

EE 335 Advanced Microcontroller Engineering

Oregon Tech Portland, Winter 2014

Lab Assignment #1 – Heartbeat
Due January 16

Objective:

The student will write an interrupt driven program which will create a “heartbeat.” This function will be incorporated into all future lab assignments.

Equipment and Software needed:

- Dragon12-Plus2-DB board and USB cable
- Computer running Windows XP or later or classroom computer
- USB Flash drive if you use the classroom computer
- AsmIDE installed on computer or EmbeddedGNU if you are using the C language

General Instructions:

It is often difficult to determine if a non-working program has “crashed” or external circuitry is not working. To assist in this determination we will design a “heartbeat indicator” which we will test in this assignment and include in all remaining assignments including the final project. The indicator, an LED, is to flash on and off at approximately one cycle per second.

Where I currently work, this indicator is called a “happy light” because when it flashes everything is happy.

We went through the design of an interrupt driven timer in the lecture. The source code for that design is at the end of this assignment. For this lab unit use the Real Time Interrupt (RTI) instead of a timer channel. Configure the RTI interrupt to occur roughly once a millisecond, then count the number of interrupts to equal one half second, as close as possible. The count might not be 500 since the interrupt rate isn't exactly 1 kHz. Configure port T pin 7 as an output and toggle the bit every half second.

The Dragon12 board has one or two LEDs (depending on the board version) you can use for the indicator labeled “LP” for *logic probe*. So you only need to connect PT7 to LP to complete the wiring. The LED should flash 0.5 seconds on, 0.5 seconds off, repeating, while the program is running.

To Turn In:

1. Commented program listing.
2. You need to ascertain that the program runs correctly. What is the expected flash rate? What is the measured flash rate and how did you perform the measurement?

```
;; This program will increment variable time every millisecond.  
;; Add your own comments!  
#include registers.inc
```

```
time    org    RAMSTART  
        ds     2  
        org    PRSTART  
        lds    #RAMEND  
        movw   #tc7int UserTimerCh7  
        movw   #0 time  
        bset   TIOS #80  
        bset   TSCR1 #90  
        bset   TIE #80  
        ldd    TCNT  
        addd   #1000*24  
        std    TC7  
        cli  
  
loop:   bra    loop  
  
tc7int: ldd    time  
        addd   #1  
        std    time  
        ldd    TC7  
        addd   #1000*24  
        std    TC7  
        rti  
  
end
```