  
5                               );
```

将一个元素送到队尾。

xQueueSend

```
1 BaseType_t xQueueSend(  
2                               QueueHandle_t xQueue,  
3                               const void * pvItemToQueue,  
4                               TickType_t xTicksToWait  
5                               );
```

插入一个元素到队列中。

xQueueOverwrite

```
1 BaseType_t xQueueOverwrite(  
2                               QueueHandle_t xQueue,  
3                               const void * pvItemToQueue  
4                               );
```

插入一个元素到队列中，如果队列已经满，则覆盖。

xQueueGenericSend

```
1 BaseType_t xQueueGenericSend(  
2                               QueueHandle_t xQueue,  
3                               const void * pvItemToQueue,  
4                               TickType_t xTicksToWait  
5                               BaseType_t xCopyPosition  
6                               );
```

和 `xQueueSend` 效果相同，但是不是推荐的API。推荐使用 `xQueueSend()`，`xQueueSendToFront()` and `xQueueSendToBack()` 这三个函数。

xQueuePeek

```
1 BaseType_t xQueuePeek(  
2                               QueueHandle_t xQueue,  
3                               void * const pvBuffer,  
4                               TickType_t xTicksToWait  
5                               );
```

获取一个队列中的元素但是不删除其在队列中的位置。

xQueueReceive

```

1 BaseType_t xQueueReceive(
2                               QueueHandle_t xQueue,
3                               void *pvBuffer,
4                               TickType_t xTicksToWait
5                               );

```

获取一个队列中的元素，成功访问后就删除该元素在队列中的位置。

uxQueueMessagesWaiting

```

1 UBaseType_t uxQueueMessagesWaiting( const QueueHandle_t xQueue );

```

返回储存在队列中信息的数目。

uxQueueSpacesAvailable

```

1 UBaseType_t uxQueueSpacesAvailable( const QueueHandle_t xQueue );

```

返回队列中的可用空间。

vQueueDelete

```

1 BaseType_t xQueueSendToFrontFromISR(
2                               QueueHandle_t xQueue,
3                               const void *pvItemToQueue,
4                               BaseType_t *pxHigherPriorityTaskWoken
5                               );

```

删除队列。

上面的函数定义是为了在任务间传输数据，下面的函数用于 **co-routines**（联合任务）

xQueueCreateSet

```

1 QueueSetHandle_t xQueueCreateSet( const UBaseType_t uxEventQueueLength ) PRIVILEGED_FUNCTION;

```

创建一个集合，以便机器能够让任务堵塞或者等待。

xQueueAddToSet

```
1 BaseType_t xQueueAddToSet( QueueSetMemberHandle_t xQueueOrSemaphore, QueueSetHandle_t
  xQueueSet ) PRIVILEGED_FUNCTION;
```

添加信号量到队列中。

xQueueRemoveFromSet

```
1 BaseType_t xQueueRemoveFromSet( QueueSetMemberHandle_t xQueueOrSemaphore, QueueSetHandle_t
  xQueueSet ) PRIVILEGED_FUNCTION;
```

从一个队列中移除一个信号量。

xQueueSelectFromSet

```
1 QueueSetMemberHandle_t xQueueSelectFromSet( QueueSetHandle_t xQueueSet, const TickType_t
  xTicksToWait ) PRIVILEGED_FUNCTION;
```

选择一个信号量。

非公共的API:

```
1 void vQueueWaitForMessageRestricted( QueueHandle_t xQueue, TickType_t xTicksToWait, const
  BaseType_t xWaitIndefinitely ) PRIVILEGED_FUNCTION;
2 BaseType_t xQueueGenericReset( QueueHandle_t xQueue, BaseType_t xNewQueue )
  PRIVILEGED_FUNCTION;
3 void vQueueSetQueueNumber( QueueHandle_t xQueue, UBaseType_t uxQueueNumber )
  PRIVILEGED_FUNCTION;
4 UBaseType_t uxQueueGetQueueNumber( QueueHandle_t xQueue ) PRIVILEGED_FUNCTION;
5 uint8_t ucQueueGetQueueType( QueueHandle_t xQueue ) PRIVILEGED_FUNCTION;
```