```c
/*
* Quad Infrared Detector Version 1.2
* COPYRIGHT 2013 S.D. "Hoffy" Hofmeister
* http://www.hoffysworld.com
* Provided under a Creative Commons Attribution, Non-Commercial Share Alike,3.0 Unported License
* TARGETED TO MSP430 LANUCHPAD W/MSP430G2553 PROCESSOR
*
* Design Notes:
*
* This code is designed to control 4 IR Sensors and light 4 independent indicator LEDS to signal that an object as been detected.
*
* Distance between Tip ofEmitter and Tip of Detector has only been tested upto 3.5 Inches under incandescant and floroescant
* lighting conditions with no failures.
*
* Circuit Pinout:
* PIN 1.0 = Cathode of IR Receiver #1   > Anode to Ground
* PIN 1.1 = UNASSIGNED - UART
* PIN 1.2 = UNASSIGNED - UART
* PIN 1.3 = Cathode of IR Receiver #2   > Anode to Ground
* PIN 1.4 = Cathode of IR Receiver #3   > Anode to Ground
* PIN 1.5 = Cathode of IR Receiver #4   > Anode to Ground
* PIN 1.6 = Anode of Indicator LED #1 \
* PIN 1.7 = Anode of Indicator LED #2  \  Cathodes to ground
* PIN 2.0 = Anode of Indicator LED #3  /
* PIN 2.1 = Anode of Indicator LED #4 /
* PIN 2.2 = UNASSIGNED
* PIN 2.3 = UNASSIGNED
* PIN 2.4 = UNASSIGNED
* PIN 2.5 = Circuit Power Indicator
*
* PINS 1.1, 1.2, 2.2, 2.3, 2.4 are left open for integration into other projects
*
* Note Anodes for the IR Emitters connect to VCC and Cathodes to Ground
*/

#include <msp430g2553.h>

void main(void) {

   WDTCTL = WDTPW + WDTHOLD;        // Stop watchdog timer


   P2DIR |= BIT5;  //  Circuit Power Indicator
   P2OUT |= BIT5;  //  Used to Trouble Shooting


   //Configure IR Detectors

   P1DIR &= ~BIT0;                  // Port 1 P1.0 (IR Detector #1) as input
   P1REN |=  BIT0;                  // Enable Port 1 P1.0 (IR Detector #1) pull-up resistor
   P1SEL &= ~BIT0;                  // Select Port 1 P1.0 (IR Detector #1)

   P1DIR &= ~BIT3;                  // Port 1 P1.3 (IR Detector #2) as input
   P1REN |=  BIT3;                  // Enable Port 1 P1.3 (IR Detector #2) pull-up resistor
```

```
55    P1SEL &= ~BIT3;                 // Select Port 1 P1.3 (IR Detector #2)
56
57    P1DIR &= ~BIT4;                 // Port 1 P1.4 (IR Detector #3) as input
58    P1REN |=  BIT4;                 // Enable Port 1 P1.4 (IR Detector #3) pull-up resistor
59    P1SEL &= ~BIT4;                 // Select Port 1 P1.4 (IR Detector #3)
60
61    P1DIR &= ~BIT5;                 // Port 1 P1.5 (IR Detector #4) as input
62    P1REN |=  BIT5;                 // Enable Port 1 P1.5 (IR Detector #4) pull-up resistor
63    P1SEL &= ~BIT5;                 // Select Port 1 P1.5 (IR Detector #4)
64
65    //Configure Outputs
66
67    P1DIR |= BIT6;                  // Port 1 P1.6 (Indicator #1) as output
68    P1OUT &= ~BIT6;                 // Port 1 P1.6 (Indicator #1) Set to off State
69
70    P1DIR |= BIT7;                  // Port 1 P1.7 (Indicator #1) as output
71    P1OUT &= ~BIT7;                 // Port 1 P1.7 (Indicator #1) Set to off State
72
73    P2DIR |= BIT0;                  // Port 2 P2.0 (Indicator #1) as output
74    P2OUT &= ~BIT0;                 // Port 1 P2.0 (Indicator #1) Set to off State
75
76
77    P2DIR |= BIT1;                  // Port 2 P2.1 (Indicator #1) as output
78    P2OUT &= ~BIT1;                 // Port 2 P2.1 (Indicator #1) Set to off State
79
80 // Let's Get Down to Business
81
82   while( 1 ) // While this value remains equal to 1 the code will loop continuously, there is
     no code written to
83 // change this state in this particular program.
84    {
85
86        //Detector #1
87        if( (P1IN & BIT0) > 0)        // When IR Detector #1 detects an object breaking the IR
     Beam
88
89           P1OUT |= BIT6;             // Set LED Indicator #1 to ON
90
91                    else              // Otherwise
92
93           P1OUT &= ~BIT6;            // Set LED Indicator #1 to OFF
94        //END of Detector #1
95
96
97        //Detector #2
98        if( (P1IN & BIT3) > 0)       // When IR Detector #2 detects an object breaking the IR
     Beam
99
100                P1OUT |= BIT7;             // Set LED Indicator #2 to ON
101
102                         else              // Otherwise
103
104                P1OUT &= ~BIT7;            // Set LED Indicator #2 to OFF
105       //END of Detector #2
106
107
108
```

```
109         //Detector #3
110         if( (P1IN & BIT4) > 0)      // When IR Detector #3 detects an object breaking the IR
    Beam
111
112                     P2OUT |= BIT0;              // Set LED Indicator #3 to ON
113
114                             else                // Otherwise
115
116                     P2OUT &= ~BIT0;             // Set LED Indicator #3 to OFF
117         //END of Detector #3
118
119
120
121         //Detector #4
122         if( (P1IN & BIT5) > 0)       // When IR Detector #4 detects an object breaking the IR
    Beam
123
124                     P2OUT |= BIT1;              // Set LED Indicator #4 to ON
125
126                             else                // Otherwise
127
128                     P2OUT &= ~BIT1;             // Set LED Indicator #4 to OFF
129         //End of Detector #4
130
131
132     } End of While
133
134 }   // END OF MAIN
135
136
137
```