+ Idle Task Hook Functions

It is possible to add application specific functionality directly into the idle task through the use of an idle hook (or idle callback) function—a function that is called automatically by the idle task once per iteration of the idle task loop.

+ Queue:

Length and data store

Send/receive data from queue

+ Binary semaphore

+ Counting semaphore

Just as binary semaphores can be thought of as queues having a length of one, so counting semaphores can be thought of as queues having a length of more than one. Tasks are not interested in the data that is stored in the queue—just whether the queue is empty or not. Each time a counting semaphore is ‘given’, another space in its queue is used. The number of items in the queue is the semaphore’s ‘count’ value.

+ Counting semaphores are typically used for two things:

1. Counting events.

In this scenario, an event handler will 'give' a semaphore each time an event occurs causing the semaphore’s count value to be incremented on each ‘give’. A handler task will 'take' a semaphore each time it processes an event—causing the semaphore’s count value to be decremented on each take. The count value is the difference between the number of events that have occurred and the number that have been processed. Counting semaphores that are used to count events are created with an initial count value of zero.

2. Resource management.

In this usage scenario, the count value indicates the number of resources available. To obtain control of a resource a task must first obtain a semaphore, decrementing the semaphore’s count value. When the count value reaches zero, there are no free resources. When a task finishes with the resource, it 'gives' the semaphore back, incrementing the semaphore’s count value. Counting semaphores that are used to manage resources are created so that their initial count value equals the number of resources that are available.