```
// led_blink_control.c
// for NerdKits with ATmega168 change to 328
// Use this file as bases for output control for relay through driver transistor
#define F_CPU 14745600
#include <stdio.h>

#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#include <inttypes.h>
// line 11
#include "../libnerdkits/delay.h"
#include "../libnerdkits/lcd.h"

// PIN DEFINITIONS:
//
// PC2, 3, 4, 5 -- LED anode
// line 19
int main() {
int time_delay = 500;


  // LED as output if I can control an led i can control a relay
  // do not need to have as many outputs
  DDRC |= (1<<PC2);
  DDRC |= (1<<PC3);
  DDRC |= (1<<PC4);
  DDRC |= (1<<PC5);
  DDRB |= (1<<PB1);
  DDRB |= (1<<PB2);

  // loop keeps looking forever
  while(1) {
  // if time_delay <<100 then time_delay = 2000;
  // time_delay = time_delay - 100
    // turn on LED PC2 & PC4  line 30
    PORTC |= (1<<PC2);
    PORTC |= (1<<PC4);
    PORTB |= (1<<PB1);
    // turn off led in PC3 &PC5
    PORTC &= ~(1<<PC3);
    PORTC &= ~(1<<PC5);
    PORTB &= ~(1<<PB2);
// fire up the LCD
  lcd_init();
  lcd_home();
  // print message to screen   line 40
  //                           20 columns wide:
  //                   01234567890123456789
  lcd_line_one();
  lcd_write_string(PSTR(" Reversing  LEDS "));
  lcd_line_two();
  lcd_write_string(PSTR("I can do this "));
  lcd_line_three();
  lcd_write_string(PSTR("  "));
  lcd_line_four();
  lcd_write_string(PSTR(" "));
    //delay for time_delay time  to let the light stay on line 51
    delay_ms(time_delay);
 // fire up the LCD
  lcd_init();
  lcd_home();

  // print message to screen
  //                           20 columns wide:
  //                   01234567890123456789  line 59
  lcd_line_one();
  lcd_write_string(PSTR(" Reversing again "));
  lcd_line_two();
  lcd_write_string(PSTR("  "));
  lcd_line_three();
  lcd_write_string(PSTR("    "));
  lcd_line_four();
  lcd_write_string(PSTR(""));
    // turn off LED
    PORTC &= ~(1<<PC2);
    PORTC &= ~(1<<PC4);
    // turn on led in PC 3&PC5  line 71
    PORTC l= (1<<PC3);
    PORTC l= (1<<PC5);
        PORTB &= ~(1<<PB1);
        PORTB l= (1<<PB2);
    //delay for time_delay time to let the light stay off
    delay_ms(time_delay);

  }

  return 0;
}
```