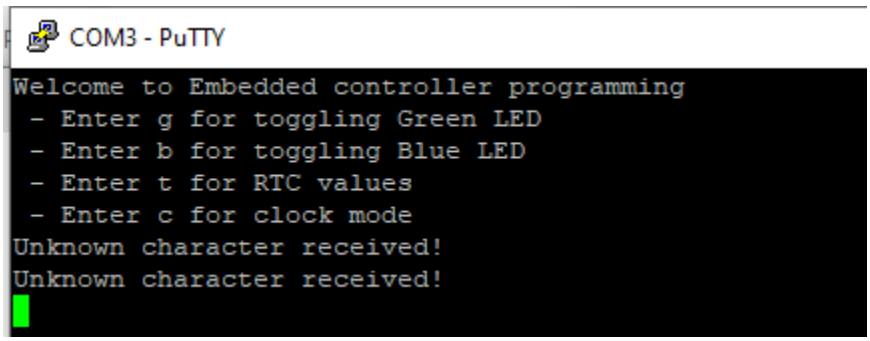
Nathan Bunnell
Embedded Controller Programming for Real-Time Systems
ECE-40097
Student ID: U08895857

Date: 8/30/2021

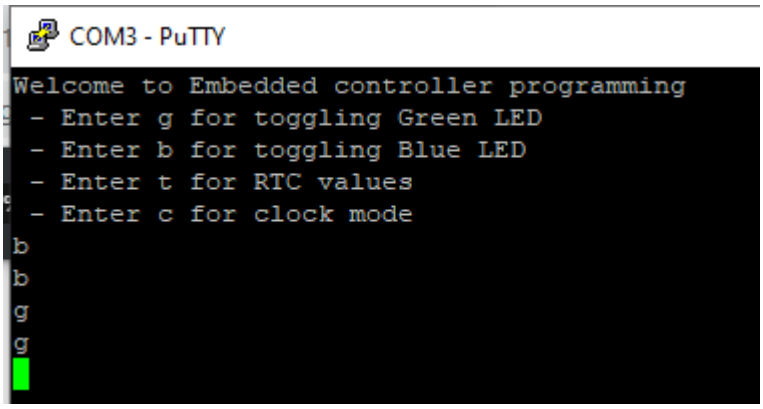
Bonus Assignment

See below for the terminal output of the methods defined for this assignment. Please see attached files for full comments within function prototypes.

Full prompt developed with the HAL_UART_Transmit_DMA() call and unknown character response. logMsg() and logGetMsg() implemented the DMA call by enabling DMA for each UART channel, defining callbacks similar to the interrupt version, and*critically*, ensuring that the MX_DMA_Init() call occurs prior to the MX_USART1_UART_Init() call. Researching into why this order matters didn't provide much info other than an ST forum post where an employee indicated that it was a known issue that should have been fixed in a later release of the IDE.



Blue/ green LED toggle. DMA behavior was similar to the interrupt example used in the final, much more responsive than standard polling.



Nathan Bunnell
Embedded Controller Programming for Real-Time Systems
ECE-40097
Student ID: U08895857

Date: 8/30/2021

Modify MX_RTC_Init() to set current date and time in main.c

Date/ time config from MX_RTC_Init():

```
/** Initialize RTC and set the Time and Date
*/
sTime.Hours = 0x12;
sTime.Minutes = 0x15;
sTime.Seconds = 0x0;
sTime.DayLightSaving = RTC_DAYLIGHTSAVING_NONE;
sTime.StoreOperation = RTC_STOREOPERATION_RESET;
if (HAL_RTC_SetTime(&hrtc, &sTime, RTC_FORMAT_BCD) != HAL_OK)
{
  Error_Handler();
}
sDate.WeekDay = RTC_WEEKDAY_FRIDAY;
sDate.Month = RTC_MONTH_AUGUST;
sDate.Date = 0x27;
sDate.Year = 0x21;

if (HAL_RTC_SetDate(&hrtc, &sDate, RTC_FORMAT_BCD) != HAL_OK)
{
```

Print current date/time from console code. I don't recall it being noted in the lecture, apparently the HAL_RTC_GetTime () function call must be followed by the HAL_RTC_GetDate() call in order for the clock registers to continue to update. I finally found the resolution to a "stuck" RTC in an ST forum and then within the comments in stm32l4xx_hal_rtc.c.

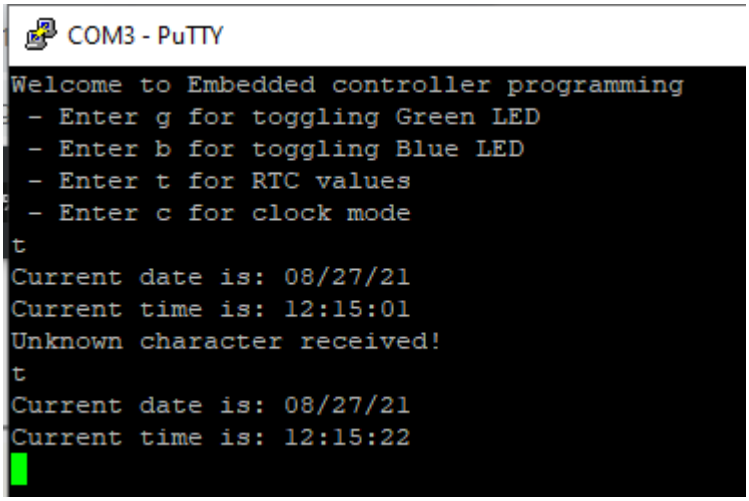
```
            // Print RTC time
            case ('t'):
            {
                logMsg(&huart1, "t\n");
                HAL_RTC_GetTime(&hrtc, &rtcTime, RTC_FORMAT_BIN);
                HAL_RTC_GetDate(&hrtc, &rtcDate, RTC_FORMAT_BIN);

                char buffer[100] = {0};       // Large char buffer for string printing
                snprintf(buffer, sizeof(buffer), "Current date is: %02d/%02d/%02d\n", rtcDate.Month, rtcDate.Date, rtcDate.Year);
                //HAL_UART_Transmit_DMA(&huart1, (uint8_t*) buffer, strlen(buffer));
                logMsg(&huart1, buffer);

                HAL_Delay(100);

                snprintf(buffer, sizeof(buffer), "Current time is: %02d:%02d:%02d\n", rtcTime.Hours, rtcTime.Minutes, rtcTime.Seconds);
                //HAL_UART_Transmit_DMA(&huart1, (uint8_t*) buffer, strlen(buffer));
                logMsg(&huart1, buffer);

                break;
            }
```

```
* @note  You must call HAL_RTC_GetDate() after HAL_RTC_GetTime() to unlock the values
*        in the higher-order calendar shadow registers to ensure consistency between the time and date values.
*        Reading RTC current time locks the values in calendar shadow registers until Current date is read
*        to ensure consistency between the time and date values.
* @param  hrtc RTC handle
```

Nathan Bunnell
Embedded Controller Programming for Real-Time Systems
ECE-40097
Student ID: U08895857

Date: 8/30/2021

Reading the current time on the console:



I also implemented a “clock display” function, where the controller would go into a while(1) loop and print the time every 5 seconds, proving that the registers will update in real time. This did require a controller reset, although in hindsight I could have used the blue user button or another interrupt to exit in a more graceful manner.

```
            case ('c'):
            {
                logMsg(&huart1, "Entering clock mode. Reset req'd to exit!\n");
                char buffer[100] = {0};       // Large char buffer for string printing
/*
                HAL_RTC_GetTime(&hrtc, &rtcTime, RTC_FORMAT_BIN);

                snprintf(buffer, sizeof(buffer), "\rCurrent time is: %02d:%02d:%02d", rtcTime.Hours, rtcTime.Minutes, rtcTime.Seconds);
                logMsg(&huart1, buffer);
*/
                while(1)
                {
                    HAL_RTC_GetTime(&hrtc, &rtcTime, RTC_FORMAT_BIN);
                    HAL_RTC_GetDate(&hrtc, &rtcDate, RTC_FORMAT_BIN);

                    snprintf(buffer, sizeof(buffer), "Current time is: %02d:%02d:%02d", rtcTime.Hours, rtcTime.Minutes, rtcTime.Seconds);
                    logMsg(&huart1, "\r");
                    logMsg(&huart1, buffer);

                    HAL_Delay(5000);
                }

                break;
            }
```

Nathan Bunnell
Embedded Controller Programming for Real-Time Systems
ECE-40097
Student ID: U08895857

Date: 8/30/2021

Console output:

COM3 - PuTTY

```
Welcome to Embedded controller programming
 - Enter g for toggling Green LED
 - Enter b for toggling Blue LED
 - Enter t for RTC values
 - Enter c for clock mode
Entering clock mode. Reset req'd to exit!

Current time is: 12:15:35
Current time is: 12:15:40
Current time is: 12:15:45
Current time is: 12:15:50
Current time is: 12:15:55
Current time is: 12:16:00
Current time is: 12:16:05
```