

```

1  /*****
2  Project : led_but.c
3  Version : v1.1 for AVR-GCC
4  Date    : 10/30/2003
5  Author  : Chris Troutner
6  Company : MyRobot
7  Chip    : ATMEGA128
8  Platform: Model -T Prototype
9  Comments: This program shows how to control the LEDs and switches on the Mini-PC.
10
11
12  Clock frequency      : 16.0000 MHz
13  Memory model        : Small
14  Internal SRAM size   : 4096
15  External SRAM size   : 0
16  Data Stack size     : 1024
17  *****/
18
19  //Compiler Directives
20
21  //Includes
22  #include <model_t.h>
23
24  //Defines
25  /*By defining a string constant as is done here, the string is stored in program memory rather than loaded
26  into RAM at run time. The ATmega128 has 128KB of program memory and only 4KB of RAM, so it is much better
27  to store strings in this way if possible. The catch is that strings stored in program memory can't be
28  manipulated (efficiently) during run time.*/
29  #define SW1MSG "Switch 1"
30  #define SW2MSG "Switch 2"
31  #define SW3MSG "Switch 3"
32  #define SW4MSG "Switch 4"
33
34  //Global Variables
35  char t[16];
36
37  //Sub-Function Prototypes
38
39  //Interrupt Handlers
40
41  INTERRUPT(SIG_INTERRUPT0)
42  {
43      //REMEMBER: SET UP FOR Falling EDGE
44      asm volatile ("cli"); //Disable global interrupts while servicing this interrupt.
45      ms_spin(20);           //Wait 20ms to let mechanical oscillations to die down.
46
47      strcpy(t, SW3MSG);     //Send a message to the LCD
48      line1(t);
49
50      led(1, 2000);          //Turn on LED1 for 2 seconds (2000 mS).
51
52      clr_lcd();             //Clear the lcd.
53
54      EIFR |= 0x0F;          //Clear all the interrupt flags. This instruction makes it so that you

```

```

55                                     //can't trigger an interrupt from another button until the first button's
56                                     //interrupt routine has been serviced.
57
58     asm volatile ("sei"); //Re-enable global interrupts
59 }
60
61 INTERRUPT(SIG_INTERRUPT1)
62 {
63     //REMEMBER: SET UP FOR Falling EDGE
64     asm volatile ("cli"); //Disable global interrupts while servicing this interrupt.
65     ms_spin(20);           //Wait 20ms to let mechanical oscillations to die down.
66
67     strcpy(t, SW4MSG);     //Send a message to the LCD
68     line1(t);
69
70     led(1, 2000);          //Turn on LED1 for 2 seconds (2000 mS).
71
72     clr_lcd();             //Clear the lcd.
73
74     EICR |= 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 INTERRUPT(SIG_INTERRUPT2)
82 {
83     //REMEMBER: SET UP FOR Falling EDGE
84     asm volatile ("cli"); //Disable global interrupts while servicing this interrupt.
85     ms_spin(20);           //Wait 20ms to let mechanical oscillations to die down.
86
87     strcpy(t, SW2MSG);     //Send a message to the LCD
88     line1(t);
89
90     led(1, 2000);          //Turn on LED1 for 2 seconds (2000 mS).
91
92     clr_lcd();             //Clear the lcd.
93
94     EICR |= 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 INTERRUPT(SIG_INTERRUPT3)
102 {
103     //REMEMBER: SET UP FOR Falling EDGE
104     asm volatile ("cli"); //Disable global interrupts while servicing this interrupt.
105     ms_spin(20);           //Wait 20ms to let mechanical oscillations to die down.
106
107     strcpy(t, SW1MSG);     //Send a message to the LCD
108     line1(t);

```

```

109         led(1, 2000);           //Turn on LED1 for 2 seconds (2000 mS).
110
111         clr_lcd();              //Clear the lcd.
112
113         EICR |= 0x0F;           //Clear all the interrupt flags. This instruction makes it so that you
114                                   //can't trigger an interrupt from another button until the first button's
115                                   //interrupt routine has been serviced.
116
117         asm volatile ("sei");   //Re-enable global interrupts
118     }
119
120 //Main
121 int main(void)
122 {
123     //Local Variables
124
125     //Initialization
126     reset();
127
128     lcd_init();
129
130     //
131     //          PERIPHERAL AND INTERRUPT INITIALIZATION
132     //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
133     //*****
134     //          External Interrupt 0-3 Initialization
135     //*****
136     EICRA=0xAA;           //Set INTO-3 to trigger by a falling edge. (pg. 86 of M128 datasheet)
137     EICRB=0x00;           //This controls INT4-7. (pg. 87 of M128 datasheet)
138     EIMSK=0x0F;           //Enable INTO-3 interrupts (Turn them on). (pg. 88 of M128 datasheet)
139     //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
140
141
142     //Enable global interrupts (Keep this instruction at the end of initialization).
143     asm volatile ("sei");
144
145     //Main Execution Code
146     while (1)
147     {
148
149     };
150 }
151
152
153
154

```