

## ECE 354: Project Part 1

### Timer

The first timer (timer0) was implemented using interrupts to generate a simple clock. The timer was set to fire an interrupt approximately every 10 ms. Two counters were used, one to count up to 100 to represent 1 full second and another to contain the total number of seconds elapsed.

In the main run loop, a local variable was used to store the last count of seconds sent to the UART1. A check is performed on each loop to detect if the current count is not equal to the last one. If the count has changed, then update the out string and prepare to output it to the UART. A rudimentary string library was implemented to handle converting the integers into strings. A integer was used both as a flag to indicate that the device needs to output a new string as well as to keep an index to the next character to print.

Printing was performed using an interrupt. A global variable, out\_char, was used as both a flag and character storage. It was used to indicate in the main run loop that the UART is ready to accept the next character. The UART interrupt sets the value of out\_char to '\0' to indicate readiness. If it is ready, the value is updated to the next character to output by the main run loop.

```
void c_timer_handler(void) {
    timer_count++;
    if (timer_count == 100) {
        counter++;
        reset timer_count;
    }
    acknowledge the interrupt
}

int main (void) {
    initialize local and global variables
    setup timer ISR vector
    setup UART ISR vector
    setup timer registers

    while(true) {
        if (counter value has changed) {
            convert counter value into hours, minutes, and seconds
            compile a string in the format: "hh:mm:ss"
            set flag for outputting the string
            save the current counter value
        }
        if (output flag is high
            AND the UART is ready
            AND there is a character to print) {
            set the output character to the next character to print
        }
    }
}
```

```
        call an interrupt on UART1
        increment index counter
    }
}
return 0;
}
```

## Memory Management