**Project #1**

**Introduction:**

The goal of this particular project was to further familiarize us with both the IAR workbench and the STM32F207ZG microcontroller. The project made use of the onboard timers, user button, and LEDs. As a result, we were exposed to the utilization of the I/O functions of the microcontroller.

**Project Description:**

Project one tasked us with getting the onboard LEDs to illuminate for a specified amount of time. A single onboard timer would handle the timing of the LEDs. Also, pressing the user button would control the lighting of the individual LEDs. Once the user button was pressed a timer was to activate LED1 would activate immediately with the other three individually following at a five hundred-millisecond interval. If the button at this point had been released by the user the process would restart, however, if the button was still being held all four LEDs were to cycle off for a period of two hundred and fifty milliseconds and on again for seven hundred and fifty milliseconds. This cycle would continue until the user button was released which would again cause the process to restart.

**Design/Development:**

The approach our group took was a simple yet efficient one. We first enabled the onboard user button so we could set up the structure our wait and check loops would take. Using a hex calculator it was found that pin 6 (0x40h) would be active if the user button was being pressed. This method allowed our group to code a simple compare with the branching either going to the restart state or continuing depending on whether or not the button was being held down. Next, we enabled the LEDs, which went fairly quickly. Once they were enabled we used the aforementioned user button loops to light the LEDs. Since they did what was expected we then moved on to the timers. The on board timers were by far the most difficult piece of this lab. They presented numerous different ways to proceed. The documentation regarding them was a bit lacking even within the reference manual. We ended up making the decision to forgo the more intricate aspects of the timers and instead made use of a single timer. Using this timer we could compare it to the known intervals we needed our LEDs to activate at. When the clock needed to be reset we used the default behavior of its update functionality to reset the timer to zero. This approach worked very well and allowed our group to bypass having to make use of functionality such as interrupts and interrupt flags. We made the timing comparison equal to the number in the counter. Even though this is generally discouraged, the wait loop executes over and over again for 16,000 clock cycles. We came to the conclusion that it was impossible to miss a count.

**Conclusion:**

Our groups project did in fact work as intended all LEDs illuminated at the correct interval. The user button controlled them as it was meant to. Overall the project was a fun one we learned a great deal about the I/O of the board and how certain elements of it function. If we had one critique to make it would most certainly be the timers. Most resources online make use of the C equivalent where the timers have an initialization method predefined. If more descriptive and meaningful documentation could be found on the timers and how each of their functions affected them I think it would be a great help to classes in the future.

**Contributions:**

Zac compiled the lab report. Brandon and Andrew proofread the lab report. All three contributed to the implementation of the project on the board.