Lab 4 Report - Proximity Sensor

**Procedure**

For this lab, an ultrasonic proximity sensor was connected to pins PB6 and PB7 on the STM µController. The sensor was connected to these pins, because those are the pins which UART1 can use with GPIOB. UART1 was initialized similar to UART2 with no parity bit, a Baud rate of 9600, 8 data bits, and one stop bit. We modified the UART2 interrupt handler to account for the additional “t” and “p” keys. When either “t” or “p” were pressed, the sensor was prompted for either the temperature in Celsius or the distance, respectively.

A queue was previously used to communicate between tasks, so the same queue was used to communicate between UART1 and UART2. UART2 was used to communicate with the user via PuTTY, while UART1 was used to communicate between the distance sensor and the µController.

A UART task was created to direct the data between the queues and the UARTs. The interrupt handler for UART2 would put data into the queue for UART1 to receive, and UART1 would poll the proximity sensor for data.

**Results**

The biggest roadblock for us was finding a time to get together and work on the lab. After working on it separately, and implementing the UART1, it stopped turning the LED on and off for a while. We went back to our code from Lab 3 and slowly built up from there.

We encountered some issues with the stack size, which we decided to switch back to the default configMINIMAL\_STACK\_SIZE and saw progress. At one point, the code stopped working on Wesley’s computer. Once we switched to Hayden’s, it was working again. We assumed there was a line of code we couldn’t see that was interfering with the tasks, but for about an hour we were attempting to debug things that we couldn’t discover.

We were able to get the sensor to send information back, but the timing appeared to be off and the UART didn’t want to receive the data. A portion of the temperature could be read, but it didn’t seem to send correct data through UART2 to the user. We were able to use sprintf to format the string printed to the putty terminal and get a partial result (see **Figure 1**).

**Figures**

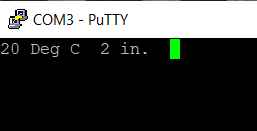


Figure 1: Putty terminal

**Conclusion**

We were very close to get things working properly. We could check and see that the proximity sensor would send us values, but the way our UART task was written, UART1 would not receive the values correctly. There was more time to complete this lab than any lab previous. Ironically, with a bit more time, we would be able to get things functioning properly.