Brian Lopez
Github root directory: github.com/brian4280/Bed_Assign

Date Submitted: **10/21/18**

Task 01:

Youtube Link: https://www.youtube.com/watch?v=L6LbPGEr4Dc

First thing that was done is to enable ADC. This is the exact same as the previous assignments so I will not include it in this report. Then a few extra variables will be declared for the ADC values:

```
    // for obtaining ADC values
    uint32_t ADCVals[4];
    uint32_t TempF, TempC, Avg;
    char Far[10];
```

Inside the while loop, ADC is turned on and the values will be obtained and converted:

```
while(1)
    {
        while(SSIDataGetNonBlocking(SSI0_BASE, &pui32DataRx[0]))
        {
        }
        // turn on ADC
        ADCIntClear(ADC0_BASE, 1);
        ADCSequenceEnable(ADC0_BASE, 1);
        ADCProcessorTrigger(ADC0_BASE, 1);

        while(!ADCIntStatus(ADC0_BASE, 1, false))
        {
            // poll until ADC is complete
        }

        // grab ADC values and convert to F
        ADCSequenceDataGet(ADC0_BASE, 1, ADCVals);
        Avg = (ADCVals[0] + ADCVals[1] + ADCVals[2] + ADCVals[3] + 2) / 4;
        TempC = (1475 - ((2475 * Avg)) / 4096) / 10;
        TempF = ((TempC * 9) + 160) / 5;
        // convert the int to a string
        ltoa(TempF, Far);
        ADCSequenceDisable(ADC0_BASE, 0);
```

I have a variable called ‘Far’ that contains the string of the temperature value. That is what will be used when sending data through SSI:

```
        for(ui32index = 0; ui32index < NUM_SSI_DATA; ui32index++)
        {
            // send the 2 numbers from the temperature
            //NUM_SSI_DATA was changed to 2 for this task
            UARTprintf("'%c' ", Far[ui32index]);
            SSIDataPut(SSI0_BASE, Far[ui32index]);
        }
```

The rest of the while loop is unchanged from task 0

Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

Brian Lopez
Github root directory: github.com/brian4280/Bed_Assign

Task 02:

Youtube Link: https://www.youtube.com/watch?v=2pChUoXNJK8

ADC and the clock is first turned on, then a few variables are declared which are mainly for the ADC values:

```
enableADC(); // set clock and turn on ADC

// used for getting temperature from adc
uint32_t Vals[4];
uint32_t Avg, TempC, TempF;
char Far[10];
```

After that SSI is turned on with the nokia.h file and the screen is initially cleard:

```
SysTick_Init();
startSSI0();
initialize_screen(BACKLIGHT_ON,SSI0);
clear_screen(SSI0);
SysTick_Wait50ms(100);
```

Inside the while loop, the screen is first cleared, then my name is written on the top left, and the word “Temp:” is written on the left, center of the screen. After that ADC is turned on and the values are obtained and converted to F. The integer value is converted to a string and then the value is written on the center of the screen every. A delay of 1 second is then called until it starts over:

```
    while(1)
    {
        // clear screen and write name and 'Temp:'
        clear_screen(SSI0);
        screen_write("Brian Lopez",ALIGN_LEFT_TOP,SSI0);
        screen_write("Temp:", ALIGN_LEFT_CENTRE, SSI0);

        // enable ADC and get value
        ADCSequenceEnable(ADC0_BASE, 1);
        ADCIntClear(ADC0_BASE, 1);
        ADCProcessorTrigger(ADC0_BASE, 1);

        while(!ADCIntStatus(ADC0_BASE, 1, false))
        {
        }

        ADCSequenceDataGet(ADC0_BASE, 1, Vals);

        Avg = (Vals[0] + Vals[1] + Vals[2] + Vals[3] + 2) / 4;
        TempC = (1475 - ((2475 * Avg)) / 4096) / 10;
        TempF = ((TempC * 9) + 160) / 5;
        // convert integer to a string
        ltoa(TempF, Far);
        // Add the F for fahrenheit after int value
        Far[2] = 'F';
        Far[3] = '\0';  // need to update NULL for screen_write function
```

Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

```
    screen_write(Far, ALIGN_CENTRE_CENTRE, SSI0);
    SysTick_Wait50ms(20);
}
```

Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.