ES-IOT Termworks

Termwork – 1

Develop an 8051 ‘C’ program to implement MOD-4 counter on LEDs connected to Port 2 using

i) Software delay

ii) Hardware delay

to generate some delay.

Source Code :

Connection Details:

Port 2 to CN11 of Microcontroller Evaluation Board.

i)

```
#include "at89c51ed2.h“
void delay(unsigned int);
void main(void) {
        while(1) {
                P2=0x00;
                delay(250);
                P2=0x10;
                delay(250);
                P2=0x20;
                delay(250);
                P2=0x30;
                delay(250);
        }
}
//SOFTWARE DELAY GENERATION
```

```
void delay(unsigned int itime) {
        unsigned int i,j;
        for(i=0;i<itime;i++)
        for(j=0;j<1275;j++);
}
ii) #include “at89c51ed2.h”
void T0M1delay(void);
void main(void) {
        while(1) {
                P2=0x00
                delay()
                P2=0x10;
                delay();
                P2=0x20;
                delay();
                P2=0x30;
                delay();
        }
}
// HARDWARE DELAY GENERATION USING TIMER 0 IN MODE1
void T0M1Delay(void) {
        TMOD=0X01;   //TIMER 0 MODE 1(16-BIT MODE)
        TL0= 0XFE;    // LOAD TL0 WITH COUNT 0FE
        TH0=0X4B;     // LOAD TH0 WITH COUNT 4B
        TR0=1;        // START TIME
        while(TF0==0); //WAIT FOR TF0 TO ROLL OVER
        TR0=0;        //TURN OFF T0
        TF0=0;        // CLEAR TF0
```

```
}
```

Termwork – 2

Develop 8051 ‘C’ program to generate the following waveforms using DAC interface

i) Square/ Rectangular

ii) Triangular

Source Code :

Connection Details

Port 0 to CN15.

Code For Square wave

```
#include “at89c5ed2.h”
void delay(void);
void main () {
        while(1) {
        P0 = 0x00; // program outputs 0 and 0xff alternatively to port P0 with a delay
        delay();
        P0 = 0xff;
        delay();
        }
}
// function to generate delay
void delay(void) {
        int i;
        for(i=0;i<=300;i++);
}
```

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 – 3

Develop 8051 ‘C’ program to interface 2x16 LCD display and to display two strings

Source Code :

Connection Details

Port 2 to CN6 of Microcontroller Evaluation Board.

```
#include "at89c51ed2.h"
#include <intrins.h>
// LCD FUNCTION PROTOTYPE
void lcd_init(void);
void lcd_comm(void);
void lcd_data(void);
unsigned char xdata arr[16]={“GITCSE"};
unsigned char temp1=0x00;
unsigned char temp2;
unsigned int i=0;

void main(void) {
        AUXR = 0x10;            // Accessing Full XRAM
        lcd_init();
        temp1 = 0x80;           // To display in 2nd line use - temp1=0XC0
        lcd_comm();             // Command writing
        for(i=0;i<6;i++) {
                temp2 = arr[i];
                lcd_data();     // Data writing
        }
        while(1) {
        }
}
```

Termwork – 4 (not in exams.. maybe)

Develop 8051 ‘C’ program to display the temperature sensor output from ADC 0809 on the LCD.

Source Code :

Connection Details

CN1- CN14

CN2- CN13

CN3- CN6

```
#include<reg51.h>
sbit ALE= P2^4;
sbit OE = P2^5;
sbit SC= P2^6;
sbit EOC= P2^7;
sbit ADDR_A =P2^0;
sbit ADDR_B =P2^1;
sbit ADDR_C =P2^2;
sfr MYDATA=P1;
void main()
{
       unsigned char value;
        MYDATA=0xFF;                    // make P1 as input
       EOC=1;                // make EOC as input
       ALE=0;                // clear ALE
       OE=0;                 // clear OE
       SC=0;                 // clear SC
    while(1)
       {
              ADDR_C=0;
              ADDR_B=0;
              ADDR_A=1;      // Select channel 1 with ABC=001
              MSDelay(1);    // wait for 1ms

              ALE=1;         // Latch the address (L to H pulse)

              MSDelay(1);

              SC=1;          // Start conversion (H to L pulse)
              MSDelay(1);
              SC=0;          // Start conversion


              while(EOC==1); // wait for data conversion (H to L indicates end of
the conversion)
              while(EOC==0);

              OE=1;          // Enable RD (L TO H PULSE)
               MSDelay(1);
              value=MYDATA;  // get the data
              OE=0;          // disable RD for the next round

              ConvertAndDisplay(value);
       }
}
```

Termwork – 5

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 – 6

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 – 7

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 – 8

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 = digitalReal (light_pin);
        if(light_data)
                Serial.println(“Light Not Detected!”);
        else
                Serial.println(“Light Detected!”);
        delay(1000);
}
```