```c
#include <lpc214x.h>
#define LED_OFF (IO0SET = 1U << 31)
#define LED_ON (IO0CLR = 1U << 31)

#define PLOCK 0x00000400  // Bit mask for checking PLL lock status

void systeminit(void);
void delay_ms(unsigned int t);

unsigned char getAlphaCode(unsigned char alpha);
void alphadisp7SEG(char* buf);

int main(){
    IO0DIR |= (1U<<31) | (1U<<19) | (1U<<20) | (1U<<30);
    LED_ON;
    delay_ms(500);
    LED_OFF;
    while(1){
      alphadisp7SEG("FIRE ");
      delay_ms(500);
      alphadisp7SEG("     ");
      delay_ms(500);
    }
}

unsigned char getAlphaCode(unsigned char alpha){
    switch(alpha){
      case 'F': return 0x8e;
      case 'i': return 0xf9;
      case 'r': return 0xce;
      case 'e': return 0x86;
      case ' ': return 0xff;
      default: break;
    }
  return 0xff;
}


void alphadisp7SEG(char* buf){
    unsigned char data, count, i, j;
    for(i=0; i<5; i++){
      data = getAlphaCode(*(buf +i));
      for(j=0; j<8; j++){
        count = (data & (0x80));
        if(count != 0){
          IO0SET |= 1<<19;
        }
        else
          IO0CLR |= 1<<19;
        IO0SET |= 1<<20;
        delay_ms(1);          //since it is not for bouncing effect, 1ms delay is ecough for clock pulse
        IO0CLR = 1<<20;
        data = (data<<1);
      }
    }

    //strobe P0.30
    IO0SET |= 1U<<30;
    delay_ms(1);
    IO0CLR = 1U<<30;
}


void delay_ms(unsigned int t){
  unsigned int i,j;
  for(i=0; i<t; i++)
    for(j=0; j<10000; j++);
}


void systeminit(void) {
    PLL0CON = 0x01;      // Enable the PLL (PLLE = 1)
```

```
73        PLL0CFG = 0x24;       // Set the multiplier and divider values (M=5, P=2)
74        PLL0FEED = 0xAA;      // Sequence to update PLL registers
75        PLL0FEED = 0x55;
76
77        while (!(PLL0STAT & PLOCK));  // Wait for the PLL to achieve lock
78
79        PLL0CON = 0x03;       // Connect the PLL (PLLE = 1 and PLLC = 1)
80        PLL0FEED = 0xAA;      // Sequence to update PLL registers after connecting
81        PLL0FEED = 0x55;
82
83        VPBDIV = 0x01;        // Set PCLK = CCLK (PCLK = 60 MHz if CCLK is 60 MHz)
84    }
85
```