Special Sensor Report

Geoff Hahn

December 9, 2002

University of Florida

Department of Electrical and Computer Engineering

EEL5666

Intelligent Machines Design Laboratory

Tail Sensor

The tail sensor is composed of 2 things: a rubber snake and a flex sensor (Jameco part # 150551). The flex sensor changes resistance based on how much it is bent. As shown above, the tail is normally bent, meaning that when it is tugged on, it will straighten and the resistance will change. The nominal resistance when it is straight is around 10 k-ohms and the maximum resistance is around 30-40 k-ohms. The tail is connected to a voltage divider circuit (Figure 1) so that the voltage around the tail is what is measured by the A/D converter. When the tail is not being pulled, the value is around 138 and when the tail is tugged on, the A/D value is greater than 140 and increases the more that it is stretched.

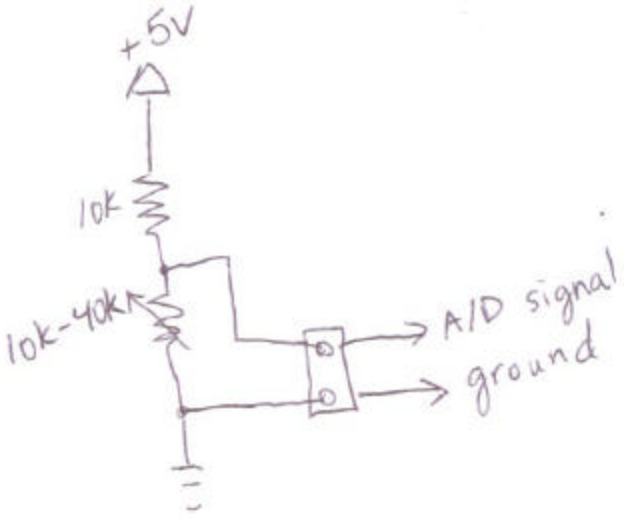

Figure 1

Tail Test Code

```
/*

        Title: tailtest
        Purpose: Test the tail and light LEDs to indicate how much it is being pulled
        A/D interface modified from code initially written by Kristen Allen


*/

#include <io.h>
#include "tailtest.h"



//Initialize the A/D converter
void init_ADC(void) {

        sbi(ADCSR, ADEN); //enable ADC
        sbi(ADMUX, ADLAR); //set ADC to left-adjusted to make ADC 8-bit
        //set the clock divider by 16
        sbi(ADCSR, ADPS2); //ADPS2 set to 1
        cbi(ADCSR, ADPS1); //ADPS1 set to 0
        cbi(ADCSR, ADPS0); //ADPS0 set to 0
        sbi(ADMUX, REFS0); //set to 2.56V reference voltage
}


int main() {

        unsigned char tempValue;
        int i, n;

        init_ADC(); //initialize A/D converter

        sbi(ADMUX, MUX2); //set to channel 4
        cbi(ADMUX, MUX1);
        cbi(ADMUX, MUX0);

for(;;) {

        sbi(ADCSR, ADSC); //start the conversion

        for (n=0; n<4000; n++) { //wait for the conversion to be complete
                i=i+1;
        }

        tempValue = inp(ADCH); //stores value measured by the ADC into value

        outp(0xff,DDRC);            // use all pins on PortC for output

        outp(0xff,PORTC);
```

```
        if(tempValue > 166 && tempValue <= 170) {
                cbi(PORTC,PC7); //turn on led7
        } else if(tempValue > 162 && tempValue <= 166) {
                cbi(PORTC,PC6); //turn on led6
        } else if(tempValue > 158 && tempValue <= 162) {
                cbi(PORTC,PC5); //turn on led5
        } else if(tempValue > 154 && tempValue <= 158) {
                cbi(PORTC,PC4); //turn on led4
        } else if(tempValue > 150 && tempValue <= 154) {
                cbi(PORTC,PC3); //turn on led3
        } else if(tempValue > 146 && tempValue <= 150) {
                cbi(PORTC,PC2); //turn on led2
        } else if(tempValue > 142 && tempValue <= 146) {
                cbi(PORTC,PC1); //turn on led1
        } else if(tempValue > 138 && tempValue <= 142) {
                cbi(PORTC,PC0); //turn on led0
        }
}
        return 0;
}
```