Work Completed

This past week I focused on programming the microprocessor to control acceleration and braking in both modes. The current main program is seen below (initializations and declarations are excluded):

```c
mode_choose = WIP_switch();
while (mode_choose==1)
{
}
while (mode_choose==0)
{
    vertical_j = read_a2d(1); // sample the analog value on RA0
    if (vertical_j<120)
    {
        diff = abs(vertical_j-123);
        duty = 128*(diff/123);
        CCPR1L = duty;
        CCPR2L = 128;
    }
    if (vertical_j>126)
    {
        diff = abs(vertical_j-133);
        duty = 128*(diff/133);
        CCPR1L = 0;
        CCPR2L = duty;
    }
}
```

This code first calls WIP_Switch() to determine the mode of operation that has been chosen by the user. WIP_Switch() returns a 1 or a 0. 1 represents wheel mode while 0
represents joystick mode. While mode=1, the functions that control acc and braking in wheel mode will be executed (not yet written). While mode =0, the functions controlling acc/braking in joystick mode will be executed. First, the analog voltage from the vertical joystick signal will be converted to a digital signal via port RA0. If this value is less than 120 (out of 256 – 8bit), the joystick is being pushed forward for acceleration. This value will be used to calculate a duty cycle for the PWM output. CCP1 will output the PWM signal to the acc system and 5V will be sent to the braking system via ccp2 (no braking action). Conversely, if the joystick is being pulled back for braking, the PWM signal will be sent to the braking system and 0 will be sent to the acceleration system.

The hardware designed to split the original vertical joystick signal in half (acceleration and braking) is no longer necessary. Now, this signal will be sent directly to RA0 of the PIC. The code will do the splitting.

I discussed the steering system with Travis. We concluded that the bi-directional dc motor speed control kit alone will not be sufficient to control the steering. This kit does not control the position of the actuator based on the position of the potentiometer. This kit controls the speed of the actuator to the left or to the right based on the position of the potentiometer. In other words, if you move the wheel all the way to the left, the actuator will quickly move to the left. However, if you move the potentiometer back to the center position, nothing happens. You have to move the potentiometer to the right, past the center position to return the actuator to center. This method of steering is confusing, counter intuitive, and unacceptable. I have been discussing viable solutions with Travis. Travis will try to solve the problem using hardware (mechanical and electrical). If this method becomes impractical, we will have to interface the dc motor speed control kit with a microprocessor.

Finally, I assembled the second DC motor speed control kit that we received for the steering system (as discussed above). I have verified its operation and it seems to working appropriately

**Future Work**

Finish and verify the main.c program to control the braking and acceleration for both modes of operation. I will work with Travis to solve the steering system problem.

**Hours Worked**

Time spent on the project 2/21/2008 – 2/27/2008: 16 hours