**1.LED BLINKING**

```
#include "lpc214x.h"
void delay (unsigned int k);
void main(void)
{
IODIR0 = 0xFFFFFFFF;
PINSEL0 = 0;
while(1)
{
IOSET0 = 0x0000ff00;
delay(1000);
IOCLR0 = 0x0000ff00;
delay(1000);
}
}
void delay(unsigned int k)
{
unsigned int i,j;
for (j=0; j<k; j++)
for(i = 0; i<=800; i++);
}
```

**2. DISPLAY ALPHA**

```
#include <LPC214X.H>
#define DS3 1<<13
#define DS4 1<<12
#define SEG_CODE 0xFF<<16
unsigned char const
seg_alphabet[] = {
 0x77,
 0x7C,
 0x39,
 0x5E,
 0x79,
 0x71
};
void delayms(int n) {
 int i, j;
 for(i = 0; i < n; i++) {
 for(j = 0; j < 5035; j++) {;}  }
}
int main(void) {
 int count;
 PINSEL0 = 0;
 PINSEL1 = 0;
 IODIR0 = SEG_CODE | DS3 |
DS4;
 IOSET0 = SEG_CODE | DS3;
IOCLR0 = DS4; while (1) {
 for (count = 0; count < 6;
count++) {
 IOCLR0 = SEG_CODE;
 IOSET0 = seg_alphabet[count]
<< 16;
 delayms(1000); // 1 sec delay
 }
 }
}
```

**3. DISPLAY NUMBERS**

```
#include <LPC214X.H>
#define DS3 1<<13 // P0.13
#define DS4 1<<12 // P0.12
#define SEG_CODE 0xFF<<16
unsigned char const seg_decimal[]
= {0x3F, 0x06, 0x5B, 0x4F, 0x66,
0x6D, 0x7D, 0x07,
0x7F, 0x6F};
void delayms(int n) {
 int i, j;
 for(i = 0; i < n; i++) {
 for(j = 0; j < 5035; j++) {;}
 }
}
int main(void) {
 int count;
 PINSEL0 = 0;
PINSEL1 = 0; IODIR0 =
SEG_CODE | DS3 | DS4 IOSET0
= SEG_CODE | DS3; // Disable
DS3 display
 IOCLR0 = DS4; while (1) {
 for (count = 0; count < 10;
count++) {
 IOCLR0 = SEG_CODE;
 IOSET0 = seg_decimal[count] <<
16;
 delayms(1000);
 }
 }
}
```

**4. DISPLAY HEXA DECIMAL**

```
#include <LPC214X.H>
#define DS3 1<<13
#define DS4 1<<12 #define
SEG_CODE 0xFF<<16 unsigned
char const seg_hexadecimal[] = {
 0x3F,
 0x06,
 0x5B,
 0x4F,
 0x66,
 0x6D,
 0x7D,
 0x07,
 0x7F,
 0x6F,
 0x77,
 0x7C,
 0x39,
 0x5E,
 0x79,
 0x71
};
void delayms(int n) {
 int i, j;
 for(i = 0; i < n; i++) {
 for(j = 0; j < 5035; j++) {;}  }
}
int main(void) {
 int count;
 PINSEL0 = 0;
 PINSEL1 = 0; IODIR0 =
SEG_CODE | DS3 | DS4;
 IOSET0 = SEG_CODE | DS3;
IOCLR0 = DS4; // Enable DS4
Display
 while (1) {
 for (count = 0; count < 16;
count++) {
 IOCLR0 = SEG_CODE;
 IOSET0 =
seg_hexadecimal[count] << 16;
 delayms(1000); // 1 sec delay
 }
 }
}
```

**5. SQUARE WAVE**

```
#include <lpc214x.h>
void delay(unsigned int count);
void generate_square_wave(void);
int main(void) {
 PINSEL1 |= (1 << 19);
 while (1) {
generate_square_wave();delay(500
00);
 }
}
void delay(unsigned int count) {
 unsigned int i, j;
 for (i = 0; i < count; i++) {
 for (j = 0; j < 6000; j++);  }
}
void generate_square_wave(void)
{
 unsigned int high = 1023 << 6;
 unsigned int low = 0 << 6; for (int
i = 0; i < 100; i++) {
 DACR = high; delay(10000);
DACR = low; delay(10000);
}
}
```

**6. TRIANGULAR**

```
#include <lpc214x.h>
void delay(unsigned int count);
void
generate_triangle_wave(void);
int main(void) {
PINSEL1 |= (1 << 19);
 while (1) {
 generate_triangle_wave();
 delay(50000);
 }
}
void delay(unsigned int count) {
 unsigned int i, j;
 for (i = 0; i < count; i++) {
 for (j = 0; j < 6000; j++);
 }
}
void generate_triangle_wave(void)
{
 unsigned int i;
for (i = 0; i < 1023; i++) {
 DACR = (i << 6);
 delay(100);
 }
 for (i = 1023; i > 0; i--) {
 DACR = (i << 6);
 delay(100);
 }
}
```