TRAFFIC LIGHT CONTOL WITH INTERUPTS

```
#include <lpc213x.h>

void delay(unsigned int count) {
   unsigned int i = 0, j = 0;
   for (i = 0; i < count; i++) {
      for (j = 0; j < 1000; j++);
   }
}

int main() {
   PINSEL0 = 0;
   IODIR0 = 0X003FFFFF;

   while (1) {
      if ((IOPIN0 & (1 << 22)) == 0) {
         IOSET0 |= (1 << 19);
         IOCLR0 |= (1 << 7) | (1 << 8) | (1 << 9) | (1 << 10) | (1 << 11) | (1 << 12) | (1 << 13);

         // segment A //9
         IOSET0 |= (1 << 18);
         IOCLR0 |= (1 << 16);
         IOSET0 |= (1 << 0) | (1 << 1) | (1 << 2) | (1 << 3) | (1 << 5) | (1 << 6);
         IOCLR0 |= (1 << 4);
         delay(300);

         // segment A //8
         IOSET0 |= (1 << 0) | (1 << 1) | (1 << 2) | (1 << 3) | (1 << 4) | (1 << 5) | (1 << 6);
         delay(300);
```

```
// segment A //7
IOSET0 |= (1 << 0) | (1 << 1) | (1 << 2);
IOCLR0 |= (1 << 3) | (1 << 4) | (1 << 5) | (1 << 6);
delay(300);

// segment A //6
IOSET0 |= (1 << 0) | (1 << 2) | (1 << 3) | (1 << 4) | (1 << 5) | (1 << 6);
IOCLR0 |= (1 << 1);
delay(300);

// segment A //5
IOSET0 |= (1 << 0) | (1 << 2) | (1 << 3) | (1 << 5) | (1 << 6);
IOCLR0 |= (1 << 1) | (1 << 4);
delay(200);
IOCLR0 |= (1 << 18);
delay(100);

// segment A //4
IOSET0 |= (1 << 18) | (1 << 1) | (1 << 2) | (1 << 5) | (1 << 6);
IOCLR0 |= (1 << 0) | (1 << 3) | (1 << 4);
delay(200);
IOCLR0 |= (1 << 18);
delay(100);

// segment A //3
IOSET0 |= (1 << 18) | (1 << 14) | (1 << 0) | (1 << 1) | (1 << 2) | (1 << 3) | (1 << 6);
IOCLR0 |= (1 << 4) | (1 << 5);
delay(200);
IOCLR0 |= (1 << 18) | (1 << 14);
delay(100);
```

```
// segment A //2
IOSET0 |= (1 << 18) | (1 << 14) | (1 << 0) | (1 << 1) | (1 << 3) | (1 << 4) | (1 << 6);
IOCLR0 |= (1 << 2) | (1 << 5);
delay(200);
IOCLR0 |= (1 << 18) | (1 << 14);
delay(100);

// segment A //1
IOSET0 |= (1 << 18) | (1 << 14) | (1 << 1) | (1 << 2);
IOCLR0 |= (1 << 0) | (1 << 3) | (1 << 4) | (1 << 5) | (1 << 6);
delay(200);
IOCLR0 |= (1 << 18) | (1 << 14);
delay(100);

// segment A //0
IOSET0 |= (1 << 18) | (1 << 14) | (1 << 0) | (1 << 1) | (1 << 2) | (1 << 3) | (1 << 4) | (1 << 5);
IOCLR0 |= (1 << 6);
delay(200);
IOCLR0 |= (1 << 18) | (1 << 14);
delay(100);

// YELLOW A
IOSET0 |= (1 << 17);
IOCLR0 |= (1 << 0) | (1 << 1) | (1 << 2) | (1 << 3) | (1 << 4) | (1 << 5) | (1 << 6);
delay(300);
IOCLR0 |= (1 << 17) | (1 << 19);

IOSET0 |= (1 << 16);
IOCLR0 |= (1 << 0) | (1 << 1) | (1 << 2) | (1 << 3) | (1 << 4) | (1 << 5) | (1 << 6);
```

```
// SEGMENT B //9
IOSET0 |= (1 << 21);
IOCLR0 |= (1 << 15);
IOSET0 |= (1 << 7) | (1 << 8) | (1 << 9) | (1 << 10) | (1 << 12) | (1 << 13);
IOCLR0 |= (1 << 11);
delay(300);

// segment B //8
IOSET0 |= (1 << 7) | (1 << 8) | (1 << 9) | (1 << 10) | (1 << 11) | (1 << 12) | (1 << 13);
delay(300);

// segment B //7
IOSET0 |= (1 << 7) | (1 << 8) | (1 << 9);
IOCLR0 |= (1 << 10) | (1 << 11) | (1 << 12) | (1 << 13);
delay(300);

// segment B //6
IOSET0 |= (1 << 7) | (1 << 9) | (1 << 10) | (1 << 11) | (1 << 12) | (1 << 13);
IOCLR0 |= (1 << 8);
delay(300);

// segment B //5
IOSET0 |= (1 << 7) | (1 << 9) | (1 << 10) | (1 << 12) | (1 << 13);
IOCLR0 |= (1 << 8) | (1 << 11);
delay(200);
IOCLR0 |= (1 << 21);
delay(100);

// segment B //4
IOSET0 |= (1 << 21) | (1 << 8) | (1 << 9) | (1 << 12) | (1 << 13);
IOCLR0 |= (1 << 7) | (1 << 10) | (1 << 11);
```

```
delay(200);
IOCLR0 |= (1 << 21);
delay(100);

// segment B //3
IOSET0 |= (1 << 21) | (1 << 15) | (1 << 7) | (1 << 8) | (1 << 9) | (1 << 10) | (1 << 13);
IOCLR0 |= (1 << 11) | (1 << 12);
delay(200);
IOCLR0 |= (1 << 21) | (1 << 15);
delay(100);

// segment B //2
IOSET0 |= (1 << 21) | (1 << 15) | (1 << 7) | (1 << 8) | (1 << 10) | (1 << 11) | (1 << 13);
IOCLR0 |= (1 << 9) | (1 << 12);
delay(200);
IOCLR0 |= (1 << 21) | (1 << 15);
delay(100);

// segment B //1
IOSET0 |= (1 << 21) | (1 << 15) | (1 << 8) | (1 << 9);
IOCLR0 |= (1 << 7) | (1 << 10) | (1 << 11) | (1 << 12) | (1 << 13);
delay(200);
IOCLR0 |= (1 << 21) | (1 << 15);
delay(100);

// segment B //0
IOSET0 |= (1 << 21) | (1 << 15) | (1 << 7) | (1 << 8) | (1 << 9) | (1 << 10) | (1 << 11) | (1 << 12);
IOCLR0 |= (1 << 13);
delay(200);
IOCLR0 |= (1 << 21) | (1 << 15);
delay(100);
```

```
            // YELLOW B
            IOSET0 |= (1 << 20);
            IOCLR0 |= (1 << 7) | (1 << 8) | (1 << 9) | (1 << 10) | (1 << 11) | (1 << 12) | (1 << 13);
            delay(500);
            IOCLR0 |= (1 << 20) | (1 << 16);
        }
        else if ((IOPIN0 & (1 << 23)) == 0) {
            IOCLR0 |= 0x003FFFFF;  // Clear all bits from 0 to 21
        }
    }
}
```