**Chapter 3-3 RTOS Queue**

Welcome back to WICED WiFi 101. In this video I am going to talk about the RTOS queue. The semaphore is a scheme to signal between threads. But what happens if you have more information that you need to send other than “wave”? In this case you use a queue.

The WICED queues are documented in components->rtos->queues. There you can see the functions to initialize and de-initalize a queue, check the state of a queue and finally add and subtract, or more properly push and pop to the queue.

Each wiced\_rtos\_queue can only pass one type of message. The type of the message is customizable by you, with the only condition being that it is one or more words (multiples of 32-bits) in size. When you create a queue with wiced\_rtos\_queue\_init you pass

* The queue handle
* A char \* The name of the queue
* The size of each message in the queue
* The number of messages that can be placed into the queue

Now, let’s look at the queue example project which can be found as ww101/03/04\_queue

This project will send a message to a thread via the queue… and the message is simply how many times to blink the led. The first time the button is pressed it will blink one time, next time the button is pressed it will blink twice… and so on

You start in the application\_start function by initializing the button (so that it can send messages). Then you initialize the queue. For this example we will have the message just be a 4-byte integer and we let the queue store up to 10 messages. Finally, we start the thread that will blink the LED based on the number in the queue.

That function – ledThread – will try to pop a value from the queue using wiced\_rtos\_pop\_from\_queue, then it will blink the LED that number of times. If there is nothing in the queue, the wiced\_rtos\_pop\_from\_queue function will halt the thread until there is something in the queue.

Again, we use a timeout value of WICED\_WAIT\_FOREVER so that the thread will never continue until a value is available in the queue.

The last bit of code is the button\_isr which will just count up using the blinks integer, then send that value into the queue using wiced\_rtos\_push\_to\_queue.

When I run the program, you can see that the first time I press the button it blinks the LED once, then twice, then three times etc.

As always, you can post your comments and questions in our Wifi developer community or you are welcome to email me at alan\_hawse@cypress.com or tweet me at @askioexpert with your comments, suggestions, criticisms and questions.

In the next video I will introduce you to RTOS Timers