Christopher Cilip **ECE 4161/5196 Lab 01 Report** **Due 01/21/09**

Cmcilip@Uncc.edu 800373993 **Page 1 of 1**

Chris Renn
Crrenn@uncc.edu 800306286

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

LED’s flashing as specified______________ ________

Other functionality works as in original code ________

The two members of the lab team who have not taken ECGR 4161/5196 must demonstrate how to use the IDE tools____________________ ________

General learning objectives:
To familiarize ourselves with the coding process of a microcontroller, in this instance, the Renases QSK16C62P. Many times sample code can be used as a base to launch code that will perform your specific operation, manipulating existing code was also a primary objective.

General steps:
First the board and coding software was loaded onto the computer. Once the controller was up and running, basic code manipulation was required to render our desired output. The code was then saved and uploaded to the controller where the desired output was displayed.

Detailed steps:

1. Selecting the correct microcontroller in the development software (QSK16C62P)
2. Changing the current 4 case structure to an 8 case structure to accommodate for the extra states that were required in the output.
3. Manipulating the counter so that it can read all 8 cases instead of the orginal 4.
4. Selecting the correct MCU updating driver, in this instance the M30622F8P was the desired MCU file.
5. Upload and Run the manipulated file on the microcontroller by using the GUI of the High-Performance Embedded Workshop

The manipulated code is as follows:

```
/* The changes made effect these entire snipets and are highlighted below
#pragma INTERRUPT          ta1_irq              // interrupt vector defined in sect30.inc
void ta1_irq(void)
{
        static char disp_count = 0;                          // LED display control variable
        static char first_time = 1;

        /* Only drop down the Xc drive after the first interrupt */
        if( first_time ) {
          cm03 = 0;                                  /* Set Xc clock to low drive */
                first_time = 0;
        }

        led_display(disp_count);       // display current count

        ++disp_count;                  // increment display counter
        if (disp_count > 8)            /*The addition of this code allows the counter to count higher than 4 cases, in this instance, 8.
                disp_count = 1;
}


 /* The entire case structure was changed to give a new desired output.  In this case the order will be G, R, Y, All, G,Y,R, All (Repeat)

void led_display(char count)
{

        switch (count) {
                case 1:                                             /* green on */
                case 5:                                             /* green on */
                        RED_LED = LED_OFF;
                        YLW_LED = LED_OFF;
                        GRN_LED = LED_ON;
                        break;
                case 3:                                             /* yellow on */
                case 6:                                             /* yellow on */
                        RED_LED = LED_OFF;
                        YLW_LED = LED_ON;
                        GRN_LED = LED_OFF;
                        break;
                case 2:                                             /* red on */
                case 7:                                             /* red on */
                        RED_LED = LED_ON;
                        YLW_LED = LED_OFF;
                        GRN_LED = LED_OFF;
                        break;
                case 4:
                case 8:                                             /* All LED on */
                        RED_LED = LED_ON;
                        YLW_LED = LED_ON;
                        GRN_LED = LED_ON;
                        break;
                default:                                    /* all LED's off */
                        RED_LED = LED_OFF;
                        YLW_LED = LED_OFF;
                        GRN_LED = LED_OFF;
        }
}
```

It was important to observe the details when setting up the microcontroller. Knowing the correct MCU file is necessary to completing your task and since there are many to choose from, knowing the correct one will greatly expedite the time required to get going.

This lab showed important specifics pertaining to the QSK16C62P starter kit. The simple manipulation of code should already be a skill possessed at this level and this lab seemed to be geared towards getting familiar with microcontrollers, specifically the QSK16C62P; this objective was achieved.