**Demonstration:** (NOTE: these change based on the lab - enter the demo tasks specified in each lab)

LCD shows names, brightness, temperature and internal voltage values \_\_\_\_\_\_\_\_

Pressing Switch 1 increases the brightness of the LEDs \_\_\_\_\_\_\_\_

Pressing Switch 2 decreases the brightness of the LEDs \_\_\_\_\_\_\_\_

Turning Analog In potentiometer controls the brightness \_\_\_\_\_\_\_\_

Brightness values are updated as the brightness of the LEDs changes \_\_\_\_\_\_\_\_

**Requirements:**

The code generated is written in C for the YRDKRX63N Evaluation Board. \_\_\_\_\_\_\_\_

The brightness of LED4 to LED15 is controlled using a PWM signal generated

using a delay function \_\_\_\_\_\_\_\_

LCD should display your group member(s) first name on line 1 and line 2 \_\_\_\_\_\_\_\_

LCD should display the brightness value on line 3 as B Value= XXX, where XXX

is the duty ratio. [Example: ‘001’ corresponds to a value of ‘1’] \_\_\_\_\_\_\_\_

LCD should display the temperature value on line 4 as T Value=XX.X, where

XX.X is the temperature in degree Celsius. \_\_\_\_\_\_\_\_

LCD should display the internal reference voltage on line 5 as V Value=X.XX,

where X.XX is the internal reference voltage. \_\_\_\_\_\_\_\_

Maximum value of brightness is 100 and the minimum is 0. Zero brightness value

means the LEDs are off\_ \_\_\_\_\_\_\_

LCD information should be centered\_ \_\_\_\_\_\_\_

Switch 1 ISR is used to increase the brightness in steps of 10\_ \_\_\_\_\_\_\_

Switch 2 ISR is used to decrease the brightness in steps of 10 \_\_\_\_\_\_\_

“Analog In” potentiometer is used to control the brightness in steps of 1 [Range: 0

– 9] \_\_\_\_\_\_\_\_

No ADC interrupts or Timers are used \_\_\_\_\_\_\_\_

The code is well documented and easy to follow. \_\_\_\_\_\_\_\_

**Learning Objectives:**

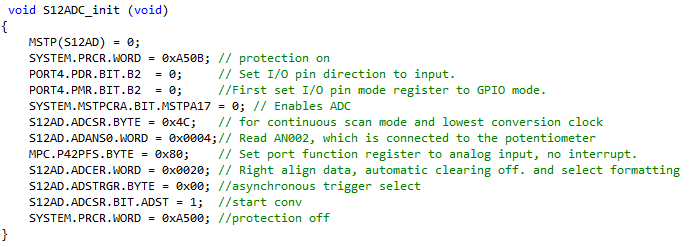
This experiment shows the conversion of analog inputs to digital and then displays it on the LCD. Also the ability to read internal reference voltages and temperature on the LCD is demonstrated.

**General Steps:**

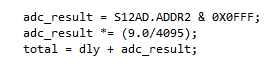
1. Go to the HEW IDE
2. Power up the Renesas RX63N by connecting it to the computer via mini-USB
3. Build upon previous lab and add ADC functionality and other requirements.
4. Demonstrate the working project.

**Detailed Steps:**

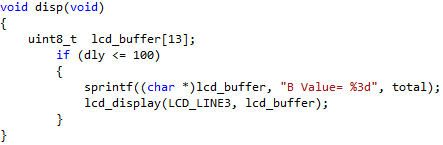
1. The third lab is an expansion upon the previous, lab 2. The switches 1 and 2 are used to increment and decrement the value of the PWM. In the present lab the values were changed to increment every 10 steps instead of 1 resulting in a value from 0-100 in steps of 10.This functioned as the rough range adjustment. The analog potentiometer is also used on the RX63N development board acting as the fine adjustment and the values range from 0-9. The combination of the two adjustments were totaled and passed into the functions that controlled the PWM for LED4-LED15.
2. After realizing requirements and copying the lab 2 project a function to initiate the ADC must be created. Following the steps outlined in the lab handout the function was created as follows.

****

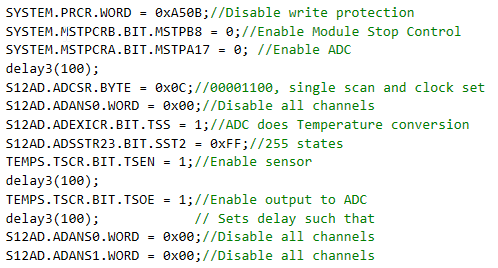
1. After successfully initializing the ADC to read in the analog potentiometer values the function must be called into the main and used in such a way that the value can be displayed onto line 3. The value from the ADC is stored in the **adc\_result** variable inside the while loop and then converted so that the steps of the analog potentiometer are from 0-9. This is then added to a total variable which is used to display the result on line 3 of the LCD.



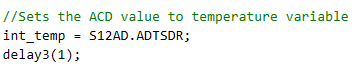
1. In order for the LCD to be updated correctly during each potentiometer or switch event a display function is created to listen for these events and control the display. This is similar to lab 2 and uses its code as a building block. A character array is created which stores the B Value and then the pointer is updated by the **total** and displayed in line 3.

****

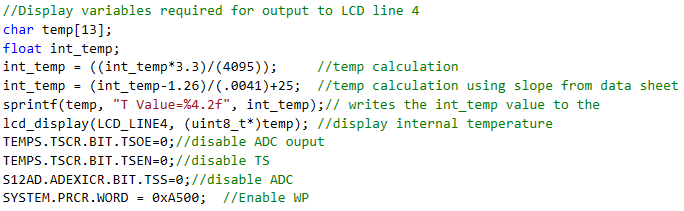
1. The requirement is to create a function that utilizes the ADC to read internal temperature of the RX63N. The **S12ADC\_temp** function creates a LCD character buffer as before, enables the ADC and temperature sensor for reading, stops conversion, converts the temperature, and finally displays it on the LCD screen at line 4.



1. Performs the conversion of the temperature value in the ADC and takes the reading while the sensor is on. ****
2. The ADC value read in from the temperature sensor is written to the **int\_temp** variable.

****

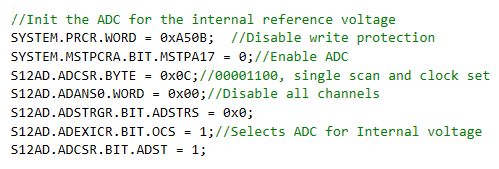
1. The character array and display variables for temperature in Celsius.

****

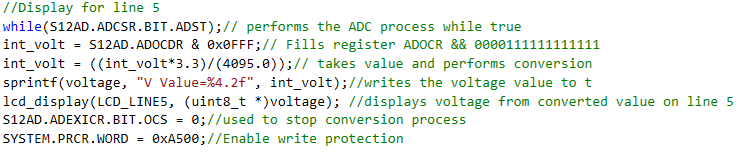
1. The last requirement was to access the internal reference voltage and display it on line 5 of the LCD. The function created to perform this access is named **voltage()**. The function similarly to the others sets up the ADC to be used with the reference voltage and then display it.
2. The initial variables are defined making it so that the value from the ADC can be written into them and then displayed using a character buffer on the LCD screen.



1. Next the ADC is initialized.

****

1. After successful initialization the value is written to a variable converted based on known conversion factor and then displayed to line 5.

****

**Observations:**

Throughout the lab the conversion of analog to digital values was evident. The ability to successfully initialize and implement the ADC was a must. Upon initialization of the internal reference voltage it was a mystery as to what the reference was for. After further investigation the value is used as a conversion constant in the ADC and helps to obtain accurate conversions. The internal reference voltage appears in the datasheet and empirically in the range of 1.45-1.55V. This display appears to be dim during some of the lower ranges of the 0-100 PWM scale. This is based on the main While loop updating faster than the LCD can write. To mitigate the situation a delay is added to allow for sufficient write times by the LCD. Ideally this is not the best solution and the display function could be modified to stop the dimming. Due to time constraints this was not possible and the dimming will have to stay until more time is available.

**Summary:**

The assigned lab experiment three outlined the basics of Analog to Digital conversion and the implementation of on board peripherals such as internal reference voltage and temperature sensors. The implementation of these components is necessary for a core understanding of ADC and without it more complex projects would not be possible.