```
//
//
// Gertboard test suite
//
// This program walks the LEDs
//
//
// This file is part of gertboard test suite.
//
//
// Copyright (C) Gert Jan van Loo & Myra VanInwegen 2012
// No rights reserved
// You may treat this program as if it was in the public domain
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
// AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
// ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
// LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
// CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
// SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
// INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
// CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
// ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
// POSSIBILITY OF SUCH DAMAGE.
//
//
// Try to strike a balance between keep code simple for
// novice programmers but still have reasonable quality code
//

// Include some additional gertboard code that supports I/O operations.
#include "gb_common.h"

// Use defines for the LEDS. In the GPIO code, GPIO pins n is controlled
// by bit n. The idea is here is that for example L1 will refer
// to the first LED, which is controlled by GPIO25 (because we will
// put a strap between GP25 anb B1). This gives a more intuitive
// name to use for the LEDs in the patterns.
//
// For novice users: don't worry about the complexity
// The compiler will optimise out all constant expressions and you
// will end up with a single constant value in your table.
#define L1 (1<<25)
#define L2 (1<<24)
#define L3 (1<<23)
#define L4 (1<<22)
// Led 5 is controlled by GP21 - on the rev2 Pi, this is GPIO27, not GPIO21.
#define L5 (1<<27)
#define L6 (1<<18)
#define L7 (1<<17)
#define L8 (1<<11)
#define L9 (1<<10)
#define L10 (1<<9)
#define L11 (1<<8)
#define L12 (1<<7)

// This will be assigned the OR of all the bits corresponding to the
// GPIO pins we are using. It will be used to turn all the LEDs off.
static int ALL_LEDS;

// LEDs test GPIO mapping:
//          Function            Mode
// GPIO0=  unused
// GPIO1=  unused
// GPIO4=  unused
// GPIO7=  LED                 Output
// GPIO8=  LED                 Output
// GPIO9=  LED                 Output
// GPIO10= LED                 Output
// GPIO11= LED                 Output
```

```
72  // GPIO14= unused (preset to be UART)
73  // GPIO15= unused (preset to be UART)
74  // GPIO17= LED                  Output
75  // GPIO18= LED                  Output
76  // GPIO21 (27 on rev2) = LED    Output
77  // GPIO22= LED                  Output
78  // GPIO23= LED                  Output
79  // GPIO24= LED                  Output
80  // GPIO25= LED                  Output
81
82  // Set 12 GPIO pins to output mode
83  void setup_gpio()
84  {
85    INP_GPIO(7);  OUT_GPIO(7);
86    INP_GPIO(8);  OUT_GPIO(8);
87    INP_GPIO(9);  OUT_GPIO(9);
88    INP_GPIO(10);  OUT_GPIO(10);
89    INP_GPIO(11);  OUT_GPIO(11);
90    // 14 and 15 are already set to UART mode
91    // by Linux. Best if we don't touch them
92    INP_GPIO(17);  OUT_GPIO(17);
93    INP_GPIO(18);  OUT_GPIO(18);
94    INP_GPIO(27);  OUT_GPIO(27); // 21 on rev1
95    INP_GPIO(22);  OUT_GPIO(22);
96    INP_GPIO(23);  OUT_GPIO(23);
97    INP_GPIO(24);  OUT_GPIO(24);
98    INP_GPIO(25);  OUT_GPIO(25);
99  } // setup_gpio
100
101 //
102 // Define the various patterns.
103 // The idea here is that each number in the arrays below specifies
104 // a collection of LEDs to turn on. The last element in each array is
105 // -1 so we can run through the patter with a a loop and detect when
106 // we are at the last item in the pattern. pattern0 and pattern1
107 // have only one LED on at a time. pattern2 starts with one on
108 // then turns on 2 of themm then 3, etc. Since each LED is controlled by
109 // a bit, we | (or) them together to turn on more than one LED as a time.
110 //
111
112 static int pattern0[] =
113   {L1, L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, -1 };
114 static int pattern1[] =
115   {L1, L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12,
116    L12, L11, L10, L9, L8, L7, L6, L5, L4, L3, L2, L1, -1 };
117 static int pattern2[] =
118   {0x0,
119    L1,
120    L1|L2,
121    L1|L2|L3,
122    L1|L2|L3|L4,
123    L1|L2|L3|L4|L5,
124    L1|L2|L3|L4|L5|L6,
125    L1|L2|L3|L4|L5|L6|L7,
126    L1|L2|L3|L4|L5|L6|L7|L8,
127    L1|L2|L3|L4|L5|L6|L7|L8|L9,
128    L1|L2|L3|L4|L5|L6|L7|L8|L9|L10,
129    L1|L2|L3|L4|L5|L6|L7|L8|L9|L10|L11,
130    L1|L2|L3|L4|L5|L6|L7|L8|L9|L10|L11|L12,
131    L2|L3|L4|L5|L6|L7|L8|L9|L10|L11|L12,
132    L3|L4|L5|L6|L7|L8|L9|L10|L11|L12,
133    L4|L5|L6|L7|L8|L9|L10|L11|L12,
134    L5|L6|L7|L8|L9|L10|L11|L12,
135    L6|L7|L8|L9|L10|L11|L12,
136    L7|L8|L9|L10|L11|L12,
137    L8|L9|L10|L11|L12,
138    L9|L10|L11|L12,
139    L10|L11|L12,
140    L11|L12,
141    L12,
142    -1};
```

```
143
144
145   // Local (to this file) variables
146   static int *pattern;  // current pattern
147   static int step;  // which pattern element we are showing
148
149
150
151   /*-------------------------------------------------------------------------*/
152   void show_LEDs(int value)
153   {
154     // first turn off all LEDs - GPIO_CLR0 selects which output pins
155     // will be set up 0
156     GPIO_CLR0 = ALL_LEDS;
157     // now light up the ones for this value - GPIO_SET0 selects which
158     // output pins will be set up 1
159     GPIO_SET0 = value;
160   } // set_pattern
161
162   void leds_off(void)
163   {
164     GPIO_CLR0 = ALL_LEDS;
165   }
166
167
168
169   /*-------------------------------------------------------------------------*/
170   //
171   // Start anew with one of the available patterns
172   //
173   void start_new_pattern(int p)
174   {
175      switch (p)
176      {
177      case 0 : pattern = pattern0; break;
178      case 1 : pattern = pattern1; break;
179      case 2 : pattern = pattern2; break;
180      default: return;
181      }
182   } // start_new_pattern
183
184
185
186   /*-------------------------------------------------------------------------*/
187   //
188   // Do single pattern step
189   // return 1 on last pattern
190   // return 0 on all others
191   //
192   int led_step()
193   {
194      step++;
195      show_LEDs(pattern[step]);
196      return (pattern[step+1] == -1) ? 1 : 0; // are we at last value?
197   } // led_step
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
```

```
214  /*-------------------------------------------------------------------------*/
215  //
216  // Quick play all patterns
217  //
218  int main(void)
219  {
220    // define local integer variables
221    int p, r, last, rev, pin21;
222
223    // display instructions to the user
224    printf ("These are the connections for the LEDs test:\n");
225    printf ("jumpers in every out location (U3-out-B1, U3-out-B2, etc)\n");
226    printf ("GP25 in J2 --- B1 in J3\n");
227    printf ("GP24 in J2 --- B2 in J3\n");
228    printf ("GP23 in J2 --- B3 in J3\n");
229    printf ("GP22 in J2 --- B4 in J3\n");
230    printf ("GP21 in J2 --- B5 in J3\n");
231    printf ("GP18 in J2 --- B6 in J3\n");
232    printf ("GP17 in J2 --- B7 in J3\n");
233    printf ("GP11 in J2 --- B8 in J3\n");
234    printf ("GP10 in J2 --- B9 in J3\n");
235    printf ("GP9 in J2 --- B10 in J3\n");
236    printf ("GP8 in J2 --- B11 in J3\n");
237    printf ("GP7 in J2 --- B12 in J3\n");
238    printf ("(If you don't have enough straps and jumpers you can install\n");
239    printf ("just a few of them, then run again later with the next batch.)\n");
240    printf ("When ready hit enter.\n");
241    (void) getchar();
242
243    ALL_LEDS = (L1|L2|L3|L4|L5|L6|L7|L8|L9|L10|L11|L12);
244
245    // set the current pattern to pattern 0
246    pattern = pattern0;
247    step = 0;
248
249    // Map the I/O sections
250    setup_io();
251
252    // Set 12 GPIO pins to output mode
253    setup_gpio();
254
255    // Outer for-loop, loops through patterns 0,1,2
256    for (p=0; p<3; p++)
257    {
258      // for-loop to run pattern several times
259      start_new_pattern(p);
260      for (r=0; r<2; r++)
261      {
262        step = -1; // we will increment this before showing any patterns
263        // inner loop, switches on individual steps of the current pattern
264        do {
265          last = led_step();
266          long_wait(3);
267        } while (!last); // continues till it reaches the end of the pattern (-1)
268      } // run the pattern 2 times
269    } // loop over patterns
270
271    // Perform tidy up operations on the leds and ports.
272    leds_off();
273    restore_io();
274
275  } // main
276
```