```c
#include <lpc214x.h>

// Macros for controlling the LED
#define LED_OFF (IO0SET = 1U << 31)
#define LED_ON (IO0CLR = 1U << 31)
#define PLOCK 0x00000400

// Function declarations
void delay_ms(unsigned int j);
void SystemInit(void);
void runDCMotor(int direction, int dutycycle);
unsigned int adc(int no, int ch);

int main() {
    int dig_val;
    // Configure P0.16 to P0.23 as outputs and LED pin (P0.31) as output
    IO0DIR |= 1U << 31 | 0x00FF0000 | 1U << 30;
    LED_ON;
    delay_ms(500);
    LED_OFF;

    SystemInit();  // Initialize system clock

    // Main loop to control the DC motor
    while (1) {
        // Read the potentiometer value using ADC channel 2 of ADC1
        dig_val = adc(1, 2) / 10;
        if (dig_val > 100) dig_val = 100; // Limit the duty cycle to 100%

        // Run the motor with the read duty cycle
        runDCMotor(2, dig_val); // Example direction is 2 (clockwise)
    }
}

void runDCMotor(int direction, int dutycycle) {
    IO0DIR |= 1U << 28; // Set P0.28 as output pin for direction control
    PINSEL0 |= 2 << 18; // Select P0.9 as PWM6 (Option 2)

    // Set direction: 1 for anti-clockwise, 0 for clockwise
    if (direction == 1) {
        IO0SET = 1U << 28; // Set to 1 for anti-clockwise
    } else {
        IO0CLR = 1U << 28; // Set to 0 for clockwise
    }

    // Configure PWM for controlling the motor speed
    PWMPCR = (1 << 14); // Enable PWM6
    PWMMR0 = 1000; // Set PWM period (frequency of the PWM signal)
    PWMMR6 = (1000U * dutycycle) / 100; // Set duty cycle based on input (0-100%)
    PWMTCR = 0x00000009; // Enable PWM and start the timer
    PWMLER = 0x40; // Load the new values into PWMMR0 and PWMMR6
}

unsigned int adc(int no, int ch) {
    unsigned int val;

    // Configure the ADC pin functions based on the channel
    switch (no) {
        case 0: // ADC0
            PINSEL1 |= (1 << (ch * 2)); // Select the corresponding ADC0 pin
            AD0CR = 0x00200600 | (1 << ch); // Set up ADC0 with 10-bit mode and select channel
            AD0CR |= (1 << 24); // Start conversion
            while ((AD0GDR & (1U << 31)) == 0); // Wait for conversion to complete
            val = AD0GDR;
            break;
        case 1: // ADC1
            PINSEL1 |= (1 << (ch * 2 + 16)); // Select the corresponding ADC1 pin
            AD1CR = 0x00200600 | (1 << ch); // Set up ADC1 with 10-bit mode and select channel
            AD1CR |= (1 << 24); // Start conversion
            while ((AD1GDR & (1U << 31)) == 0); // Wait for conversion to complete
            val = AD1GDR;
            break;
```

```
73            default:
74                val = 0; // Default value in case of an invalid ADC number
75                break;
76        }
77
78        val = (val >> 6) & 0x03FF; // Extract the 10-bit result (bits 6-15)
79        return val;
80    }
81
82    void SystemInit(void) {
83        // Initialize the system PLL and clock
84        PLL0CON = 0x01;
85        PLL0CFG = 0x24;
86        PLL0FEED = 0xAA;
87        PLL0FEED = 0x55;
88        while (!(PLL0STAT & PLOCK)) { ; } // Wait for the PLL to lock
89        PLL0CON = 0x03;
90        PLL0FEED = 0xAA;
91        PLL0FEED = 0x55; // Lock the PLL registers
92        VPBDIV = 0x01; // PCLK is the same as CCLK (60 MHz)
93    }
94
95    void delay_ms(unsigned int j) {
96        // Simple delay loop for milliseconds
97        unsigned int x, i;
98        for (i = 0; i < j; i++) {
99            for (x = 0; x < 10000; x++);
100       }
101   }
102
```