

ES-IOT Termworks

Termwork – 1

Develop an 8051 'C' program to implement MOD-4 (UP/ DOWN) counter on LEDs connected to Port 2. Include 1second delay between each count. Generate delay using for loop.

Source Code :

Connection Details:

Port 2 to CN11 of Microcontroller Evaluation Board.

```
#include "at89c51ed2.h"
```

```
void delay(unsigned int);
```

```
void main(void) {
```

```
    while(1) {
```

```
        P2=0x00;
```

```
        delay(1000);
```

```
        P2=0x10;
```

```
        delay(1000);
```

```
        P2=0x20;
```

```
        delay(1000);
```

```
        P2=0x30;
```

```
        delay(1000);
```

```
    }
```

```
}
```

```
// DELAY GENERATION USING FOR LOOP
```

```
void delay(unsigned int itime) {
```

```
    unsigned int i,j;
```

```
    for(i=0;i<itime;i++)
```

```
        for(j=0;j<1275;j++);
```

```
}
```

Termwork – 2

Develop an 8051 'C' program to implement MOD-4 (UP/ DOWN) counter on LEDs connected to Port 2. Include 0.5 second delay between each count. Generate delay using for loop.

Source Code :

Connection Details:

Port 2 to CN11 of Microcontroller Evaluation Board.

```
#include "at89c51ed2.h"
```

```
void delay(unsigned int);
```

```
void main(void) {
```

```
    while(1) {
```

```
        P2=0x00;
```

```
        delay(500);
```

```
        P2=0x10;
```

```
        delay(500);
```

```
        P2=0x20;
```

```
        delay(500);
```

```
        P2=0x30;
```

```
        delay(500);
```

```
    }
```

```
}
```

```
// DELAY GENERATION USING FOR LOOP
```

```
void delay(unsigned int itime) {
```

```
    unsigned int i,j;
```

```
    for(i=0;i<itime;i++)
```

```
        for(j=0;j<1275;j++);
```

```
}
```

Termwork – 3

Develop an 8051 'C' program to implement MOD-4 counter on LEDs connected to Port 2 using Hardware delay. Use Timer1 in Mode 1 to generate a delay of --- ms.

Source Code :

Connection Details:

Port 2 to CN11 of Microcontroller Evaluation Board.

```
#include "at89c51ed2.h"
```

```
void T1M1delay(void);
```

```
void main(void) {
```

```
    while(1) {
```

```
        P2=0x00
```

```
        T1M1delay()
```

```
        P2=0x10;
```

```
        T1M1delay();
```

```
        P2=0x20;
```

```
        T1M1delay();
```

```
        P2=0x30;
```

```
        T1M1delay();
```

```
    }
```

```
}
```

```
void T1M1Delay(void) {
```

```
    TMOD=0X10; //TIMER 1 MODE 1(16-BIT MODE)
```

```
    TL1= 0XFE; // LOAD TL1 WITH COUNT 0FE
```

```
    TH1=0X4B; // LOAD TH1 WITH COUNT 4B
```

```
    TR1=1; // START TIME
```

```
    while(TF1==0); //WAIT FOR TF1 TO ROLL OVER
```

```
    TR1=0; //TURN OFF T1
```

```
    TF1=0;          // CLEAR TF1
}
```

Termwork – 4

Develop an 8051 'C' program to implement MOD-4 counter on LEDs connected to Port 2 using Hardware delay. Use Timer1 in Mode 2 to generate a delay of ---
- ms.

Source Code :

Connection Details:

Port 2 to CN11 of Microcontroller Evaluation Board.

```
#include "at89c51ed2.h"
```

```
void T1M2delay(void);
```

```
void main(void) {
```

```
    while(1) {
```

```
        P2=0x00;
```

```
        T1M2delay()
```

```
        P2=0x10;
```

```
        T1M2delay();
```

```
        P2=0x20;
```

```
        T1M2delay();
```

```
        P2=0x30;
```

```
        T1M2delay();
```

```
    }
```

```
}
```

```
void T1M2Delay(void) {
```

```
    TMOD=0x20;
```

```
    TH1=E9;
```

```
    TR1=1;
```

```
while (TF1==0);  
TR1=0;  
TF1=0;  
}
```

Termwork – 5

Develop an 8051 'C' program to generate the following waveforms using DAC 0800 interface

- i) Square
- ii) Triangular

Source Code :

Connection Details

Port 0 to CN15.

Code For Square wave

```
#include "at89c5ed2.h"  
void delay(unsigned int);  
void main () {  
    while(1) {  
        P0 = 0x00;  
        delay(200);  
        P0 = 0xff;  
        delay(200);  
    }  
}  
  
// function to generate delay  
void delay(unsigned int itime) {  
    unsigned int i,j;  
    for(i=0;i<itime;i++)
```

```
    for(j=0;j<1275;j++);  
}
```

Code For Triangular wave

```
#include "at89c51ed2.h  
unsigned char count;  
void main () {  
    while(1) {  
        for(count=0;count!=0xff;count++) {  
            P0=count;  
        }  
        for(count=0xff; count>0;count--) {  
            P0=count;  
        }  
    }  
}
```

Termwork – 6

Develop an 8051 'C' program to generate the following waveforms using DAC 0800 interface

- i) Rectangular
- ii) Positive Ramp

Source Code :

Connection Details

Port 0 to CN15.

Code For Rectangular wave

```
#include "at89c5ed2.h"  
void delay(unsigned int);
```

```

void main () {
    while(1) {
        P0 = 0x00;
        delay(100);
        P0 = 0xff;
        delay(200);
    }
}

// function to generate delay
void delay(unsigned int itime) {
    unsigned int i,j;
    for(i=0;i<itime;i++)
        for(j=0;j<1275;j++);
}

```

Code For Positive ramp wave

```

#include "at89c51ed2.h"
unsigned char count;
void main () {
    while(1) {
        for(count=0;count<=0xff;count++) {
            P0=count;
        }
        for(count=0xff; count>0;count--) {
            P0=count;
        }
    }
}

```

Termwork – 7

Develop an 8051 'C' program to generate the following waveforms using DAC interface

- i) Square
- ii) Negative Ramp

Source Code :

Connection Details

Port 0 to CN15.

Code For Square wave

```
#include "at89c5ed2.h"
```

```
void delay(unsigned int);
```

```
void main () {
```

```
    while(1) {
```

```
        P0 = 0x00;
```

```
        delay(200);
```

```
        P0 = 0xff;
```

```
        delay(200);
```

```
    }
```

```
}
```

```
// function to generate delay
```

```
void delay(unsigned int itime) {
```

```
    unsigned int i,j;
```

```
    for(i=0;i<itime;i++)
```

```
        for(j=0;j<1275;j++);
```

```
}
```

Code For negative ramp wave

```
#include "at89c51ed2.h"
```



```

unsigned char count;

void main () {
    while(1) {
        for(count=0;count!=0xff;count++) {
            P0=count;
        }
        for(count=0xff; count>=0;count--) {
            P0=count;
        }
    }
}

```

Termwork – 8

Develop an 8051 'C' program to interface 2x16 LCD display and to display the following two strings. (Start displaying from 1st position on both lines)

- i) KLS GIT
- ii) ESIoT LAB

Source Code :

Connection Details

Port 2 to CN6 of Microcontroller Evaluation Board.

```
#include "at89c51ed2.h"
```

```
#include<intrins.h>
```

```
void lcd_init(void);
```

```
void lcd_comm(void);
```

```
void lcd_data(void);
```

```
unsigned char xdata arr[16] = {"KLS GIT"};
```

```

unsigned char xdata arr1[16] = {"ESIoT LAB"};
unsigned char temp1 = 0x00;
unsigned char temp2;
unsigned int i = 0;
void main(void) {
    AUXR = 0x10;
    lcd_init();
    temp1 = 0x80;
    lcd_comm();
    for (i = 0; i < 8; i++) {
        temp2 = arr[i];
        lcd_data();
    }
    temp1 = 0xC0;
    lcd_comm();
    for (i = 0; i < 8; i++) {
        temp2 = arr1[i];
        lcd_data();
    }
}

```

Termwork – 9

Develop an 8051 ‘C’ program to interface 2x16 LCD display and to display the following two strings. (Start displaying from 6th position on both lines)

iii) CSE

iv) BRANCH

Source Code :

Connection Details

Port 2 to CN6 of Microcontroller Evaluation Board.

```
#include "at89c51ed2.h"
```

```
#include<intrins.h>
```

```
void lcd_init(void);
```

```
void lcd_comm(void);
```

```
void lcd_data(void);
```

```
unsigned char xdata arr[16] = {"    CSE"};          //6 spaces for 6th position
```

```
unsigned char xdata arr1[16] = {"    BRANCH"}; //6 spaces for 6th position
```

```
unsigned char temp1 = 0x00;
```

```
unsigned char temp2;
```

```
unsigned int i = 0;
```

```
void main(void) {
```

```
    AUXR = 0x10;
```

```
    lcd_init();
```

```
    temp1 = 0x80;
```

```
    lcd_comm();
```

```
    for (i = 0; i < 8; i++) {
```

```
        temp2 = arr[i];
```

```
        lcd_data();
```

```
    }
```

```
    temp1 = 0xC0;
```

```
    lcd_comm();
```

```
    for (i = 0; i < 8; i++) {
```

```
        temp2 = arr1[i];
```

```
        lcd_data();
```

```
}  
  
}
```

Termwork – 10

Develop an Embedded 'C' program to blink the LEDs connected to Arduino SBC upon pressing the push buttons.

Source Code :

Connection Details

Connect CN9 To CN4

```
const int buttonPin1 = 13;
```

```
int buttonState1 = LOW;
```

```
const ledPin1 = A5;
```

```
const int buttonPin2 = 12;
```

```
int buttonState2 = LOW;
```

```
const ledPin2 = A4;
```

```
const int buttonPin3 = 11;
```

```
int buttonState3 = LOW;
```

```
const ledPin3 = A3;
```

```
const int buttonPin4 = 10;
```

```
int buttonState4 = LOW;
```

```
const ledPin4 = A2;
```

```
void setup() {
```

```
    pinMode(buttonPin1, INPUT);
```

```
    pinMode(ledPin1, OUTPUT);
```

```
    pinMode(buttonPin2, INPUT);
```

```
    pinMode(ledPin2, OUTPUT);
```

```
    pinMode(buttonPin3, INPUT);
```

```
pinMode(ledPin3, OUTPUT);
pinMode(buttonPin4, INPUT);
pinMode(ledPin4, OUTPUT);
Serial.begin(9600);
}

void loop() {
    buttonState1 = digitalRead(buttonPin1);
    buttonState2 = digitalRead(buttonPin2);
    buttonState3 = digitalRead(buttonPin3);
    buttonState4 = digitalRead(buttonPin4);
    if (buttonState1 == HIGH){
        digitalWrite(ledPin1, LOW);
    }else{
        digitalWrite(ledPin1, HIGH);
    }
    Serial.println(buttonState1);
    if (buttonState2 == HIGH){
        digitalWrite(ledPin2, LOW);
    }else{
        digitalWrite(ledPin2, HIGH);
        delay(10);
    }
    Serial.println(buttonState2);
    if (buttonState3 == HIGH){
        digitalWrite(ledPin3, LOW);
    }else{
        digitalWrite(ledPin3, HIGH);
```

```

        delay(10);
    }
    Serial.println(buttonState3);
    if (buttonState4 == HIGH){
        digitalWrite(ledPin4, LOW);
    }else{
        digitalWrite(ledPin4, HIGH);
        delay(10);
    }
    Serial.println(buttonState4);
}

```

Termwork – 11

Develop an Embedded ‘C’ program to interface the sensor DHT11 to Arduino SBC and display the data acquired from sensors on serial monitor.

Source Code :

Connection Details

Connect RM2 – RM19.

// INCLUDE DHT LIBRARY (* Sketch → Include Library→ add ZIP library)

```
#include <dht.h>
```

```
dht DHT;
```

```
#define DHT11_PIN 4
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
    int chk = DHT.read11(DHT11_PIN);
```

```
    Serial.print("Temperature = ");
```

```
    Serial.println(DHT.temperature);
```

```
Serial.print("Humidity = ");  
Serial.println(DHT.humidity);  
delay(2000);  
}
```

Termwork – 12

Develop an Embedded 'C' program to control the relay through Arduino UNO.

Source Code :

Connection Details

Connect RM17 To RM9

```
int relay_pin = 8;  
  
void setup() {  
    pinMode(relay_pin, OUTPUT);  
    Serial.begin(9600);  
    digitalWrite(relay_pin, HIGH);  
}  
  
void loop() {  
    digitalWrite(relay_pin, LOW);  
    delay(1000);  
    digitalWrite(relay_pin, HIGH);  
    delay(1000);  
}
```

Termwork – 13

Develop an Embedded 'C' program to interface the sensor LDR to Arduino SBC and display the data acquired from sensor on serial monitor.

Source Code :

Connection Details

Connect RM3 – RM20.

```
int light_pin = 5;

void setup(){
    pinMode(light_pin, INPUT);
    Serial.begin(9600);
}

void loop(){
    int light_data = digitalRead (light_pin);
    if(light_data)
        Serial.println("Light Not Detected!");
    else
        Serial.println("Light Detected!");
    delay(1000);
}
```

Termwork – 14

Develop an Embedded ‘C’ program to blink the LEDs connected to Arduino SBC upon pressing the push buttons.

Source Code :

Connection Details

Connect CN9 To CN4

```
const int buttonPin1 = 13;
int buttonState1 = LOW;
const ledPin1 = A5;
const int buttonPin2 = 12;
int buttonState2 = LOW;
```



```
const ledPin2 = A4;
const int buttonPin3 = 11;
int buttonState3 = LOW;
const ledPin3 = A3;
const int buttonPin4 = 10;
int buttonState4 = LOW;
const ledPin4 = A2;

void setup() {
    pinMode(buttonPin1, INPUT);
    pinMode(ledPin1, OUTPUT);
    pinMode(buttonPin2, INPUT);
    pinMode(ledPin2, OUTPUT);
    pinMode(buttonPin3, INPUT);
    pinMode(ledPin3, OUTPUT);
    pinMode(buttonPin4, INPUT);
    pinMode(ledPin4, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    buttonState1 = digitalRead(buttonPin1);
    buttonState2 = digitalRead(buttonPin2);
    buttonState3 = digitalRead(buttonPin3);
    buttonState4 = digitalRead(buttonPin4);
    if (buttonState1 == HIGH){
        digitalWrite(ledPin1, LOW);
    }else{
```

```
        digitalWrite(ledPin1, HIGH);
    }
    Serial.println(buttonState1);
    if (buttonState2 == HIGH){
        digitalWrite(ledPin2, LOW);
    }else{
        digitalWrite(ledPin2, HIGH);
        delay(10);
    }
    Serial.println(buttonState2);
    if (buttonState3 == HIGH){
        digitalWrite(ledPin3, LOW);
    }else{
        digitalWrite(ledPin3, HIGH);
        delay(10);
    }
    Serial.println(buttonState3);
    if (buttonState4 == HIGH){
        digitalWrite(ledPin4, LOW);
    }else{
        digitalWrite(ledPin4, HIGH);
        delay(10);
    }
    Serial.println(buttonState4);
}
```

Develop an Embedded 'C' program to interface the sensor DHT11 to Arduino SBC and display the data acquired from sensors on serial monitor. Turn ON the relay when temperature is greater than 22 degrees centigrade.

Source Code :

Connection Details

Connect RM2 – RM19.

Connect RM17 To RM9

```
// INCLUDE DHT LIBRARY (* Sketch → Include Library→ add ZIP library)
```

```
#include <dht.h>
```

```
dht DHT;
```

```
#define DHT11_PIN 4
```

```
int relay_pin = 8;
```

```
void setup() {
```

```
    pinMode(relay_pin, OUTPUT);
```

```
    Serial.begin(9600);
```

```
    digitalWrite(relay_pin, HIGH);
```

```
}
```

```
void loop() {
```

```
    int chk = DHT.read11(DHT11_PIN);
```

```
    Serial.print("Temperature = ");
```

```
    Serial.println(DHT.temperature);
```

```
    if(DHT.temperature){
```

```
        digitalWrite(relay_pin, HIGH);
```

```
    }
```

```
    else{
```

```
        digitalWrite(relay_pin, LOW);
```

```
    }
```

```
    delay(2000);
```

```
}
```