

```
1  /*
2   * Mech-Lab2.c
3   *
4   * Created: 2/12/2020 1:14:53 PM
5   * Author : Megan Shapiro
6   */
7
8  #define F_CPU 16000000L //Clock speed of Arduino 16 MHz
9  #define BAUD 9600 //Desired Baud Rate of Serial Communication
10 #define MYUBRR F_CPU/16/BAUD-1 //Calculated value of UBRR to initialize USART
11 #define TRANSMIT_READY UCSR0A & (1<<UDRE0)
12
13 #include <util/delay.h>
14 #include <avr/io.h>
15 #include <avr/interrupt.h>
16 #include <math.h>
17 #include <stdlib.h>
18 #include "Serial.h"
19 #include "Ring_Buffer.h"
20 #include "Digital_Filter.h"
21
22
23 void timer0_init();
24 void timer1_init();
25 void adc_init();
26 uint16_t adc_read(uint8_t ch);
27
28 union floatChars {
29     float asFloat;
30     char[4] asChars;
31 }
32
33 int main(void)
34 {
35     struct Ring_Buffer_F input_queue;
36     struct Ring_Buffer_C output_queue;
37     USART_Init(MYUBRR);
38     rb_initialize_F(&input_queue);
39     rb_initialize_C(&output_queue);
40
41     timer0_init();
42     timer1_init();
43     adc_init();
44     digital_filter_init(0);
45
46     //Set AI0 to Output and rest as Input
47     DDRC |= 0b00000001;
48
49     //Set output to 1 to power sensor
```

```
50     PORTC |= 0b00000001;
51
52     //Sampling frequency for converting to velocity, 1/0.001
53     float sampPer = 1000;
54     float volt = 0;
55     float angPos = 0;
56     float angPosLast = 0;
57     float angVel = 0;
58     float filteredVel = 0;
59     union floatChars printVal;
60     float convertCoeff[] = {-354.5305, 7.2116, -0.0543, 1.9698E-4, -3.5356E-7, 3.0609E-10, -1.0193E-13};
61     float tempSum = convertCoeff[0];
62
63     while (1)
64     {
65         //if TIMER0_flag
66         if((TIFR0 & (1 << OCF0A)))
67         {
68             printVal.asFloat = filteredVel; //edit so we don't drop readings during prints
69             for(int i = 0; i < 4; i++){
70                 rb_push_back_C(&output_queue, printVal.asChars[i]);
71             }
72             //reset TIMER0_flag
73             TIFR0 |= (1 << OCF0A);
74         }
75         //if TIMER1_flag
76         if((TIFR1 & (1 << OCF1A)))
77         {
78             //read voltage
79             volt = adc_read(1);
80
81             //convert to position in radians
82             // Apply 6th order best fit line found in Matlab
83             for (int i = 1; i <= 6; i++){
84                 tempSum += convertCoeff[i]*volt;
85                 volt *= volt;
86             }
87             //wrap result
88             angPos = abs(tempSum);
89
90             //convert to velocity
91             angVel = (angPos - angPosLast) *0.00277778*sampPer; // rev/s
92
93             //add angPos to queue
94             angPosLast = angPos;
95
96             //filter velocity
```

```
97         filteredVel = filterValue(angVel);
98
99         //reset TIMER1_flag
100         TIFR1 |= (1 << OCF1A);
101     }
102     if (rb_length_C(&output_queue) > 0 && TRANSMIT_READY){
103         print_byte(rb_pop_front_C(&output_queue));
104     }
105 }
106 }
107
108 void timer0_init()
109 {
110     // enable CTC for Timer0
111     TCCR0A |= (1 << WGM01);
112     // enable prescaler of 1024 for Timer0
113     TCCR0B |= (1 << CS02)|(1 << CS00);
114     // initialize counter to zero
115     TCNT0 = 0;
116     // initialize compare value for CTC
117     OCR0A = 155;
118 }
119
120 void timer1_init()
121 {
122     TCCR1A |= 0;
123     // Enable CTC for Timer1 with no prescaler
124     TCCR1B |= (1 << WGM12)|(1 << CS10);
125     // initialize counter to zero
126     TCNT1 = 0;
127     // initialize compare value
128     OCR1B = 15999;
129 }
130
131
132 void adc_init() {
133
134     //Set reference to built in channels
135     ADMUX = (1<<REFS0);
136     //Enable ADC w/ prescaler
137     ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);
138
139 }
140
141 uint16_t adc_read(uint8_t ch)
142 {
143     //select channel to read
144     ch &= 0b00000111;
145     ADMUX = (ADMUX & 0xF8)|ch;
```

```
146     //start conversion
147     ADCSRA |= (1<<ADSC);
148     //wait for conversion to complete
149     while(ADCSRA & (1<<ADSC));
150     //return result
151     return (ADC);
152 }
```