Week 2 Lab Work

Q1. Write a C code to display name in LCD.

# include <LPC17xx.h>

#define RS (1 << 10)

#define EN (1 << 11)

#define DATA\_PINS (0xFF << 15)

void delay(uint32\_t ms){

uint32\_t i, j;

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

for(j = 0; j < 1250; j++){}

}

}

void cmd\_reg(char cmd){

//SETTING DATA PINS AND CMD TO WRITE

LPC\_GPIO0 -> FIOCLR = DATA\_PINS;

LPC\_GPIO0 -> FIOSET = cmd << 15;

//MAKING RS 0 AND EN 1

LPC\_GPIO0 -> FIOCLR = RS;

LPC\_GPIO0 -> FIOSET = EN;

delay(100);

//RESETTING EN TO 0

LPC\_GPIO0 -> FIOCLR = EN;

}

void data\_reg(char data){

//SETTING DATA PINS AND CMD TO WRITE

LPC\_GPIO0 -> FIOCLR = DATA\_PINS;

LPC\_GPIO0 -> FIOSET = data << 15;

//MAKING RS 1 AND EN 1

LPC\_GPIO0 -> FIOSET = RS;

LPC\_GPIO0 -> FIOSET = EN;

delay(100);

//RESETTING EN TO 0

LPC\_GPIO0 -> FIOCLR = EN;

}

void string\_write(char \*str){

while(\*str != '\0'){

data\_reg(\*str);

str++;

}

}

int main(){

LPC\_GPIO0 -> FIODIR |= RS;

LPC\_GPIO0 -> FIODIR |= EN;

LPC\_GPIO0 -> FIODIR |= DATA\_PINS;

cmd\_reg(0x38); //8-bit mode write in 2 lines.

cmd\_reg(0x0E); //display on, cursor on.

cmd\_reg(0x01); //clear the screen

data\_reg(0x53);

data\_reg('O');

data\_reg('M');

data\_reg('Y');

data\_reg(0x41);

data\_reg(' ');

data\_reg(0x41);

data\_reg('N');

data\_reg(0x41);

data\_reg('N');

data\_reg('D');

delay(2000);

cmd\_reg(0xC0); //move cursor to second line

string\_write("SUNNY KR");

delay(3000);

cmd\_reg(0x01); //clear the screen

}

Q2. Write a C Code to display year and temp on LCD.

# include <LPC17xx.h>

# include <stdio.h>

#define RS (1 << 10)

#define EN (1 << 11)

#define DATA\_PINS (0xFF << 15)

void delay(uint32\_t ms){

uint32\_t i, j;

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

for(j = 0; j < 1250; j++){}

}

}

void cmd\_reg(char cmd){

//SETTING DATA PINS AND CMD TO WRITE

LPC\_GPIO0 -> FIOCLR = DATA\_PINS;

LPC\_GPIO0 -> FIOSET = cmd << 15;

//MAKING RS 0 AND EN 1

LPC\_GPIO0 -> FIOCLR = RS;

LPC\_GPIO0 -> FIOSET = EN;

delay(100);

//RESETTING EN TO 0

LPC\_GPIO0 -> FIOCLR = EN;

}

void data\_reg(char data){

//SETTING DATA PINS AND DATA TO WRITE

LPC\_GPIO0 -> FIOCLR = DATA\_PINS;

LPC\_GPIO0 -> FIOSET = data << 15;

//MAKING RS 1 AND EN 1

LPC\_GPIO0 -> FIOSET = RS;

LPC\_GPIO0 -> FIOSET = EN;

delay(100);

//RESETTING EN TO 0

LPC\_GPIO0 -> FIOCLR = EN;

}

void string\_write(char \*str){

while(\*str != '\0'){

data\_reg(\*str);

str++;

}

}

int main(){

char sval[20];

int year = 2024;

float temp = 32.4;

LPC\_GPIO0 -> FIODIR |= RS;

LPC\_GPIO0 -> FIODIR |= EN;

LPC\_GPIO0 -> FIODIR |= DATA\_PINS;

cmd\_reg(0x38); //8-bit mode write in 2 lines.

cmd\_reg(0x0E); //display on, cursor on.

cmd\_reg(0x01); //clear the screen

sprintf(sval, "%d", year);

string\_write(sval);

cmd\_reg(0xC0);

sprintf(sval, "Temp: %.1f C", temp);

string\_write(sval);

cmd\_reg(0xC0);

cmd\_reg(0x01);

}

Q3. Write a C Code to display name in LCD with header and functions in a separate file.

**lcd.h Code:**

# include <LPC17xx.h>

# include <stdio.h>

#define RS (1 << 10)

#define EN (1 << 11)

#define DATA\_PINS (0xFF << 15)

void delay(uint32\_t ms);

void cmd\_reg(char cmd);

void data\_reg(char data);

void string\_write(char \*str);

**lcd\_def.c Code:**

# include "lcd.h"

void delay(uint32\_t ms){

uint32\_t i, j;

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

for(j = 0; j < 1250; j++){}

}

}

void cmd\_reg(char cmd){

//SETTING DATA PINS AND CMD TO WRITE

LPC\_GPIO0 -> FIOCLR = DATA\_PINS;

LPC\_GPIO0 -> FIOSET = cmd << 15;

//MAKING RS 0 AND EN 1

LPC\_GPIO0 -> FIOCLR = RS;

LPC\_GPIO0 -> FIOSET = EN;

delay(100);

//RESETTING EN TO 0

LPC\_GPIO0 -> FIOCLR = EN;

}

void data\_reg(char data){

//SETTING DATA PINS AND CMD TO WRITE

LPC\_GPIO0 -> FIOCLR = DATA\_PINS;

LPC\_GPIO0 -> FIOSET = data << 15;

//MAKING RS 1 AND EN 1

LPC\_GPIO0 -> FIOSET = RS;

LPC\_GPIO0 -> FIOSET = EN;

delay(100);

//RESETTING EN TO 0

LPC\_GPIO0 -> FIOCLR = EN;

}

void string\_write(char \*str){

while(\*str != '\0'){

data\_reg(\*str);

str++;

}

}

**lcd\_app.c Code:**

# include "lcd.h"

int main(){

LPC\_GPIO0 -> FIODIR |= RS;

LPC\_GPIO0 -> FIODIR |= EN;

LPC\_GPIO0 -> FIODIR |= DATA\_PINS;

cmd\_reg(0x38); //8-bit mode write in 2 lines.

cmd\_reg(0x0E); //display on, cursor on.

cmd\_reg(0x01); //clear the screen

/\*

data\_reg(0x53);

data\_reg('O');

data\_reg('M');

data\_reg('Y');

data\_reg(0x41);

data\_reg(' ');

data\_reg(0x41);

data\_reg('N');

data\_reg(0x41);

data\_reg('N');

data\_reg('D');

\*/

string\_write("SOMYA ANAND");

delay(2000);

cmd\_reg(0xC0); //move cursor to second line

string\_write("SUNNY KR");

delay(2000);

cmd\_reg(0x01); //clear the screen

}

Q4. Write a C Code to display counting number till 100 in LCD.

**count.h**

# include <LPC17xx.h>

# include <stdio.h>

#define RS (1 << 10)

#define EN (1 << 11)

#define DATA\_PINS (0xFF << 15)

void delay(uint32\_t ms);

void cmd\_reg(char cmd);

void data\_reg(char data);

void string\_write(char \*str);

void lcd\_init(void);

**count\_def.c**

# include "count.h"

void delay(uint32\_t ms){

uint32\_t i, j;

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

for(j = 0; j < 1250; j++){}

}

}

void cmd\_reg(char cmd){

//SETTING DATA PINS AND CMD TO WRITE

LPC\_GPIO0 -> FIOCLR = DATA\_PINS;

LPC\_GPIO0 -> FIOSET = cmd << 15;

//MAKING RS 0 AND EN 1

LPC\_GPIO0 -> FIOCLR = RS;

LPC\_GPIO0 -> FIOSET = EN;

delay(100);

//RESETTING EN TO 0

LPC\_GPIO0 -> FIOCLR = EN;

}

void data\_reg(char data){

//SETTING DATA PINS AND CMD TO WRITE

LPC\_GPIO0 -> FIOCLR = DATA\_PINS;

LPC\_GPIO0 -> FIOSET = data << 15;

//MAKING RS 1 AND EN 1

LPC\_GPIO0 -> FIOSET = RS;

LPC\_GPIO0 -> FIOSET = EN;

delay(100);

//RESETTING EN TO 0

LPC\_GPIO0 -> FIOCLR = EN;

}

void string\_write(char \*str){

while(\*str != '\0'){

data\_reg(\*str);

str++;

}

}

void lcd\_init(void){

LPC\_GPIO0 -> FIODIR |= RS;

LPC\_GPIO0 -> FIODIR |= EN;

LPC\_GPIO0 -> FIODIR |= DATA\_PINS;

cmd\_reg(0x38); //8-bit mode write in 2 lines.

cmd\_reg(0x0E); //display on, cursor on.

cmd\_reg(0x01); //clear the screen

}

**count\_app.c**

# include "count.h"

int main(){

int count = 0;

char str\_count[10];

lcd\_init();

while(1){

sprintf(str\_count, "%02d", count);

string\_write(str\_count);

delay(100);

count++;

if(count == 100){

count = 0;

}

cmd\_reg(0x87);

}

}

Q5. Write a C Code to display time in hrs, mins and secs in LCD.

# include "count.h"

int main(){

int hr = 12, min = 59, sec = 58;

char str\_time[10];

lcd\_init();

while(1){

cmd\_reg(0x84);

sprintf(str\_time, "%02d:%02d:%02d", hr, min, sec);

string\_write(str\_time);

delay(1000);

sec = sec + 1;

if(sec == 60){

sec = 0;

min = min + 1;

}

if(min == 60){

min = 0;

hr = hr + 1;

}

if(hr == 24){

hr = 0;

}

}

}

Q6. Write a C Code to display the pressed key on LCD.

# include <LPC17xx.h>

# include "lcd.h"

#define ROW\_PINS (0x0F << 4)

#define COL\_PINS (0x0F)

int main(){

uint8\_t code[4] = {0x0E, 0x0D, 0x0B, 0x07};

char key[4][4] = {

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

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

{'8', '9', 'A', 'B'},

{'C', 'D', 'E', 'F'}

};

uint8\_t i, j, col\_val;

LPC\_GPIO2 -> FIODIR |= ROW\_PINS; //MAKE ROW\_PINS AS OUTPUT PINS

LPC\_GPIO2 -> FIODIR &= ~COL\_PINS; //MAKE COL\_PINS AS INPUT PINS

lcd\_init();

cmd\_reg(0x0C);

string\_write("Key Pressed: ");

while(1){

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

LPC\_GPIO2 -> FIOCLR = ROW\_PINS; //MAKE ROW\_PINS AS 0 for output

LPC\_GPIO2 -> FIOSET = code[i] << 4; //MAKE COL\_PINS AS INPUT PINS

col\_val = LPC\_GPIO2 -> FIOPIN & 0x0F; //GET ALL 1s OF KEYPAD

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

if(col\_val == code[j]){

break;

}

}

if(col\_val != 0x0F){

data\_reg(key[i][j]);

cmd\_reg(0x10);

}

}

}

}

Q7. Write a C Code to create password and check whether entered password is correct or not.

# include <LPC17xx.h>

# include <string.h>

#include "lcd.h"

#define ROW\_PINS (0x0F << 4) // P2.4 to P2.7 as rows

#define COL\_PINS (0x0F) // P2.0 to P2.3 as columns

int main() {

uint8\_t row\_codes[4] = {0x0E, 0x0D, 0x0B, 0x07}; // Row activation codes

char keys[4][4] = {

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

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

{'8', '9', 'A', 'B'},

{'C', 'D', 'E', 'F'}

};

char entered\_password[5] = ""; // Array to store entered password

char correct\_password[] = "1710"; // Predefined correct password

uint8\_t password\_index = 0; // Index to track password input

uint8\_t i, j, col\_val;

// Configure GPIO pins

LPC\_GPIO2->FIODIR |= ROW\_PINS; // Set row pins as output

LPC\_GPIO2->FIODIR &= ~COL\_PINS; // Set column pins as input

// Initialize LCD

lcd\_init();

cmd\_reg(0x0C); // Display ON, cursor OFF

string\_write("Enter Password:");

while (1) {

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

LPC\_GPIO2->FIOCLR = ROW\_PINS; // Clear all row pins

LPC\_GPIO2->FIOSET = row\_codes[i] << 4; // Set the current row active

col\_val = LPC\_GPIO2->FIOPIN & COL\_PINS; // Read column pins

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

if (col\_val == row\_codes[j]) { // Match column value

break;

}

}

if (col\_val != 0x0F) { // A key is pressed

if (password\_index < 4) { // Limit password to 4 characters

entered\_password[password\_index++] = keys[i][j];

data\_reg(' '); // Display '' for entered characters

}

if (password\_index == 4) { // When 4 characters are entered

entered\_password[4] = '\0'; // Null-terminate the password string

cmd\_reg(0x01); // Clear the screen

if (strcmp(entered\_password, correct\_password) == 0) {

string\_write("Correct Password");

} else {

string\_write("Wrong Password");

}

delay(3000); // Delay to show result

cmd\_reg(0x01); // Clear the screen for next input

string\_write("Enter Password:");

password\_index = 0; // Reset index for new input

}

while ((LPC\_GPIO2->FIOPIN & COL\_PINS) != 0x0F); // Wait for key release

}

}

}

return 0; // Return from main

}

Q8. Write a C Code to display Temperature on LCD using ADC.

# include <LPC17xx.h>

# include "lcd.h"

#define CH\_SEL (1 << 2)

#define CLK\_DIV (0xFF << 8)

#define BURST\_MODE (0x01 << 16)

#define POWER\_UP (0x01 << 21)

#define START\_CONV (0x01 << 24)

#define DONE\_CONV (0x01U << 31)

#define TEMP\_COEF 100.0f

int main(){

uint32\_t adc\_val;

float volt, temp;

char sval[20];

LPC\_SC -> PCONP |= (1 << 12); //ENABLE ADC POWER

LPC\_PINCON -> PINSEL1 |= (1 << 18); //P0.25 IS CONFIGURED AS ADC CHANNEL 2 INPUT

LPC\_PINCON -> PINSEL1 &= ~(1 << 19);

LPC\_ADC -> ADCR |= CH\_SEL; //SELECT ADC CHANNEL 2

LPC\_ADC -> ADCR &= ~CLK\_DIV; //ADC CLK = PCLK / (0+1) = 1MHz

LPC\_ADC -> ADCR &= ~BURST\_MODE; //SELECT ADC SOFTWARE MODE

LPC\_ADC -> ADCR |= POWER\_UP; //MAKE ADC OPERATIONAL

lcd\_init();

while(1){

LPC\_ADC -> ADCR |= START\_CONV; //STARTS CONVERSION

while((LPC\_ADC -> ADGDR & DONE\_CONV) == 0){ }//WAIT FOR CONVERSION TO END

adc\_val = (LPC\_ADC -> ADGDR >> 4) & 0xFFF;

volt = adc\_val \*3.3/4096;

temp = volt \* TEMP\_COEF;

sprintf(sval, "Temp: %.2f C", temp);

cmd\_reg(0x0C);

string\_write(sval);

delay(100);

cmd\_reg(0x80);

}

}

Q9. Develop a LPC1768 C program to toggle LEDs at every 2secs using timer delay.

# include <LPC17xx.h>

#define ALL\_LED (0xFF << 19)

int main(){

LPC\_GPIO1 -> FIODIR |= ALL\_LED;

LPC\_GPIO1 -> FIOCLR = ALL\_LED;

LPC\_SC -> PCONP |= (1 << 1); //ENABLE POWER TO TIMER-0

LPC\_TIM0 -> PR = 0x00;

LPC\_TIM0 -> MR0 = 2000000; //DELAY = COUNT/PCLK = (PR+1)\*(MR0)

LPC\_TIM0 -> MCR = (1 << 1) | (1 << 0); //INTERRUPT AND REPEAT

LPC\_TIM0 -> TCR |= (1 << 1); //LOAD PC = 0

LPC\_TIM0 -> TCR |= (1 << 0); //START TIMER

while(1){

LPC\_GPIO1 -> FIOSET = ALL\_LED;

while((LPC\_TIM0 -> IR & (1 << 0)) == 0){ }

LPC\_TIM0 -> IR |= (1 << 0);

LPC\_GPIO1 -> FIOCLR = ALL\_LED;

while((LPC\_TIM0 -> IR & (1 << 0)) == 0){ }

LPC\_GPIO1 -> FIOCLR = ALL\_LED;

}

}

Q10. Develop a LPC1768 C program to toggle 1st , 4th and 7th LED at different intervals using timer delay.

# include <LPC17xx.h>

#define ALL\_LED (0xFF << 19)

#define LED1 (1 << 19)

#define LED2 (1 << 23)

#define LED3 (1 << 26)

int main(){

LPC\_GPIO1 -> FIODIR |= ALL\_LED;

LPC\_GPIO1 -> FIOCLR = ALL\_LED;

LPC\_SC -> PCONP |= (1 << 1); //ENABLE POWER TO TIMER-0

LPC\_TIM0 -> PR = 0;

LPC\_TIM0 -> MR0 = 1000000; //DELAY = COUNT/PCLK = (PR+1)\*(MR0)

LPC\_TIM0 -> MR1 = 5000000;

LPC\_TIM0 -> MR2 = 8000000;

LPC\_TIM0 -> MR3 = 10000000;

LPC\_TIM0 -> MCR = (1 << 10) | (1 << 9) | (1 << 6) | (1 << 3) | (1 << 0); //INTERRUPT AND REPEAT

LPC\_TIM0 -> TCR |= (1 << 1); //LOAD PC = 0

LPC\_TIM0 -> TCR |= (1 << 0); //START TIMER

while(1){

while((LPC\_TIM0 -> IR & (1 << 0)) == 0){ }

LPC\_GPIO1 -> FIOSET = LED1;

LPC\_TIM0 -> IR |= (1 << 0);

while((LPC\_TIM0 -> IR & (1 << 1)) == 0){ }

LPC\_GPIO1 -> FIOSET = LED2;

LPC\_TIM0 -> IR |= (1 << 1);

while((LPC\_TIM0 -> IR & (1 << 2)) == 0){ }

LPC\_GPIO1 -> FIOSET = LED3;

LPC\_TIM0 -> IR |= (1 << 2);

while((LPC\_TIM0 -> IR & (1 << 3)) == 0){ }

LPC\_GPIO1 -> FIOSET = ALL\_LED;

LPC\_TIM0 -> IR |= (1 << 3);

}

}

Q11. Write a C Code to configure Real-Time Clock on LED.

# include <LPC17xx.h>

# include "lcd.h"

int main(){

char stime[16];

char sdate[16];

lcd\_init();

cmd\_reg(0x0C);

LPC\_SC -> PCONP |= (1 << 9); //ENABLE POWER to RTC

LPC\_RTC -> CCR = 0x00; //DISABLE TIME COUNTERS

LPC\_RTC -> HOUR = 11;

LPC\_RTC -> MIN = 31;

LPC\_RTC -> SEC = 0;

LPC\_RTC -> DOM = 14;

LPC\_RTC -> MONTH = 12;

LPC\_RTC -> YEAR = 2024;

LPC\_RTC -> CCR = 0x01; //ENABLE TIME COUNTERS

while(1){

sprintf(stime, "%02d:%02d:%02d", LPC\_RTC->HOUR, LPC\_RTC->MIN, LPC\_RTC->SEC);

string\_write(stime);

cmd\_reg(0xC0);

sprintf(sdate, "%02d/%02d/%04d", LPC\_RTC->DOM, LPC\_RTC->MONTH, LPC\_RTC->YEAR);

string\_write(sdate);

cmd\_reg(0x80);

}

}

Q12. Write a C Code to configure clock with Alarm.

# include <LPC17xx.h>

# include "lcd.h"

#define BUZZER (1 << 27)

int main(){

char stime[16];

char sdate[16];

LPC\_GPIO1 -> FIODIR |= BUZZER;

LPC\_GPIO1 -> FIOCLR = BUZZER;

lcd\_init();

cmd\_reg(0x0C);

LPC\_SC -> PCONP |= (1 << 9); //ENABLE POWER to RTC

LPC\_RTC -> CCR = 0x00; //DISABLE TIME COUNTERS

LPC\_RTC -> HOUR = 23;

LPC\_RTC -> MIN = 59;

LPC\_RTC -> SEC = 55;

LPC\_RTC -> DOM = 31;

LPC\_RTC -> MONTH = 12;

LPC\_RTC -> YEAR = 2024;

LPC\_RTC -> ALHOUR = 00;

LPC\_RTC -> ALMIN = 00;

LPC\_RTC -> AMR = 0xF9; //ENABLE ONLY HOUR AND MIN FOR COMPARISON

LPC\_RTC -> CCR = 0x01; //ENABLE TIME COUNTERS

while(1){

sprintf(stime, "%02d:%02d:%02d", LPC\_RTC->HOUR, LPC\_RTC->MIN, LPC\_RTC->SEC);

string\_write(stime);

cmd\_reg(0xC0);

sprintf(sdate, "%02d/%02d/%04d", LPC\_RTC->DOM, LPC\_RTC->MONTH, LPC\_RTC->YEAR);

string\_write(sdate);

cmd\_reg(0x80);

if((LPC\_RTC -> ILR & (1 << 1)) != 0){ //IF CURRENT TIME == ALARM TIME

LPC\_RTC -> ILR |= (1 << 1); //CLEAR ALARM INTERRUPT

LPC\_GPIO1 -> FIOSET = BUZZER;

}

else{

LPC\_GPIO1 -> FIOCLR = BUZZER;

}

}

}

Q13. Write a C Code to print Happy New Year.

# include <LPC17xx.h>

# include "lcd.h"

#define BUZZER (1 << 27)

int main(){

LPC\_GPIO1 -> FIODIR |= BUZZER;

LPC\_GPIO1 -> FIOCLR = BUZZER;

lcd\_init();

cmd\_reg(0x0C);

LPC\_SC -> PCONP |= (1 << 9); //ENABLE POWER to RTC

LPC\_RTC -> CCR = 0x00; //DISABLE TIME COUNTERS

LPC\_RTC -> HOUR = 23;

LPC\_RTC -> MIN = 59;

LPC\_RTC -> SEC = 55;

LPC\_RTC -> DOM = 31;

LPC\_RTC -> MONTH = 12;

LPC\_RTC -> YEAR = 2024;

LPC\_RTC -> ALHOUR = 00;

LPC\_RTC -> ALMIN = 00;

LPC\_RTC -> AMR = 0xF9; //ENABLE ONLY HOUR AND MIN FOR COMPARISON

LPC\_RTC -> CCR = 0x01; //ENABLE TIME COUNTERS

while(1){

if((LPC\_RTC -> ILR & (1 << 1)) != 0){ //IF CURRENT TIME == ALARM TIME

LPC\_RTC -> ILR |= (1 << 1); //CLEAR ALARM INTERRUPT

LPC\_GPIO1 -> FIOSET = BUZZER;

string\_write("HAPPY NEW YEAR!");

cmd\_reg(0x01);

}

else{

LPC\_GPIO1 -> FIOCLR = BUZZER;

}

}

}