EXP1a: BASIC ARITHMETIC OPERATIONS USING 8086 MICROPROCESSORS

```
ADD:                 SUB:
MOV AX, 8212H        MOV AX, 1212H
MOV BX, 9313H        MOV BX, 1313H
ADD AX, BX           SUB AX, BX
MOV [1200], AX       MOV [1200], AX
HLT                  HLT
```

```
MULTI:                 DIV:
MOV DX,0000            MOV DX,0000
MOV AX, 1212           MOV AX, 1212
MOV BX, 02             MOV BX, 02
MUL BX                 DIV BX
MOV DI, 1520           MOV DI, 1520
MOV [DI], AL           MOV [DI], AL
INC DI                 INC DI
MOV [DI], AH           MOV [DI], AH
INC DI                 INC DI
MOV [DI], DX           MOV [DI], DX
HLT                    HLT
```

**EXP1b: (ARRAY PROGRAMMING-
FINDING LARGEST AND SMALLEST NUMBER)**

```
MOV SI,2000H
MOV CL, [SI]
MOV CH,00H
INC SI
MOV AL, [SI]
DEC CL
INC SI
CMP AL,[SI]
JNC/JC , L1 (1013)
MOV AL,[SI]
INC SI
LOOP L2 (100D)
```

EXP2 (BASIC ARITHMETIC OPERATIONS USING 8051 MICROCONTROLLERS)

```
ADD:                 SUB:
MOV A, #13H          MOV A, #20H
ADD A, #14H          SUBB A, #10H
MOV DPTR, #8600      MOV DPTR, #8600
MOVX, @DPTR          MOVX, @DPTR
HERE: SJMP HERE      HERE:SJMP HERE
```

MULTI:

```
MOV A, #06H
MOV F0, #03H
MUL AB
MOV DPTR, #8600
MOVX, @DPTR, A
INC DPTR
MOV A,F0
MOVX, @DPTR, A
HERE: SJMP HERE
```

DIV:

```
MOV A, #09H
MOV F0, #03H
DIV AB
MOV DPTR, #8600
MOVX, @DPTR, A
INC DPTR
MOV A,F0
MOVX, @DPTR, A
HERE: SJMP HERE
```

EXP3: STEPPER MOTOR CONTROL USING 8086 MICROPROCESSOR

```
MOV AL,80H
MOV DX,FF36
OUT DX,AL
MOV SI,1200
MOV BL,04
MOV AL,[SI]
MOV DX,FF30
OUT DX,AL
CALL DELAY
INC SI
DEC BL
JNE REPEAT
JMP START
MOV CX,0903
DEC CX
JNE LOOP
RET
```

EXP4: Digital to Analog Converter

```
CLR P0.7
MOV P1, #50H
LCALL DELAY
MOV P1, #0BBH
LCALL DELAY
SJMP START
MOV R1, #45H
L2 DJNZ R1, L2
RET
```

Exp 4 : Echoing the Switches on the LEDs

```
MOV P1, P2
JMP START
```

EXP5: INTERFACING ANALOG TO DIGITAL CONVERTER TO 8051

```
MOV DPTR,#FFC8      MOV A, #10
MOVX @DPTR,A        MOV A,#18
MOVX @DPTR, A       MOV DPTR,#FFD0
MOV A,#01           MOVX @DPTR,A
MOV A,#00           MOV @DPTR,A
MOV DPTR,#FFD8      MOVX A,@DPTR
JNB E0,WAIT         MOV DPTR,#FFC0
MOVX A,@DPTR        MOV DPTR,#4150
MOVX @DPTR,A        SJMP HERE
```

EXP6: LED BLINK

```
void setup() {
pinMode(LED_BUILTIN, OUTPUT); }
void loop() {
digitalWrite(LED_BUILTIN, HIGH);
delay(1000);
digitalWrite(LED_BUILTIN, LOW);
delay(1000);  }
```

EXP6: LED FADE

```
const int ledPin = 9;
const int fadeRate = 5;
void setup() {
pinMode(ledPin, OUTPUT); }
void loop() {
for (int brightness = 0; brightness <= 255; brightness++) {
analogWrite(ledPin, brightness);
delay(fadeRate); }
for (int brightness = 255; brightness >= 0; brightness--) {
analogWrite(ledPin, brightness);
delay(fadeRate); } }
```

Ex No: 7 DESIGN OF A TRAFFIC LIGHT CONTROLLER USING ARDUINO

```
void setup() {
pinMode(A0,OUTPUT);   pinMode(A1,OUTPUT);
pinMode(A2,OUTPUT);   Serial.begin(9600); }
void loop() {  digitalWrite(A0,HIGH);
digitalWrite(A1,LOW);   digitalWrite(A2,LOW);
Serial.println("STOP");   delay(5000);
digitalWrite(A0,LOW);   digitalWrite(A1,HIGH);
digitalWrite(A2,LOW);    Serial.println("Ready");
delay(2000);    digitalWrite(A0,LOW);
digitalWrite(A1,LOW);    digitalWrite(A2,HIGH);
Serial.println("Go");  delay(5000);  }
```

Ex No: 8 (DESIGN A SIMPLE CHAT SERVER USING ARDUINO)

```
#include <SPI.h>
#include <Ethernet.h>
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,1, 177);
IPAddress gateway(192,168,1, 1);
IPAddress subnet(255, 255, 0, 0);
EthernetServer server(23);
boolean gotAMessage = false;
void setup() {   Ethernet.begin(mac, ip, gateway, subnet);
server.begin();  Serial.begin(9600); }
void loop() {   EthernetClient client = server.available();
if (client) {  if (!gotAMessage) {
Serial.println("We have a new client");
client.println("Hello, client!");  gotAMessage = true;  }
char thisChar = client.read();    server.write(thisChar);
Serial.print(thisChar);  }  }
```

Ex No:9 BASIC PROGRAMMING USING ARM PROCESSOR USING KEIL C

1 LED Blink #include <LPC214x.H>

```
unsigned int delay;
int main(void) { IO0DIR = (1<<20);
 while(1)  { IO0CLR = (1<<20);
for(delay=0; delay<500000; delay++);
IO0SET = (1<<20);
for(delay=0; delay<500000; delay++);  }  }
```

2 LED Switch #include <lpc214x.h>

```
int main(void) {  IO1DIR &= ~(1<<16);
IO0DIR |= (1<<16);
While (1)  {
If (!(IO1PIN & (1<<16)))  {
IO0CLR |= (1<<16); } else {
IO0SET |= (1<<16);  } }
return 0;  }
```

Ex No:10 INTERFACING LCD DISPLAY WITH ARM USING KEIL C

```
#include <lpc214x.h>
void initLCD(void);     void LCD_Write(unsigned int c);
void LCD_Cmd(unsigned int LCD_Cmd);
void delay(void);    int main(void) {
unsigned char ch[]="Embedotronics";
unsigned char ch1[]="Technologies";
unsigned int i,j,k,t;   initLCD();     for(i=0;ch[i]!='\0';i++)
LCD_Write(ch[i]);    LCD_Cmd(0xc3);
for(j=0;ch1[j]!='\0';j++) {  LCD_Write(ch1[j]);  }  while(1){
for(k=0;k<16;k++) {   LCD_Cmd(0x1c);
for(t=0;t<300000;t++);  } } }
void initLCD(void) {  IO0DIR = 0x0FFFF00;  delay();
LCD_Cmd(0x38);  LCD_Cmd(0x01);
LCD_Cmd(0x0c);  LCD_Cmd(0x83);
LCD_Cmd(0x06);  }  void LCD_Write(unsigned int c) {
IO0PIN = (c<<16)|(1<<10);   delay();  }
void LCD_Cmd(unsigned int LCD_Cmd) {
IO0PIN = (LCD_Cmd<<16)|(0<<10);
delay();  }  void delay(void) {
int i=0,x=0;  IO0PIN|=(1<<13);
for(i=0; i<19999; i++) { x++; }
IO0PIN&=~(1<<13);
```

Ex No: 11 DESIGN SOLUTIONS FOR SMART HOME USING ARDUINO PROCESSOR

1 Lighting Control

```
int ldrPin = A0;
int relayPin = 8;
int ldrValue;
void setup() {
pinMode(relayPin, OUTPUT); }
void loop() {
ldrValue = analogRead(ldrPin);
if (ldrValue < 300) {
digitalWrite(relayPin, HIGH);  } else {
digitalWrite(relayPin, LOW);  }
delay(1000);  }
```

2 Temperature Control

```
#include <DHT.h>
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
int relayPin = 8;
float temperature;
void setup() {
pinMode(relayPin, OUTPUT);
dht.begin(); }
void loop() {
temperature = dht.readTemperature();
if (temperature > 25.0) {
digitalWrite(relayPin, HIGH);  } else {
digitalWrite(relayPin, LOW);  }
delay(2000);  }
```