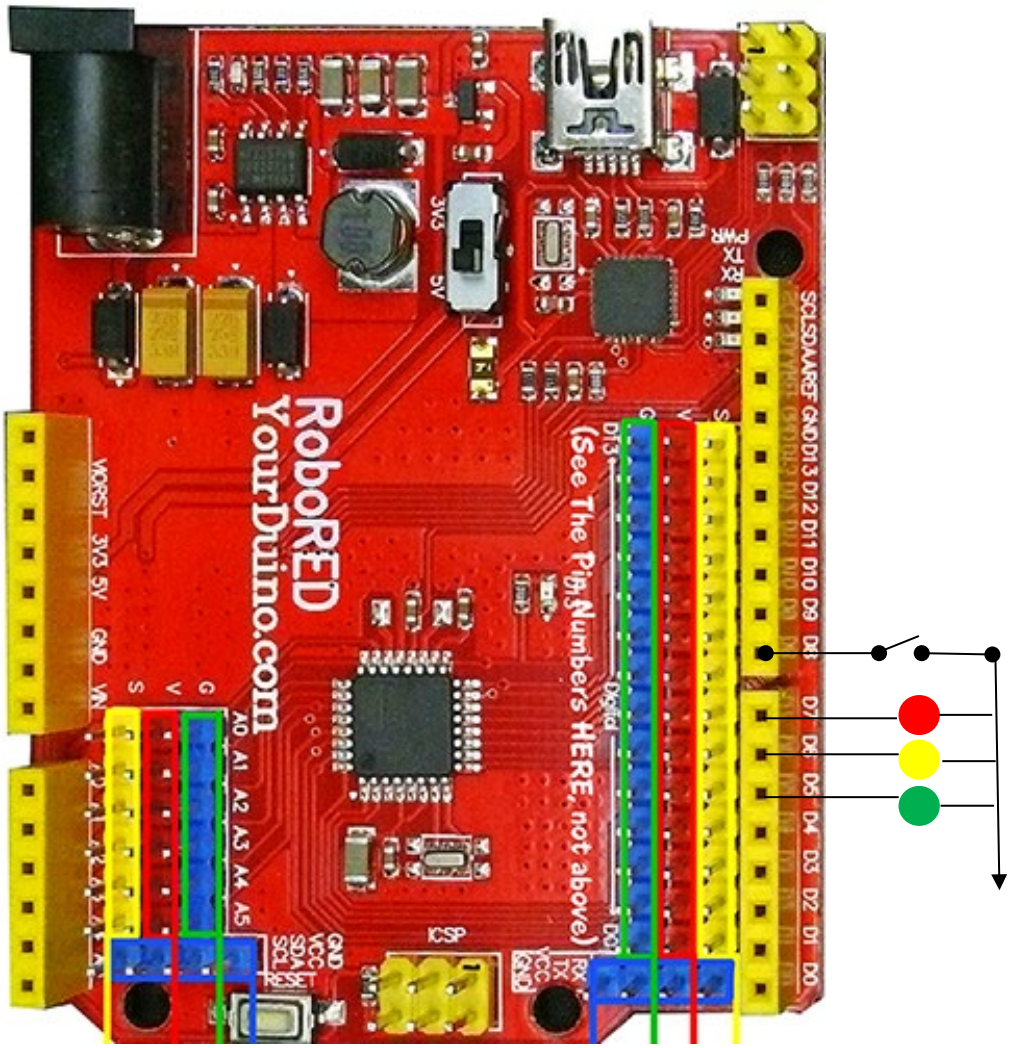
Homework #5 – Due 2/19/20
Please submit to the Dropbox in MyCourses

Problem:

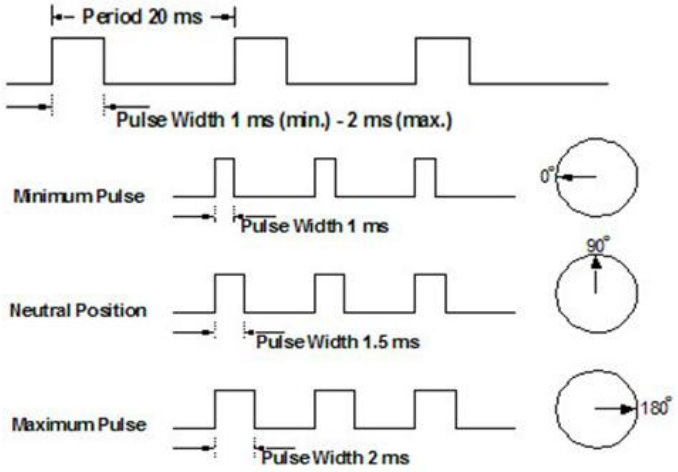
1. The circuit below is for a reaction timer. It operates as follows:
   - After a random amount of time the yellow LED goes ON, and a timer starts
   - If the user presses the button before the timer times out, the green LED goes ON and the yellow goes OFF.
   - If the timer times out, the red LED goes ON, and the yellow goes OFF

   Write the code for the reaction timer.
   - You can use delay() and random() for the delay before yellow light
   - Check for the button push in the main program loop, and use the timer interrupt to know that time is up
   - You can start with setting the timer for ½ second and shorten it from there.
   - Be sure to disable the interrupt as soon as the user presses the button, otherwise the interrupt will still fire!



2. Rewrite lab #3 – section 6 using registers and timer interrupts. Note that when interrupts are used for the timers, you do not have to increment a state timer. Remember that when a variable is updated outside the main loop (like in an ISR) you must use the volatile designation so that the compiler does not optimize the variable out.

3. Typically, servo motors should react to a PWM signal as shown below.



   a. In reality though, a pulse width of 1 ms does not always put the servo at 0° and pulse width of 2 ms does not put the servo at 180°. Using timer1 and a FAST PWM mode (see the handout entitled PWM guide), create a pwm signal to drive a servo motor. You may need to adjust the registers for pulse width to get a full 180 degrees of rotation. Note, you cannot just copy the code from lab 4 as that was not fast PWM mode.

4. Research question:

   Read sections 'Reset and Interrupt handling' in the datasheet and answer the following questions:
   1. How does uC handle multiple interrupts arriving from different peripherals while an ISR is being serviced
   2. What does the uC do in the 4 cycle response time to service an ISR.