

```

1  #include <lpc214x.h>
2  #define LED_OFF (IO0SET = 1U << 31)
3  #define LED_ON (IO0CLR = 1U << 31)
4
5  #define PLOCK 0x00000400 // Bit mask for checking PLL lock status
6
7  void systeminit(void);
8  void delay_ms(unsigned int t);
9
10
11
12 void delay_ms(unsigned int t){
13     unsigned int i,j;
14     for(i=0; i<t; i++)
15         for(j=0; j<10000; j++);
16 }
17 #define SW2 (IO0PIN & (1 << 14))
18 #define SW3 (IO0PIN & (1 << 15))
19 #define SW4 (IO1PIN & (1 << 18))
20 #define SW5 (IO1PIN & (1 << 19))
21 #define SW6 (IO1PIN & (1 << 20))
22
23 short int sine_table[ ] = {512+0,512+53,512+106,512+158,512+208,512+256,512+300,512+342,512+380,512+413,
512+442,
24                             512+467,512+486,512+503,512+510,512+511,512+510,512+503,512+486,512+467,512+
442,512+413,
25                             512+380,512+342,512+300,512+256,512+208,512+158,512+106,512+53,512+0,512-53,
512-106,512-158,
26                             512-208,512-256,512-300,512-342,512-380,512-413,512-442,512-467,512-486,512-
503,512-510,512-511,
27                             512-510,512-503,512-486,512-467,512-442,512-413,512-380,512-342,512-300,512-
256,512-208,512-158,
28                             512-106,512-53};
29
30 short int sine_rect_table[ ] = {512+0,512+53,512+106,512+158,512+208,512+256,512+300,512+342,512+380,512
+413,
31                             512+442,512+467,512+486,512+503,512+510,512+511,512+510,512+503,512+486,
512+467,
32                             512+442,512+413,512+380,512+342,512+300,512+256,512+208,512+158,512+106,
512+53,512+0};
33
34 int main(){
35     short int value,i=0;
36     systeminit();
37     PINSEL1 |= 0x00080000; /* P0.25 as DAC output :option 3 - 10 (bits18,19)*/
38     IOODIR |= 1U << 31 | 0x00FF0000 ; // to set P0.16 to P0.23 as o/ps
39     while(1){
40         if (!SW2) /* If switch for sine wave is pressed */
41         {
42             while (i!=60 ){
43                 value = sine_table[i++];
44                 DACR = ( (1<<16) | (value<<6) );
45                 delay_ms(1);
46             }
47             i=0;
48         }
49
50         else if (!SW3){
51             while ( i!=30 ){
52                 value = sine_rect_table[i++];
53                 DACR = ( (1<<16) | (value<<6) );
54                 delay_ms(1);
55             }
56             i=0;
57         }
58
59         else if ( !SW4) /* If switch for triangular wave is pressed */
60         {
61             value = 0;
62             while ( value != 1023 ){
63                 DACR = ( (1<<16) | (value<<6) );
64                 value++;

```

```
65     }
66     while ( value != 0 ){
67         DACR = ( (1<<16) | (value<<6) );
68         value--;
69     }
70 }
71
72 else if ( !SW5 ) /* If switch for sawtooth wave is pressed */
73 {
74     value = 0;
75     while ( value != 1023 ){
76         DACR = ( (1<<16) | (value<<6) );
77         value++;
78     }
79 }
80
81 else if ( !SW6 ) /* If switch for square wave is pressed */
82 {
83     value = 1023;
84     DACR = ( (1<<16) | (value<<6) );
85     delay_ms(1);
86     value = 0;
87     DACR = ( (1<<16) | (value<<6) );
88     delay_ms(1);
89 }
90 else /* If no switch is pressed, 3.3V DC */
91 {
92     value = 1023;
93     DACR = ( (1<<16) | (value<<6) );
94 }
95 }
96 }
97
98
99 void systeminit(void) {
100     PLL0CON = 0x01;      // Enable the PLL (PLLE = 1)
101     PLL0CFG = 0x24;      // Set the multiplier and divider values (M=5, P=2)
102     PLL0FEED = 0xAA;     // Sequence to update PLL registers
103     PLL0FEED = 0x55;
104
105     while (!(PLL0STAT & PLOCK)); // Wait for the PLL to achieve lock
106
107     PLL0CON = 0x03;      // Connect the PLL (PLLE = 1 and PLLC = 1)
108     PLL0FEED = 0xAA;     // Sequence to update PLL registers after connecting
109     PLL0FEED = 0x55;
110
111     VPBDIV = 0x01;       // Set PCLK = CCLK (PCLK = 60 MHz if CCLK is 60 MHz)
112 }
113
114
```