**Chapter 3-1 RTOS Threads**

Welcome back to WICED WiFi 101. In the old school way of doing things, embedded programs looked like the while(1) infinite loop. And this is a fine way of doing things until you program gets more complicated. When you find that you have multiple jobs to do and they all have different timing constraints, it is time to start using an RTOS.

The most fundamental unit in an RTOS is a Thread. You can think of a thread as an independent or mostly independent job that needs to be done. Each thread probably looks like a bit of initialization code and then the old school while(1) infinite loop.

To create a thread in WICED we gave you an API called “wiced\_rtos\_create\_thread”

You can read about this API in the documentation under Components 🡪 RTOS 🡪 Thread. The function takes 6 arguments.

* The handle structure which is of type wiced\_thread\_t \*
* A priority which can be 0 to 31 (higher numbers are lower priority)
* A char \* which is the name of the thread … the name is used by the debugger to help you see which threads are running
* The function which you want to run … the while(1) loop function
* The size of the stack
* And a general-purpose argument which you can use to make several threads using the same function but with different parameters

Lets look at the blinking LED example, but this time we will make the led blink using a thread. You can find this example in the text book in chapter 3 exercise 1 and in the answer key under ww101key/03/01\_thread.c.

This is very straight forward.

1. We create a function called ledThread
2. It has a while(1) loop
3. The loop toggles the LED state
4. Then waits
5. And goes on until the end of time… or the power goes out

When you call wiced\_rtos\_delay\_milliseconds it actually puts the thread to sleep and allows other threads to run.

This would be a good time to observe that application\_start is just a thread… it just so happens to be automatically run for you when the chip starts up.

The other thing to note is that when you get to the end of the function … like if it falls out the bottom of the function .. the thread just dies and the system goes on with its life.

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 talk about semaphores