1. **Interfacing ADC**

#include <lpc214x.h>

#include <stdio.h>

#include <string.h>

#include "lcd.h"

int main(void)

{

unsigned int result;

char voltage[16];

LCD\_INIT();

PINSEL1 = 0x01000000;

AD0CR = 0x001200402;

while(1)

{

AD0CR = 0x001200402;

while ( !(AD0DR1 & 0x80000000));

result = AD0DR1;

result = (result >> 6);

result = (result & 0x000003FF);

snprintf(voltage, 16, "V=%f", ((result/1023.0) \* 3.3 ));

LCD\_CHAR(0x01, 0);

LCD\_CHAR(0x80, 0);

delay\_ms(10);

LCD\_STRING(voltage);

delay\_ms(90);

}

return 0;

}

1. **DAC**

#include<lpc214x.h>

#include<stdio.h>

#include<math.h>

void delay(int i)

{

while(i)

{

i--;

}

}

void sawtooth(int del)

{

int i;

for(i = 0; i<1024; i++)

{

DACR = i<<6;

delay(del);

}

}

void square(int del)

{

DACR = 0<<6;

delay(del\*1024);

DACR = 1023<<6;

delay(del\*1024);

}

void triangle(int del)

{

int i;

for(i = 0; i<1024; i++)

{

DACR = i<<6;

delay(del);

}

for(i = 1023; i>=0; i--)

{

DACR = i<<6;

delay(del);

}

}

void dc(float val)

{

int result;

result = floor(val/3.3\*1024);

if(result>1024)

{

DACR = 1023<<6;

}

else if(result<0)

{

DACR = 0<<6;

}

else{

DACR = result<<6;

}

}

void sinewave(int del)

{

int i, result;

float s[32] = {0.5,0.597,0.691,0.778,0.853,0.916,0.962,0.99,1,0.99,0.962,0.916,0.854,0.778,0.692,0.598,0.501,0.403,0.309,0.223,0.147,0.085,0.038,0.01,0,0.009,0.038,0.084,0.145,0.221,0.307,0.401};

for(i=0; i<32; i++)

{

result = 1023\*s[i];

DACR = result<<6;

delay(del\*32);

}

}

int main(void)

{

PINSEL1 = 0x00080000;

while(1)

{

}

return 0;

}

1. **LCD and PWM**

#include<lpc214x.h>

#include<stdint.h>

void PWM5\_INIT(uint32\_t cycle\_rate)

{

PINSEL1 = 0x00000400;

PWMMR0 = cycle\_rate;

PWMMCR = 0x00000003;

PWMPCR = 0x00002000;

PWMTCR = 0x00000009;

}

int main()

{

uint32\_t i = 0x00000000;

PWM5\_INIT(0x000004FF);

PWMPR = 60-1;

PWMMR5 = 0x00000000;

PWMLER = 0x00000020;

while(1)

{

while (i != (PWMMR0/3))

{

while(PWMIR != 0x00000001);

PWMIR = 0x00000001;

PWMMR5 = i;

PWMLER = 0x00000020;

i++;

}

while (i != 1)

{

while(PWMIR != 0x00000001);

PWMIR = 0x00000001;

PWMMR5 = i;

PWMLER = 0x00000020;

i--;

}

}

return 0;

}

1. **Real Time Clock (RTC) with Serial Port**

#include<lpc214x.h>

#include<stdio.h>

#include<string.h>

#include "lcd.h"

#include "i2c.h"void LCD\_PRINT\_LINE(char\* msg)

{

LCD\_CHAR(0x01, 0);

LCD\_CHAR(0x80, 0);

LCD\_STRING(msg);

delay\_ms(1000);

LCD\_CHAR(0x01, 0);

LCD\_CHAR(0x80, 0);

}

char bcdToAsciiLow(unsigned char bcd)

{

char result;

result = (char) ((bcd & 0x0f) + 48);

return result;

}

char bcdToAsciiUp(unsigned char bcd)

{

char result;

result = (char) (((bcd & 0xf0) >> 4) + 48);

return result;

}

void setTime(void){

I2C\_WRITE(0xD0, 0x00, 0x00);

I2C\_WRITE(0xD0, 0x01, 0x31);

I2C\_WRITE(0xD0, 0x02, 0x18);

I2C\_WRITE(0xD0, 0x03, 0x01);

I2C\_WRITE(0xD0, 0x04, 0x04);

I2C\_WRITE(0xD0, 0x05, 0x09);

I2C\_WRITE(0xD0, 0x06, 0x20)

}

int main(void)

{

char sec[2], min[2], hrs[2];

char time[16];

LCD\_INIT();

I2C\_INIT();

while(1)

{

sec[0] = I2C\_READ(0xD0, 0x00);

min[0] = I2C\_READ(0xD0, 0x01);

hrs[0] = I2C\_READ(0xD0, 0x02);

sec[1] = bcdToAsciiUp(sec[0]);

sec[0] = bcdToAsciiLow(sec[0]);

min[1] = bcdToAsciiUp(min[0]);

min[0] = bcdToAsciiLow(min[0]);

hrs[1] = bcdToAsciiUp(hrs[0]);

hrs[0] = bcdToAsciiLow(hrs[0]);

snprintf(time, 16, "%c%c:%c%c:%c%c",hrs[1],hrs[0],min[1],min[0],sec[1],sec[0]);

LCD\_PRINT\_LINE(time);

}

return 0;

}

1. **Keyboard with LCD**

#include <lpc214x.h>

#include <stdint.h>

#include <stdio.h>

#include "lcd.h"

char keys[4][4] = {{'1', '2', '3', '4'},

{'5', '6', '7', '8'},

{'9', '0', 'a', 'b'},

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

uint32\_t rows[4] = {0x00fe0000, 0x00fd0000, 0x00fb0000, 0x00f70000};

uint32\_t cols[4] = {0x00100000, 0x00200000, 0x00400000, 0x00800000};

char KEYPAD\_CHECK(void){

int i,j;

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

IO1SET = rows[i];

delay\_ms(5);

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

if((IO1PIN&cols[j]) == 0){

while((IO1PIN&cols[j]) == 0);

return(keys[i][j]);

}

}

IO1CLR = rows[i];

delay\_ms(5);

}

return 'x';

}

int main(void)

{

char c\_prt;

LCD\_INIT();

LCD\_CHAR(0xC0, 0);

IO1DIR = 0x000F0000;

while(1){

delay\_ms(5);

c\_prt = KEYPAD\_CHECK();

if(c\_prt != 'x'){

LCD\_CHAR(0xC0, 0);

LCD\_CHAR(c\_prt, 1);

}

}

return 0;

}

1. **EPROM**

#include<lpc214x.h>

#include<stdio.h>

#include<string.h>

#include "lcd.h"

#include "i2c.h"

void LCD\_PRINT\_LINE(char\* msg)

{

LCD\_CHAR(0x01, 0);

LCD\_CHAR(0x80, 0);

LCD\_STRING(msg);

delay\_ms(1000);

LCD\_CHAR(0x01, 0);

LCD\_CHAR(0x80, 0);

}

void EEPROM\_I2C\_WRITE(char dev\_addr, char data, unsigned short addr)

{

char eff\_dev\_addr;

eff\_dev\_addr = ((dev\_addr & 0xf8) | ((addr & 0x0300) >> 7));

I2C\_WRITE(eff\_dev\_addr, data, addr);

}

char EEPROM\_I2C\_READ(char dev\_addr, unsigned short addr)

{

char eff\_dev\_addr, result;

eff\_dev\_addr = ((dev\_addr & 0xf8) | ((addr & 0x0300) >> 7));

result = I2C\_READ(eff\_dev\_addr, addr);

return result;

}

int main(void)

{

char store;

char lcd\_tx[16];

I2C\_INIT();

LCD\_INIT();

EEPROM\_I2C\_WRITE(0xA0, 0x41, 0x000F);

snprintf(lcd\_tx, 16,"%c -> 0x00F",0x41);

LCD\_PRINT\_LINE(lcd\_tx);

EEPROM\_I2C\_WRITE(0xA0, 0x5A, 0x001F);

snprintf(lcd\_tx, 16,"%c -> 0x01F",0x5A);

LCD\_PRINT\_LINE(lcd\_tx);

store = EEPROM\_I2C\_READ(0xA0, 0x000F);

snprintf(lcd\_tx, 16,"%c <- 0x00F", store);

LCD\_PRINT\_LINE(lcd\_tx);

store = EEPROM\_I2C\_READ(0xA0, 0x001F);

snprintf (lcd\_tx, 16,"%c <- 0x01F", store);

LCD\_PRINT\_LINE(lcd\_tx);

return 0;

}

1. **Flashing of LED’s**

#include<lpc214x.h>

#include<stdio.h>

void delay\_ms (int ms)

{

int i, j;

for(j=0;j<ms;j++)

{

for(i=0;i<12000;i++);

}

}

int main()

{

IO1DIR=0x00FF0000;// GPIO pins P1.16-P1.23 configured as O/P

while(1)

{

IO1SET=0x00550000;

delay\_ms(500);

IO1CLR=0X00550000;

IO1SET=0X00AA0000;

delay\_ms(500);

IO1CLR=0x00AA0000;

}

return 0;

}

1. **Stepper Motor**

#include<lpc214x.h>

#include<stdint.h>

void delay\_ms(int x)

{

int i,j;

for(i=0;i<x;i++)

for(j=0;j<120000;j++);

}

void half\_step()

{

int i,j;

uint32\_t val[8] = {0x008C0000, 0x00CC0000, 0x004C0000, 0x006C0000, 0x002C0000, 0x003C0000, 0x001C0000, 0x009C0000};

for(i=0;i<12;i++)

{

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

{

IO0SET = val[j];

delay\_ms(10);

IO0CLR = val[j];

}

}

for(i=0;i<12;i++)

{

for(j=7;j>=0;j--)

{

IO0SET = val[j];

delay\_ms(10);

IO0CLR = val[j];

}

}

}

void full\_step(void)

{

int i,j;

uint32\_t val[4] = {0x008C0000, 0x004C0000, 0x002C0000, 0x001C0000};

for(i=0;i<12;i++)

{

for(j=0;j<4;j++)

{

IO0SET = val[j];

delay\_ms(10);

IO0CLR = val[j];

}

}

for(i=0;i<12;i++)

{

for(j=3;j>=0;j--)

{

IO0SET = val[j];

delay\_ms(10);

IO0CLR = val[j];

}

}

}

int main(){

IO0DIR = 0x00FC0000;

while(1)

{

full\_step();

}

return 0;

}