**UART**

Welcome back to Cypress Academy, WICED WiFi 101. In the last video I used the built-in and pre-configured UART to display information in a terminal window. But, what if you want to use a different configuration – for example a different baud rate – or if you want to accept input from the terminal in addition to sending output to the terminal? I will cover both of these cases in this video.

First, let’s send data to the UART at 9600 baud instead of at 115200 baud. I will copy the 05\_interrupt project to 10\_uartsend and update the files as necessary.

I am going to re-use the same physical UART pins as the one used for WPRINT\_APP\_INFO so I need to tell the SDK to not initialize it with the default settings. That is done by adding the following line to the make file:

GLOBAL\_DEFINES += WICED\_DISABLE\_STDIO

You can use this same method can to define any macro at build time…. In this case this line has exactly the same effect as doing “#define WICED\_DISABLE\_STDIO”

Now that I have disabled the default UART settings, I need to configure and initialize the UART with my desired settings. I start by setting up the configuration structure a shown and then calling the initialization function with that structure.

Notice that you can find the allowed selections for each parameter by using “Open Declaration” as I showed you in the earlier video.

Now that the UART is configured and initialized, I can send data using the wiced\_uart\_transmit\_bytes function. In this case, I will print a single digit representing how many times the button has been pressed.

Now I will program the board and test it out using a putty serial terminal set for 9600 baud.

Next let’s create a project that will receive input from the terminal. I will use a press of the “0” on the keyboard to turn the LED off and “1” to turn it on.

I will copy 10\_uartsend to 11\_uartrecieve and make the necessary updates to the project configuration. Note that the WICED\_DISABLE\_STDIO define is needed in the make file for this project as well.

In this case, in the initialization function I provide a receive buffer in addition to the UART configuration. This buffer is where the receive data will be stored. I read data from the buffer using wiced\_uart\_receive\_bytes and then turn the LED on or off depending on the character received.

Let’s program the board and test it out. Now I can turn the LED on and off from the keyboard.

In the next video, I’ll show you how to use PWMs that are built into the chip.

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