

main.c

```
1 /*
2 * Welding Visual Effect Version 1.2
3 * COPYRIGHT © 2013 S.D. "Hoffy" Hofmeister
4 * http://www.hoffysworld.com
5 * Provided under a Creative Commons Attribution, Non-Commercial Share Alike,3.0 Unported
  License
6 * TARGETED TO MSP430 LANUCHPAD W/MSP430G2553 PROCESSOR
7 *****
  ****
8 * Design Notes:
9 *
10 * This code uses a random number between 0 and the defined count
11 * to select between 10 defined effects for two high brightness LEDs.
12 *
13 * This has been designed to use two high brightness white light LEDs in order to make the
  effect
14 * more intense but you can operate with one of you choose.
15 *
16 * I have purposely left room for a lot of growth and improvement on this project, have fun!
17 *****
  ****
18 * Circuit Pinout:
19 * PIN 1.0 = Anode of LED #1 > Cathode to Ground
20 * PIN 1.1 = UNASSIGNED - UART
21 * PIN 1.2 = UNASSIGNED - UART
22 * PIN 1.3 = Anode of LED #2 > Cathode to Ground
23 * PIN 1.4 = UNASSIGNED
24 * PIN 1.5 = UNASSIGNED
25 * PIN 1.6 = UNASSIGNED
26 * PIN 1.7 = UNASSIGNED
27 * PIN 2.0 = UNASSIGNED
28 * PIN 2.1 = UNASSIGNED
29 * PIN 2.2 = UNASSIGNED
30 * PIN 2.3 = UNASSIGNED
31 * PIN 2.4 = UNASSIGNED
32 * PIN 2.5 = UNASSIGNED
33 *****
  ****
34 *****
  ****/
35 #include <msp430g2553.h>
36 #include <stdlib.h>
37
38 volatile long reps = 0;
39 volatile long t = 0;
40 unsigned short int res;
41 //-----
42 // Effect Adjustments
43
44 // Modify these numbers to alter effect
45 // XXXXXXXXX_rep = number of flashes
46 // gapX = How long LEDs are on
47 // gapXa = How long LEDs are off
48
49 // Original value comments have been added so you can restore them to the original value
50
51 unsigned short int count = 50; // (must remain higher than 10) Low number effect more often,
```

main.c

```
    higher less often
52 // number higher than 10 creates a greater number of gaps and pauses between the effects
53
54 volatile long one_rep = 20;    //20
55 int gap1 = 500;                //500
56 int gap1a = 1000;              //1000
57
58 volatile long two_rep = 5;     //5
59 int gap2 = 500;                //500
60 int gap2a = 1000;              //1000
61
62 volatile long three_rep = 10;  //10
63 int gap3 = 500;                //500
64 int gap3a = 1000;              //1000
65
66 volatile long four_rep = 15;   //15
67 int gap4 = 250;                //250
68 int gap4a = 500;               //500
69
70 volatile long five_rep = 20;   //10
71 int gap5 = 500;                //500
72 int gap5a = 1000;              //500
73
74 volatile long six_rep = 5;     //5
75 int gap6 = 250;                //250
76 int gap6a = 500;               //500
77
78 volatile long seven_rep = 5;   //5
79 int gap7 = 250;                //250
80 int gap7a = 250;               //500
81
82 volatile long eight_rep = 10;  //10
83 int gap8 = 500;                //500
84 int gap8a = 500;               //500
85
86 volatile long nine_rep = 20;   //20
87 int gap9 = 250;                //250
88 int gap9a = 500;               //500
89
90 volatile long ten_rep = 5;     //5
91 int gap10 = 500;               //500
92 int gap10a = 500;              //500
93 //-----
94 /*
95  * Main Code
96  */
97 void main(void){
98
99     // Configure Pins and stop Watchdog
100
101 WDTCTL = WDTPW + WDTHOLD; //Stop Watchdog Timer
102
103 P1DIR |= BIT0; //Set Port 1 Pin 0 as Output
104 P1OUT &= ~BIT0; //Set Port 1 Pin 0 as OFF
105
106 P1DIR |= BIT3; //Set Port 1 Pin 3 as Output
107 P1OUT &= ~BIT3; //Set Port 1 Pin 3 as OFF
```

main.c

```
108
109 //-----
110 while ( 1 ) { // Continuous Loop
111
112 res = (rand() % count); // returns random value
113
114 switch (res){ // switches between each of the ten choices based on the returned number above
115
116     case 1:
117         for (reps; reps < one_rep; reps++) // repeat choice based on the number
of reps set above
118             {
119                 P1OUT |= BIT0; //ON
120                 P1OUT |= BIT3; //ON
121
122                 for (t; t < gap1; t++) //delay loop
123                 {
124
125
126                     P1OUT &= ~BIT0; //OFF
127                     P1OUT &= ~BIT3; //OFF
128                     t=0; //RESET COUNTER
129
130                     for (t; t < gap1a; t++) //delay loop
131                     {
132
133
134                         t=0; //RESET COUNTER
135                         reps=0; //RESET REPS
136                         break; // Leave choice and get new random number
137
138                     case 2:
139                         for (reps; reps < two_rep; reps++) // repeat choice based on the
number of reps set above
140                         {
141                             P1OUT |= BIT0; //ON
142                             P1OUT |= BIT3; //ON
143
144                             for (t; t < gap2; t++) //delay loop
145                             {
146
147
148
149                                 P1OUT &= ~BIT0; //OFF
150                                 P1OUT &= ~BIT3; //OFF
151                                 t=0; //RESET COUNTER
152                                 for (t; t < gap2a; t++) //delay loop
153                                 {
154
155
156
157                                     t =0; //RESET COUNTER
158                                     reps=0; //RESET REPS
159                                     break; // Leave choice and get new random number
160
161                                 case 3:
162
```

main.c

```
163         for (reps; reps < three_rep; reps++) // repeat choice based on the
number of reps set above
164     {
165         P1OUT |= BIT0; //ON
166         P1OUT |= BIT3; //ON
167
168         for (t; t < gap3; t++) //delay loop
169         {
170         }
171
172         P1OUT &= ~BIT0; //OFF
173         P1OUT &= ~BIT3; //OFF
174         t=0; //RESET COUNTER
175
176         for (t; t < gap3a; t++) //delay loop
177         {
178         }
179         t=0; //RESET COUNTER
180         reps=0; //RESET REPS
181         break; // Leave choice and get new random number
182
183
184
185     case 4:
186         for (reps; reps < four_rep; reps++) // repeat choice based on the
number of reps set above
187     {
188         P1OUT |= BIT0; //ON
189         P1OUT |= BIT3; //ON
190
191         for (t; t < gap4; t++) //delay loop
192         {
193         }
194
195         P1OUT &= ~BIT0; //OFF
196         P1OUT &= ~BIT3; //OFF
197         t=0; //RESET COUNTER
198
199         for (t; t < gap4a; t++) //delay loop
200         {
201         }
202         t=0; //RESET COUNTER
203         reps=0; //RESET REPS
204         break; // Leave choice and get new random number
205
206
207
208     case 5:
209         for (reps; reps < five_rep; reps++) // repeat choice based on the
number of reps set above
210     {
211         P1OUT |= BIT0; //ON
212         P1OUT |= BIT3; //ON
213
214         for (t; t < gap5; t++) //delay loop
215         {
216         }
```

main.c

```
217
218     P1OUT &= ~BIT0; //OFF
219     P1OUT &= ~BIT3; //OFF
220     t=0; //RESET COUNTER
221
222     for (t; t < gap5a; t++)      //delay loop
223     {
224     }
225     }
226     t=0; //RESET COUNTER
227     reps=0; //RESET REPS
228     break; // Leave choice and get new random number
229
230
231     case 6:
232     for (reps; reps < six_rep; reps++) // repeat choice based on the number
of reps set above
233     {
234     P1OUT |= BIT0; //ON
235     P1OUT |= BIT3; //ON
236
237     for (t; t < gap6; t++)      //delay loop
238     {
239     }
240
241     P1OUT &= ~BIT0; //OFF
242     P1OUT &= ~BIT3; //OFF
243     t=0; //RESET COUNTER
244
245     for (t; t < gap6a; t++)      //delay loop
246     {
247     }
248     }
249     t=0; //RESET COUNTER
250     reps=0; //RESET REPS
251     break; // Leave choice and get new random number
252
253
254     case 7:
255     for (reps; reps < seven_rep; reps++) // repeat choice based on the
number of reps set above
256     {
257     P1OUT |= BIT0; //ON
258     P1OUT |= BIT3; //ON
259
260     for (t; t < gap7; t++)      //delay loop
261     {
262     }
263
264     P1OUT &= ~BIT0; //OFF
265     P1OUT &= ~BIT3; //OFF
266     t=0; //RESET COUNTER
267
268     for (t; t < gap7a; t++)      //delay loop
269     {
270     }
271     }
```

main.c

```
272         t=0; //RESET COUNTER
273         reps=0; //RESET REPS
274         break; // Leave choice and get new random number
275
276
277         case 8:
278             for (reps; reps < eight_rep; reps++) // repeat choice based on the
number of reps set above
279             {
280                 P10UT |= BIT0; //ON
281                 P10UT |= BIT3; //ON
282
283                 for (t; t < gap8; t++) //delay loop
284                 {
285                 }
286
287                 P10UT &= ~BIT0; //OFF
288                 P10UT &= ~BIT3; //OFF
289                 t=0; //RESET COUNTER
290
291                 for (t; t < gap8a; t++) //delay loop
292                 {
293                 }
294
295                 t=0; //RESET COUNTER
296                 reps=0; //RESET REPS
297                 break; // Leave choice and get new random number
298
299
300         case 9:
301             for (reps; reps < nine_rep; reps++) // repeat choice based on the
number of reps set above
302             {
303                 P10UT |= BIT0; //ON
304                 P10UT |= BIT3; //ON
305
306                 for (t; t < gap9; t++) //delay loop
307                 {
308                 }
309
310                 P10UT &= ~BIT0; //OFF
311                 P10UT &= ~BIT3; //OFF
312                 t=0; //RESET COUNTER
313
314                 for (t; t < gap9a; t++) //delay loop
315                 {
316                 }
317
318                 t=0; //RESET COUNTER
319                 reps=0; //RESET REPS
320                 break; // Leave choice and get new random number
321
322
323         case 10:
324             for (reps; reps < ten_rep; reps++) // repeat choice based on the
number of reps set above
325             {
```

```

                                main.c
326      P1OUT |= BIT0; //ON
327      P1OUT |= BIT3; //ON
328
329      for (t; t < gap10; t++)    //delay loop
330      {
331      }
332
333      P1OUT &= ~BIT0; //OFF
334      P1OUT &= ~BIT3; //OFF
335      t=0; //RESET COUNTER
336
337      for (t; t < gap10a; t++)    //delay loop
338      {
339      }
340      }
341      t=0; //RESET COUNTER
342      reps=0; //RESET REPS
343      break; // Leave choice and get new random number
344
345 } // END switch
346 } //END while
347 } //END main
348

```