Answer Key

Chapter 02

Exercise 01

1. The table at the top of platform.h says that WICED_SH_LED0 connects to WICED_GPIO_7, Arduino header D10, and WICED_PWM_2. Explain how this mapping was determined. You will need to refer to platform.h, platform.c and the schematic for the base board.

   From platform.h line 403, WICED_SH_LED1 is mapped to WICED_GPIO_7

   From platform.c line 48, WICED_GPIO_7 is assigned to PIN_GPIO_11

   From platform.c line 119, PIN_GPIO_11 is assigned to WICED_PWM_2

   From the schematic page 9, GPIO_11 connects to D10

Exercise 02

1. Why can't you read the value of the LED using the *wiced_gpio_input_get* function instead of using a variable to remember the state?

   The wiced_gpio_input_get function is only valid if the pin is configured as an input pin.

2. In what file and on what line does the WICED_SH_LED1 get assigned to the correct pin for this kit?

   platform.h, line 360.

3. In what file and on what line is the pin connected to the LED set as an output?

   platform.c, line 333.

Exercise 08

1. What I2C addresses are found?

   0x42   Analog Co-processor

   0x3C   OLED Display

Chapter 03

Exercise 02

1. Do you need *wiced_rtos_delay_millisecon*ds in the LED thread? Why or why not?

   No, because the semaphore causes the thread to suspend until it is set by the button ISR.

2. What happens if you use a value of 100 for the semaphore timeout? Why?

   The LED will blink every 100ms because the semaphore will timeout even when the button is not pressed.

Exercise 03

1. What happens if you forget to unlock the mutex in one of the threads? Why?

   The thread that has the lock will keep running but the other thread will stay suspended because it can never get access to the mutex. Therefore, only one of the buttons will cause the LED to blink (the one that has the lock).

Exercise 05

1. What happens if you don't remove the while(1) loop from the function that blinks the LED? Why?

   The LED will appear to stay on all the time (in fact, it is blinking on/off rapidly) so it appears dim. The reason is that as soon as the timer executes the LED blinking function once, it never exits so it continually blinks the LED with no delay.

2. What happens if the application_start doesn't have a while(1) loop? Why?

   The chip will continuously reset because there are no active threads once application_start exits. Remember that the timer is NOT a thread on its own.

3. Does the while(1) loop in application_start need a delay? Why or why not?

   No, because application_start is the only thread in the project.

Chapter 05

Exercise 02

1. There are three changes required in the wifi_config_dct.h file:

   CLIENT_AP_SSID changes to *"WW101OPEN"*

   CLIENT_AP_PASSPHRASE changes to *""*

   CLIENT_AP_SECURITY changes to ***WICED_SECURITY_OPEN***

   > Hint: you can find all of the security types available by right clicking on WICED_SECURITY_OPEN (or any other security name) from the DCT file and selecting "Open Declaration".

Chapter 07b

Exercise 04

2. How do the MQTT library functions (e.g. wiced_mqtt_publish) get into your project?

   The line "`$(NAME)_COMPONENTS :=` protocols/MQTT" in the make file causes the MQTT library functions to be included in the project.

3. What function is called when the button is pressed?

   `publish_callback`

4. How does the button callback unlock the main thread?

   It sets a semaphore using "`wiced_rtos_set_semaphore( &wake_semaphore );`"

5. What WICED-SDK RTOS mechanism does the "wait_for_response" function use to "wait"?

   It gets a semaphore using "`wiced_rtos_get_semaphore( &msg_semaphore, timeout )`"

6. Why did the firmware author create a function called "wait_for_response"?

   This function is used in several of the functions in the main application to cause the thread to sleep until a specific MQTT event occurs.

7. Are all messages sent to the AWS IOT MQTT Message Broker required to be in JSON format?

   No, but messages that affect the shadow have to be JSON.

8. What are the 7 WICED MQTT events? What file are they defined in?

   WICED_MQTT_EVENT_TYPE_CONNECT_REQ_STATUS
   WICED_MQTT_EVENT_TYPE_DISCONNECTED
   WICED_MQTT_EVENT_TYPE_PUBLISHED
   WICED_MQTT_EVENT_TYPE_SUBSCRIBED
   WICED_MQTT_EVENT_TYPE_UNSUBSCRIBED
   WICED_MQTT_EVENT_TYPE_PUBLISH_MSG_RECEIVED
   WICED_MQTT_EVENT_TYPE_UNKNOWN
   They are defined in mqtt_common.h.

9. Do you have to name the client certificate client.cer? How would you change the name?

   No, the name can be changed in the make file ($(NAME)_RESOURCES).

10. What is the naming convention used to differentiate WICED MQTT library functions versus wrappers around those functions in the publisher app?

    The library functions all start with "wiced_mqtt" while the wrapper functions start with "mqtt".

11. What steps are required to get an MQTT connection established?
    - Initialize wiced_mqtt_security_t with the credentials from the DCT using resource_get_readonly_buffer
    - Allocate memory for the MQTT object using malloc
    - Get the IP address of the MQTT message broker using wiced_hostname_lookup.
    - Call wiced_mqtt_init to initialize.
    - Call mqtt_open_connection to open the connection.

12. What prevents a hung connection from deadlocking the publisher app?

    The wait_for_response function has a timeout parameter that is passed as a timeout to the semaphore.

13. What is the name of the flag that prevents the firmware from sending multiple button presses before the publish is finished?

    pub_in_progress

Exercise 07

1. What is the sequence of events that changes the LED from On to Off?
   - A shadow *update* message is published by the device. This contains a JSON message to turn off the LED.
   - The document is updated and a shadow *update/documents* message is published by AWS.
   - A shadow *update/accepted* message is published by AWS.