Elevator Program  
#include <lpc214x.h>

#define LED\_ON (IO0CLR = 1U << 31)

#define LED\_OFF (IO0SET = 1U << 31)

void delay\_ms(unsigned int j);

void elevator\_run(void) {

int i, val;

unsigned int counter;

IO1CLR = 1 << 24; // Enable elevator section

IO0CLR = 0x000F0000; // Set ground floor LED

while (1) {

IO0CLR = 0x00F00000; // Clear latches

IO0SET = 0x00F00000;

do {

counter = (IO1PIN >> 16) & 0xF;

} while (counter == 0xF);

switch (counter) {

case 0xE: val = 3; break; // Floor 1

case 0xD: val = 6; break; // Floor 2

case 0xB: val = 8; break; // Floor 3

case 0x7: val = 10; break; // Floor 4

default: val = 0; break;

}

for (i = 0; i < val; i++) { // Move up

IO0CLR = 0x000F0000;

IO0SET = i << 16;

delay\_ms(250);

}

for (i = val - 1; i >= 0; i--) { // Move down

IO0CLR = 0x000F0000;

IO0SET = i << 16;

delay\_ms(250);

}

}

}

int main() {

IO0DIR |= (1U << 31) | (0xFF << 16); // P0.31, P0.16–P0.23 as outputs

IO1DIR |= 1 << 24; // P1.24 as output

LED\_ON;

elevator\_run();

while (1);

}

void delay\_ms(unsigned int j) {

for (unsigned int x = 0; x < j \* 6000; x++);

}

Seven Segment  
#include <lpc214x.h>

#define LED\_ON (IO0CLR = 1U << 31)

#define LED\_OFF (IO0SET = 1U << 31)

#define PLOCK 0x00000400

void delay\_ms(unsigned int j);

void SystemInit(void);

unsigned char getAlphaCode(unsigned char ch);

void alphadisp7SEG(char \*buf);

int main() {

IO0DIR |= (1U << 31) | (1U << 19) | (1U << 20) | (1U << 30); // P0.19, 20, 30, 31 as output

LED\_ON;

SystemInit();

while (1) {

alphadisp7SEG("fire ");

delay\_ms(500);

alphadisp7SEG("help ");

delay\_ms(500);

}

}

unsigned char getAlphaCode(unsigned char ch) {

switch (ch) {

case 'f': return 0x8E;

case 'i': return 0xF9;

case 'r': return 0xCE;

case 'e': return 0x86;

case 'h': return 0x89;

case 'l': return 0xC7;

case 'p': return 0x8C;

case ' ': return 0xFF;

default : return 0xFF;

}

}

void alphadisp7SEG(char \*buf) {

for (int i = 0; i < 5; i++) {

unsigned char seg = getAlphaCode(buf[i]);

for (int j = 0; j < 8; j++) {

(seg & 0x80) ? (IOSET0 |= 1 << 19) : (IOCLR0 |= 1 << 19);

IOSET0 |= 1 << 20;

delay\_ms(1);

IOCLR0 |= 1 << 20;

seg <<= 1;

}

}

IOSET0 |= 1 << 30;

delay\_ms(1);

IOCLR0 |= 1 << 30;

}

void SystemInit(void) {

PLL0CON = 0x01; PLL0CFG = 0x24;

PLL0FEED = 0xAA; PLL0FEED = 0x55;

while (!(PLL0STAT & PLOCK));

PLL0CON = 0x03;

PLL0FEED = 0xAA; PLL0FEED = 0x55;

VPBDIV = 0x01;

}

void delay\_ms(unsigned int j) {

for (unsigned int i = 0; i < j; i++)

for (volatile unsigned int x = 0; x < 10000; x++);

}  
  
  
Stepper Motor

#include <lpc214x.h>

#define LED\_ON (IO0CLR = 1U << 31)

#define LED\_OFF (IO0SET = 1U << 31)

#define PLOCK 0x00000400

void delay\_ms(unsigned int j);

void SystemInit(void);

int main() {

unsigned int i;

unsigned int steps\_clk = 100, steps\_aclk = 100;

unsigned int seq[4] = { 0x00010000, 0x00020000, 0x00040000, 0x00080000 };

IO0DIR |= (1U << 31) | 0x00FF0000; // P0.16–P0.23 as outputs

LED\_ON; delay\_ms(500); LED\_OFF; // Flash LED

SystemInit();

// Clockwise rotation

while (steps\_clk--) {

for (i = 0; i < 4 && steps\_clk; i++, steps\_clk--) {

IO0CLR = 0x000F0000;

IO0SET = seq[i];

delay\_ms(10);

}

}

// Anti-clockwise rotation

while (steps\_aclk--) {

for (i = 4; i-- > 0 && steps\_aclk; steps\_aclk--) {

IO0CLR = 0x000F0000;

IO0SET = seq[i];

delay\_ms(10);

}

}

IO0CLR = 0x00FF0000; // Stop motor

while (1);

}

void delay\_ms(unsigned int j) {

for (unsigned int i = 0; i < j; i++)

for (volatile unsigned int x = 0; x < 10000; x++);

}

void SystemInit(void) {

PLL0CON = 0x01; PLL0CFG = 0x24;

PLL0FEED = 0xAA; PLL0FEED = 0x55;

while (!(PLL0STAT & PLOCK));

PLL0CON = 0x03;

PLL0FEED = 0xAA; PLL0FEED = 0x55;

VPBDIV = 0x01; // Set PCLK = CCLK

}

Matrix

#include <lpc214x.h>

#define LED\_ON (IO0CLR = 1U << 31)

#define LED\_OFF (IO0SET = 1U << 31)

#define PLOCK 0x00000400

#define ROW\_MASK 0x000F0000 // P0.16 to P0.19

#define COL\_MASK 0x000F0000 // P1.16 to P1.19

void SystemInit(void);

void delay\_ms(unsigned int);

void uart\_init(void);

unsigned char lookup\_table[4][4] = {

{'0','1','2','3'},

{'4','5','6','7'},

{'8','9','a','b'},

{'c','d','e','f'}

};

int main() {

unsigned char row, col;

SystemInit(); uart\_init();

IO0DIR |= (1U << 31) | ROW\_MASK; // LED + Rows as output

LED\_ON; delay\_ms(500); LED\_OFF; delay\_ms(500);

while (1) {

for (row = 0; row < 4; row++) {

IO0SET = ROW\_MASK;

IO0CLR = 1U << (16 + row); // Make one row LOW

for (col = 0; col < 4; col++) {

if (!(IO1PIN & (1U << (19 - col)))) { // Active LOW

delay\_ms(50); // Debounce

while (!(IO1PIN & (1U << (19 - col)))); // Wait for release

delay\_ms(50);

IO0SET = ROW\_MASK; // Reset all rows

U0THR = lookup\_table[row][col];

break;

}

}

if (col < 4) break; // Exit outer loop if key detected

}

}

}

void uart\_init(void) {

PINSEL0 |= 0x00000005; // P0.0 = TXD0, P0.1 = RXD0

U0LCR = 0x83; // 8-bit, no parity, 1 stop bit, DLAB=1

U0DLM = 0; U0DLL = 8; // 115200 baud rate (with 60MHz PCLK)

U0LCR = 0x03; // DLAB = 0

U0FCR = 0x07; // Enable and reset TX, RX FIFO

}

void SystemInit(void) {

PLL0CON = 0x01; PLL0CFG = 0x24;

PLL0FEED = 0xAA; PLL0FEED = 0x55;

while (!(PLL0STAT & PLOCK));

PLL0CON = 0x03;

PLL0FEED = 0xAA; PLL0FEED = 0x55;

VPBDIV = 0x01;

}

void delay\_ms(unsigned int j) {

for (unsigned int i = 0; i < j; i++)

for (volatile unsigned int x = 0; x < 10000; x++);

}

DC motor

#include <lpc214x.h>

#define LED\_ON (IO0CLR = 1U << 31)

#define LED\_OFF (IO0SET = 1U << 31)

#define PLOCK 0x00000400

void delay\_ms(unsigned int);

void SystemInit(void);

void runDCMotor(int dir, int duty);

unsigned int adc(int no, int ch);

int main() {

IO0DIR |= (1U << 31) | (1U << 30) | 0x00FF0000; // LED and outputs

LED\_ON; delay\_ms(500); LED\_OFF;

SystemInit();

while (1) {

int duty = adc(1, 2) / 10;

if (duty > 100) duty = 100;

runDCMotor(2, duty); // Direction 2, duty from pot

}

}

void runDCMotor(int dir, int duty) {

IO0DIR |= 1U << 28; // P0.28 = direction control

PINSEL0 |= 2 << 18; // P0.9 as PWM6

if (dir == 1) IO0SET = 1 << 28; else IO0CLR = 1 << 28;

PWMPCR = 1 << 14; // Enable PWM6

PWMMR0 = 1000;

PWMMR6 = (1000U \* duty) / 100;

PWMTCR = 0x09; // Enable PWM and counter

PWMLER = 0x70; // Load match values

}

unsigned int adc(int no, int ch) {

unsigned int val = 0;

PINSEL0 |= 0x0F300000; // Enable ADC pins P0.10, P0.12, P0.13

if (no == 0) {

AD0CR = 0x00200600 | (1 << ch);

AD0CR |= 1 << 24;

while (!(AD0GDR & (1U << 31)));

val = AD0GDR;

} else {

AD1CR = 0x00200600 | (1 << ch);

AD1CR |= 1 << 24;

while (!(AD1GDR & (1U << 31)));

val = AD1GDR;

}

return (val >> 6) & 0x03FF;

}

void SystemInit(void) {

PLL0CON = 0x01; PLL0CFG = 0x24;

PLL0FEED = 0xAA; PLL0FEED = 0x55;

while (!(PLL0STAT & PLOCK));

PLL0CON = 0x03;

PLL0FEED = 0xAA; PLL0FEED = 0x55;

VPBDIV = 0x01;

}

void delay\_ms(unsigned int j) {

for (unsigned int i = 0; i < j; i++)

for (volatile unsigned int x = 0; x < 10000; x++);

}

LCD

#include <lpc214x.h>

// LCD Control Pins

#define RS\_ON (IO0SET = 1U << 20)

#define RS\_OFF (IO0CLR = 1U << 20)

#define EN\_ON (IO1SET = 1U << 25)

#define EN\_OFF (IO1CLR = 1U << 25)

// LED Control

#define LED\_ON (IO0CLR = 1U << 31)

#define LED\_OFF (IO0SET = 1U << 31)

#define PLOCK 0x00000400

// Function Prototypes

void SystemInit(void);

void delay\_ms(unsigned int);

void delay\_us(unsigned int);

void LCD\_Init(void);

void LCD\_CmdWrite(unsigned char);

void LCD\_DataWrite(unsigned char);

void LCD\_DisplayString(const char \*);

void LCD\_SendHigherNibble(unsigned char);

void LCD\_SendCmdSignals(void);

void LCD\_SendDataSignals(void);

void LCD\_Reset(void);

int main() {

SystemInit();

IO0DIR |= (1U << 31) | 0x00FF0000; // P0.16–P0.23 as output (LCD, LED)

IO1DIR |= (1U << 25); // P1.25 as EN output

LED\_ON; delay\_ms(500); LED\_OFF; delay\_ms(500);

LCD\_Init();

LCD\_CmdWrite(0x80); LCD\_DisplayString("RV College Of Engrng");

LCD\_CmdWrite(0xC0); LCD\_DisplayString(" Computer Science");

LCD\_CmdWrite(0x94); LCD\_DisplayString(" 4th Semester");

LCD\_CmdWrite(0xD4); LCD\_DisplayString(" B Section");

while (1);

}

// LCD Functions

void LCD\_Init(void) {

delay\_ms(100);

LCD\_Reset();

LCD\_CmdWrite(0x28); // 4-bit, 2-line, 5x7 dots

LCD\_CmdWrite(0x0E); // Display ON, Cursor ON

LCD\_CmdWrite(0x01); // Clear display

LCD\_CmdWrite(0x80); // Start at first line

}

void LCD\_Reset(void) {

for (int i = 0; i < 3; i++) {

LCD\_SendHigherNibble(0x30);

LCD\_SendCmdSignals();

delay\_ms(i == 0 ? 100 : 1);

}

LCD\_SendHigherNibble(0x20);

LCD\_SendCmdSignals();

delay\_us(200);

}

void LCD\_CmdWrite(unsigned char cmd) {

LCD\_SendHigherNibble(cmd); LCD\_SendCmdSignals();

LCD\_SendHigherNibble(cmd << 4); LCD\_SendCmdSignals();

}

void LCD\_DataWrite(unsigned char data) {

LCD\_SendHigherNibble(data); LCD\_SendDataSignals();

LCD\_SendHigherNibble(data << 4); LCD\_SendDataSignals();

}

void LCD\_DisplayString(const char \*str) {

while (\*str) LCD\_DataWrite(\*str++);

}

void LCD\_SendHigherNibble(unsigned char data) {

IO0CLR = 0x000F0000;

IO0SET = ((data >> 4) & 0x0F) << 16;

}

void LCD\_SendCmdSignals(void) {

RS\_OFF; EN\_ON; delay\_us(100); EN\_OFF;

}

void LCD\_SendDataSignals(void) {

RS\_ON; EN\_ON; delay\_us(100); EN\_OFF;

}

// Delay and System Init

void delay\_ms(unsigned int ms) {

for (unsigned int i = 0; i < ms; i++)

for (volatile unsigned int j = 0; j < 10000; j++);

}

void delay\_us(unsigned int us) {

for (unsigned int i = 0; i < us; i++)

for (volatile unsigned int j = 0; j < 10; j++);

}

void SystemInit(void) {

PLL0CON = 0x01; PLL0CFG = 0x24;

PLL0FEED = 0xAA; PLL0FEED = 0x55;

while (!(PLL0STAT & PLOCK));

PLL0CON = 0x03;

PLL0FEED = 0xAA; PLL0FEED = 0x55;

VPBDIV = 0x01;

}

Seven Segment  
#include <lpc214x.h>

#define LED\_ON (IO0CLR = 1U << 31)

#define LED\_OFF (IO0SET = 1U << 31)

#define PLOCK 0x00000400

void delay\_ms(unsigned int j);

void SystemInit(void);

unsigned char getAlphaCode(unsigned char ch);

void alphadisp7SEG(char \*buf);

int main() {

IO0DIR |= (1U << 31) | (1U << 19) | (1U << 20) | (1U << 30); // P0.19, 20, 30, 31 as output

LED\_ON;

SystemInit();

while (1) {

alphadisp7SEG("send ");

delay\_ms(500);

alphadisp7SEG("nudes");

delay\_ms(500);

}

}

unsigned char getAlphaCode(unsigned char ch) {

switch (ch) {

case 's': return 0x92;

case 'n': return 0xAB;

case 'd': return 0xA1;

case 'u': return 0xE1;

case 'e': return 0x86;

case ' ': return 0xFF;

default : return 0xFF;

}

}

void alphadisp7SEG(char \*buf) {

for (int i = 0; i < 5; i++) {

unsigned char seg = getAlphaCode(buf[i]);

for (int j = 0; j < 8; j++) {

(seg & 0x80) ? (IOSET0 |= 1 << 19) : (IOCLR0 |= 1 << 19);

IOSET0 |= 1 << 20;

delay\_ms(1);

IOCLR0 |= 1 << 20;

seg <<= 1;

}

}

IOSET0 |= 1 << 30;

delay\_ms(1);

IOCLR0 |= 1 << 30;

}

void SystemInit(void) {

PLL0CON = 0x01; PLL0CFG = 0x24;

PLL0FEED = 0xAA; PLL0FEED = 0x55;

while (!(PLL0STAT & PLOCK));

PLL0CON = 0x03;

PLL0FEED = 0xAA; PLL0FEED = 0x55;

VPBDIV = 0x01;

}

void delay\_ms(unsigned int j) {

for (unsigned int i = 0; i < j; i++)

for (volatile unsigned int x = 0; x < 10000; x++);

}