This past week I focused on programming the microprocessor to control acceleration and braking in both modes. The joystick mode was completed and reported in my last weekly report. This week I focused on the wheel mode. The current wheel mode section of the main program is seen below (initializations and declarations are excluded):

```c
while (mode_choose==1)  
{

  if (acc_w==1)  
  { 
    for(dc = 0 ; dc < 128 ; dc++)
    {
      CCPR1L = dc;
      CCPR2L = 128;
      DelayMs(10);
    
    do
      CCPR1L = 128;
      CCPR2L = 128;
    while (acc_w==1);
    
  
  if (brake_w==1)  
  { 
    do
      CCPR1L = 0;
      CCPR2L = 128;
    while (brake_w==1);
    
  }

  if (brake_w==1 && acc_w==1)  
  { 
    CCPR1L = 0;
    CCPR2L = 128;
  
  
```

if (brake_w==0 && acc_w==0)
{
    CCPR1L = 0;
    CCPR2L = 128;
}
mode_choose = WIP_switch();
}

This part of the main.c program is controlled by two digital inputs on port b. One input is the acceleration and one is braking. Both buttons output 5V to a pin on port b when they are pressed. Based on these inputs, the program outputs an appropriate signal to the acceleration system and the braking system via ccp1 and ccp2, respectively. The wheel mode code consists of 4 if statements. If both buttons are being pressed simultaneously or if neither button is being pressed, then the acceleration is turned off and the braking actuator is fully extended (brake cable relaxed). These ‘if’ statements are highlighted in yellow. If the acceleration button is pressed, a ramp sequence is run to get the kart up to speed. The braking actuator is held at full extension (brake cable relaxed). This code is highlighted in blue. If the braking button is pressed, a braking sequence forces the brake cable to be contracted relatively slowly. A trip switch will stop this contraction after .25 inches. This code is not highlighted.

The program is mostly working; however there are still a few bugs that need to be worked out. These bugs are described in the future work section.

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. We have decided to interface the dc motor speed control kit with a separate pic microprocessor. Also, some hardware will be required to tell the pic where the actuator is located. This will be accomplished using a differential amplifier. Travis has built the diff amp. I will incorporate it into my protoboard. Travis will work on mechanically connecting the potentiometer to the steering actuator.

Future Work

My biggest concern for the next week will be finalizing the program main.c code. The wheel mode part of this code was described in this report while the joystick mode part of the code was described in last week’s report. The switching code that was previously working now is not working. This problem needs to be debugged. The voltage
ranges being output from the CCP pins need to be modified slightly and verified. The output for the braking actuator needs to be tested with the dc motor speed control kit. A limit switch needs to be obtained from Firgelli Automations to stop the braking actuator after moving 0.25 inches. This needs to be installed and tested with the program.

After I am done with all of this, I will start working on the PIC to be used for steering control. This PIC will be incorporated with another dc motor speed control kit.

**Hours worked**

Time spent on the project 2/27/2008 – 3/5/2008: 19 hours