Laynie Tierney

CS-350-11296-M01 Emerging Sys Arch & Tech  
27 July 2024

Milestone 2: Q&A

After working through this process, you will need to address the following questions:

**• How does the macro UART\_DATA\_BINARY impact the UART?**

1. The UART\_DATA\_BINARY macro configures the UART (Universal Asynchronous Receiver/Transmitter) to handle data in a binary format. When this macro is used, the UART is set up to transmit and receive raw data bytes without any interpretation or transformation. This means that the data is sent exactly as it is, which is essential for accurate communication, especially when dealing with non-textual or binary data.

**• How does the macro UART\_RETURN\_FULL impact the UART?**

1. The UART\_RETURN\_FULL macro configures the UART to ensure that it will only return when the buffer is completely full. This means that the UART will wait until the entire buffer is filled before it processes or returns the data. This setting can be useful when you want to handle large blocks of data in one go, rather than processing smaller chunks of data incrementally.

**• What driver call would you use to write 10 characters out of the UART?**

1. To write 10 characters to the UART, you would use the UART2\_write function provided by the UART driver. This function allows you to specify the number of bytes to write, which in this case would be 10. The function will handle the task of sending these 10 characters to the UART device. For example in C, [UART2\_write(uartHandle, dataBuffer, 10, &bytesWritten);] would be an example call.
2. So, in this call:

* uartHandle is the handle to your UART device.
* dataBuffer is the pointer to the buffer containing the data to be written.
* 10 is the number of characters to write.
* &bytesWritten is a pointer to a variable that will receive the number of bytes actually written.

**• What is the driver call to turn off LED 0?**

1. To turn off LED 0 using the GPIO driver, you would use the GPIO\_write function. You pass in the GPIO pin identifier for LED 0 and set its state to low. Example call in C: [GPIO\_write(CONFIG\_GPIO\_LED\_0, CONFIG\_GPIO\_LED\_OFF);]
2. So, in this call:

* CONFIG\_GPIO\_LED\_0 is the identifier for the LED 0 GPIO pin.
* CONFIG\_GPIO\_LED\_OFF is the value to set the pin to, which turns the LED off.

**• What is the UART baud rate?**

1. The UART baud rate is a measure of how fast data is transmitted over the UART communication channel. It specifies the number of bits transmitted per second. In your setup, the baud rate is set to 115200 bits per second. This is a common baud rate for serial communication, providing a balance between speed and reliability.