

```

1  /*****
2  Project : mtrsp.c
3  Version : v1.0 for AVR-GCC
4  Date    : 8/4/2003
5  Author  : Chris Troutner
6  Company : MyRobot
7  Chip    : ATMEGA128
8  Platform: Model -T Prototype
9  Comments: This program sends a 9-bit resolution PWM signal at 32Khz to PB4 and PB6 which are the
10             signal lines to the motor driver. The duty cycle of the signal can be set using the 4
11             buttons on the board.
12
13             This program uses an button menu system to let the user select the duty cycle of the PWM signal
14             that gets sent to each track. The following table shows what each button does:
15
16             |-----|
17             |      SW1      | |      SW2      | |      SW3      | |      SW4      | |
18             |-----|
19             | Left Track | | Left Track | | Right Track | | Right Track | |
20             | + Speed   | | - Speed   | | + Speed   | | - Speed   | |
21             |-----|
22
23 Clock frequency      : 16.0000 MHz
24 Memory model        : Small
25 Internal SRAM size   : 4096
26 External SRAM size   : 0
27 Data Stack size     : 1024
28 *****/
29 //Compiler Directives
30
31 //Includes
32 #include <model_t.h>
33
34 //Defines
35
36
37 //Global Variables
38 char t[16];
39 uint8_t sw=0;
40
41 //Sub-Function Prototypes
42
43 //Interrupt Handlers
44
45 INTERRUPT(SIG_OUTPUT_COMPARE1A) {
46     PORTB &= 0xEF;      //Clear PB4
47
48 }
49
50 INTERRUPT(SIG_OUTPUT_COMPARE1B) {
51     PORTB &= 0xBF;      //Clear PB6
52
53 }
54

```

```

55     }
56
57     INTERRUPT(SIG_OUTPUT_COMPARE1C) {
58         PORTB |= 0x50;           //Set PB4 and PB6
59         TCNT1 = 0;
60     }
61
62     //Switch 3
63     INTERRUPT(SIG_INTERRUPT0)
64     {
65         //REMEMBER: SET UP FOR Falling EDGE
66         asm volatile ("cli"); //Disable global interrupts while servicing this interrupt.
67         ms_spin(2);           //Wait 20ms to let mechanical oscillations to die down.
68
69         if(OCR1B > 0)
70             OCR1B--;           //Increment duty cycle for Right Track (Track 2).
71
72         SW = 3;
73
74         EIFR |= 0x0F;          //Clear all the interrupt flags. This instruction makes it so that you
75                                //can't trigger an interrupt from another button until the first button's
76                                //interrupt routine has been serviced.
77
78         asm volatile ("sei"); //Re-enable global interrupts
79     }
80
81     //Switch 4
82     INTERRUPT(SIG_INTERRUPT1)
83     {
84         //REMEMBER: SET UP FOR Falling EDGE
85         asm volatile ("cli"); //Disable global interrupts while servicing this interrupt.
86         ms_spin(2);           //Wait 20ms to let mechanical oscillations to die down.
87
88         if(OCR1B < 511)
89             OCR1B++;           //Increment duty cycle for Right Track (Track 2).
90
91                                //Display new duty cycle.
92         SW = 4;
93
94         EIFR |= 0x0F;          //Clear all the interrupt flags. This instruction makes it so that you
95                                //can't trigger an interrupt from another button until the first button's
96                                //interrupt routine has been serviced.
97
98         asm volatile ("sei"); //Re-enable global interrupts
99     }
100
101     //Switch 2
102     INTERRUPT(SIG_INTERRUPT2)
103     {
104         //REMEMBER: SET UP FOR Falling EDGE
105         asm volatile ("cli"); //Disable global interrupts while servicing this interrupt.
106         ms_spin(2);           //Wait 20ms to let mechanical oscillations to die down.
107
108         if(OCR1A < 511)

```

```

109         OCR1A++;           //Increment duty cycle for Right Track (Track 2).
110
111     SW = 2;
112
113
114     EIFR |= 0x0F;          //Clear all the interrupt flags. This instruction makes it so that you
115                             //can't trigger an interrupt from another button until the first button's
116                             //interrupt routine has been serviced.
117
118     asm volatile ("sei");   //Re-enable global interrupts
119 }
120
121 //Switch 1
122 INTERRUPT(SIG_INTERRUPT3)
123 {
124     //REMEMBER: SET UP FOR Falling EDGE
125     asm volatile ("cli");   //Disable global interrupts while servicing this interrupt.
126     ms_spin(2);             //Wait 20ms to let mechanical oscillations to die down.
127
128     if(OCR1A > 0)
129         OCR1A--;           //Increment duty cycle for Right Track (Track 2).
130
131                             //Display new duty cycle.
132
133     SW = 1;
134
135     EIFR |= 0x0F;          //Clear all the interrupt flags. This instruction makes it so that you
136                             //can't trigger an interrupt from another button until the first button's
137                             //interrupt routine has been serviced.
138
139     asm volatile ("sei");   //Re-enable global interrupts
140 }
141
142 //Main
143 int main(void)
144 {
145     //Local Variables
146
147     //Initialization
148     reset();
149
150     lcd_init();
151
152     //          PERIPHERAL AND INTERRUPT INITIALIZATION
153
154     //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
155     //*****
156     //          Timer0/Counter0 Initialization
157     //*****
158     TCCR1A=0x00;           // Nothing going on in this register.
159     TCCR1B=0x01;           // set clock to 16Mhz
160     TCCR1C=0x00;           // Nothing going on in this register.
161     TCNT1=0;               // This counter buffer is actually composed of 2 8-bit registers TCNT1H & TCNT1L
162
163     // These registers should have a value between 0 and 511. (1 - OCR1n/511) = Duty Cycle %.

```

```

163 OCR1A=384;      // PWM Duty Cycle for Track1
164 OCR1B=384;      // PWM Duty Cycle for Track2
165 OCR1C=511;      // Top of Cycle Window
166
167 // Timer(s)/Counter(s) Interrupt(s) initialization
168 TIMSK=0x18;      //Compare A & B interrupts enabled. (pg. 137 of M128 datasheet)
169 ETIMSK=0x01;     //Compare C interrupt enabled. (pg. 137 of M128 datasheet)
170 //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
171
172 //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
173 //*****
174 //          External Interrupt 0-3 Initialization
175 //*****
176 EICRA=0xAA;      //Set INTO-3 to trigger by a falling edge. (pg. 86 of M128 datasheet)
177 EICRB=0x00;      //This controls INT4-7. (pg. 87 of M128 datasheet)
178 EIMSK=0x0F;      //Enable INTO-3 interrupts (Turn them on). (pg. 88 of M128 datasheet)
179 //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
180
181 //Send initial Duty Cycle Setting
182 strcpy(t, "Change w/ switch");
183 line1(t);
184 sprintf(t, "L: %3.1f R: %3.1f", 100.0 * ((float) OCR1A/512.0), 100.0 * ((float) OCR1B/512.0));
185 line2(t);
186
187 //Enable global interrupts (Keep this instruction at the end of initialization).
188 asm volatile ("sei");
189
190 DDRB |= 0xF0;      //Enable Motor Driver Port
191 //PORTB = (0x50|(PORTB&0x0F)); //Initialize Motor Driver
192 PORTB |= 0xF0;
193
194 //Main Execution Code
195 while (1)
196 {
197     if(sw != 0) {
198         //Display new duty cycle.
199         sprintf(t, "Switch %d", sw);
200         line1(t);
201
202         strcpy(t, " ");
203         line2(t);
204         sprintf(t, "L: %4.2f R: %4.2f", 100.0 * ((float) OCR1A/512.0), 100.0 * ((float) OCR1B/512.0));
205         line2(t);
206
207         sw = 0;
208     }
209 }
210 };
211 }
212
213
214

```