## Experiment-1A:Software Delay - Code

### 

#include "at89c51ed2.h"

void delay(unsigned int);

void main(void) {

while(1) {

P0 = 0x00;

delay(100);

P0 = 0x10;

delay(100);

P0 = 0x20;

delay(100);

P0 = 0x30;

delay(100);

}

}

void delay(unsigned int itime) {

unsigned int i, j;

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

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

}

}

## Experiment-1B: Hardware Delay - Code

#include "at89c51ed2.h"

void T0M1Delay();

void main(void) {

while(1) {

P0 = 0x00;

T0M1Delay();

P0 = 0x10;

T0M1Delay();

P0 = 0x20;

T0M1Delay();

P0 = 0x30;

T0M1Delay();

}

}

void T0M1Delay() {

unsigned char z;

for (z = 0; z < 5; z++) {

TMOD = 0x01; // configuring Timer 0 in mode 1

TH0 = 0x4B;

TL0 = 0xFD;

TR0 = 1; // start the timer

while (TF0 == 0); // monitor the timer overflow flag

TR0 = 0; // stop the timer

TF0 = 0; // clear timer overflow flag

}

}

## Experiment-2: Code

##### Square and Rectangular waveform

#include "at89c51ed2.h"

void delay(unsigned int);

void main(void) {

while(1) {

P0 = 0x00;

delay(200);

P0 = 0xff;

delay(200);

}

}

void delay(unsigned int itime) {

unsigned int i, j;

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

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

}

}

Square wave: equal delay (200, 200)

Rectangular wave: unequal delay (100, 200)

##### Triangular waveform

#include "at89c51ed2.h"

void main(void) {

unsigned char count;

while(1) {

for (count = 0; count != 0xff; count++)

P0 = count;

for (count = 0xff; count != 0; count--)

P0 = count;

}

}

##### Positive ramp

#include "at89c51ed2.h"

void main(void) {

unsigned char count;

while(1) {

for (count = 0; count != 0xff; count++){

P0 = count;

}

P0 = 0;

}

}

##### Negative ramp

#include "at89c51ed2.h"

void main(void) {

unsigned char count;

while(1) {

for (count = 0xff; count != 0; count–){

P0 = count;

}

P0 = 0xff;

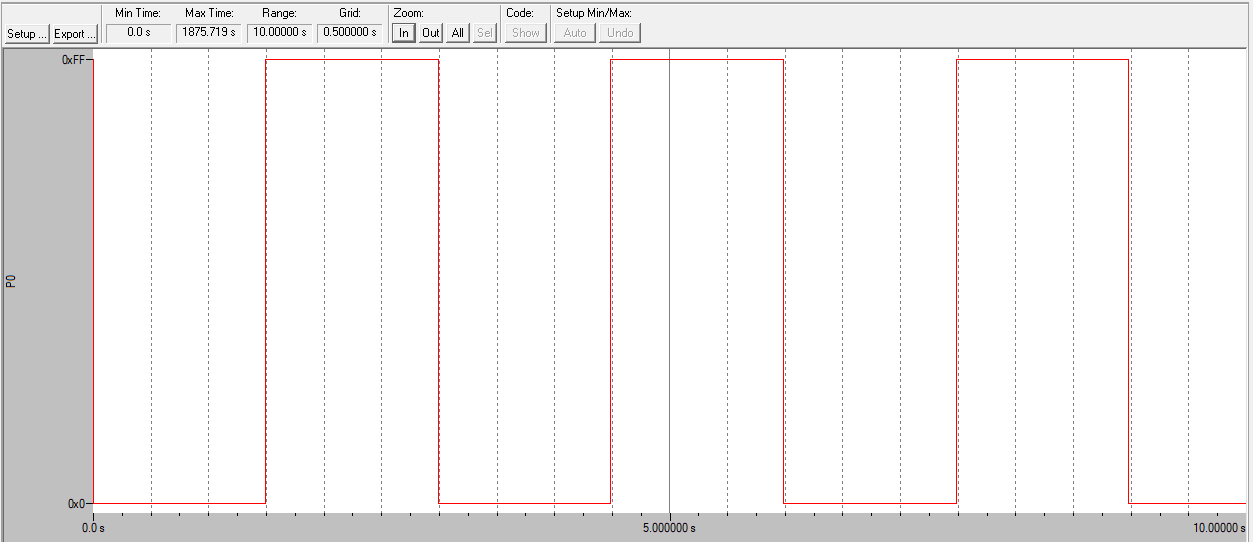
}

}

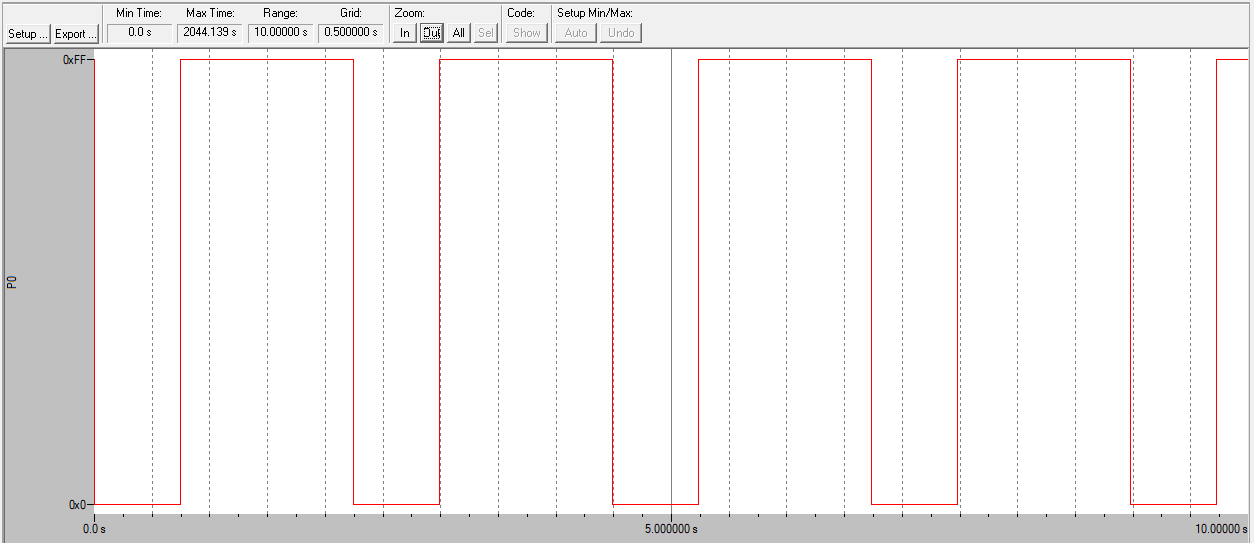
## Experiment-2: Outputs

##### 

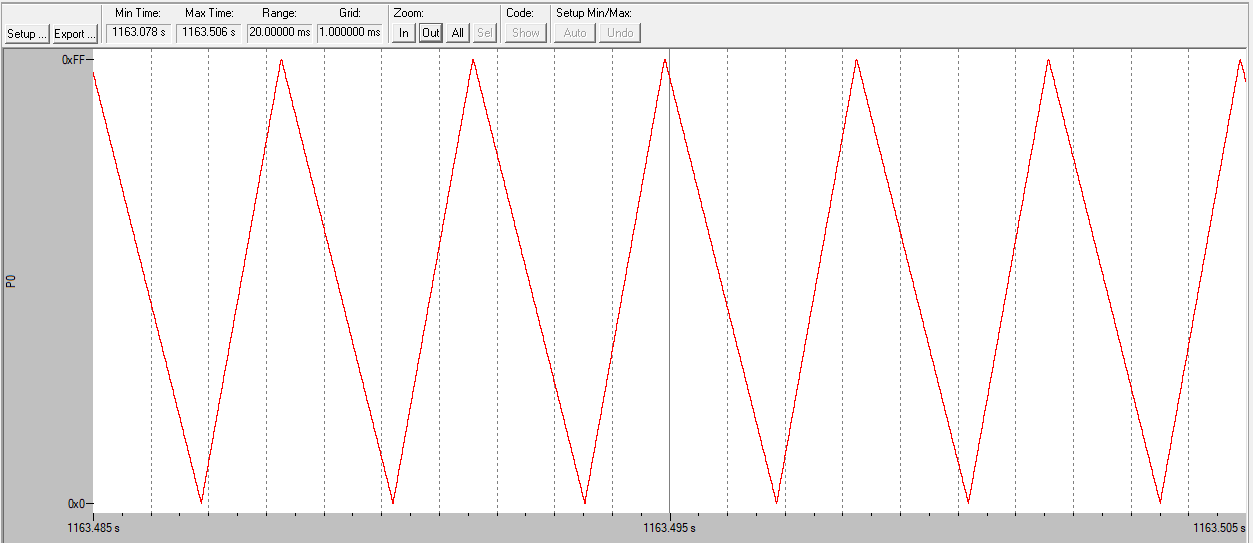
##### Square wave (TON = TOFF)



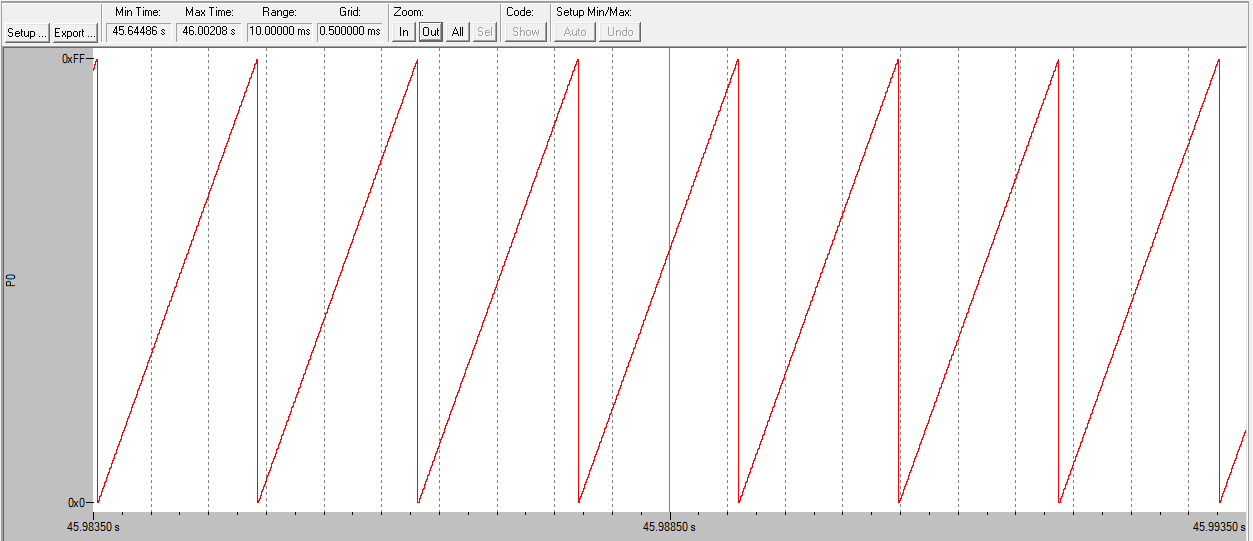
##### Rectangular wave (TON ≠ TOFF)



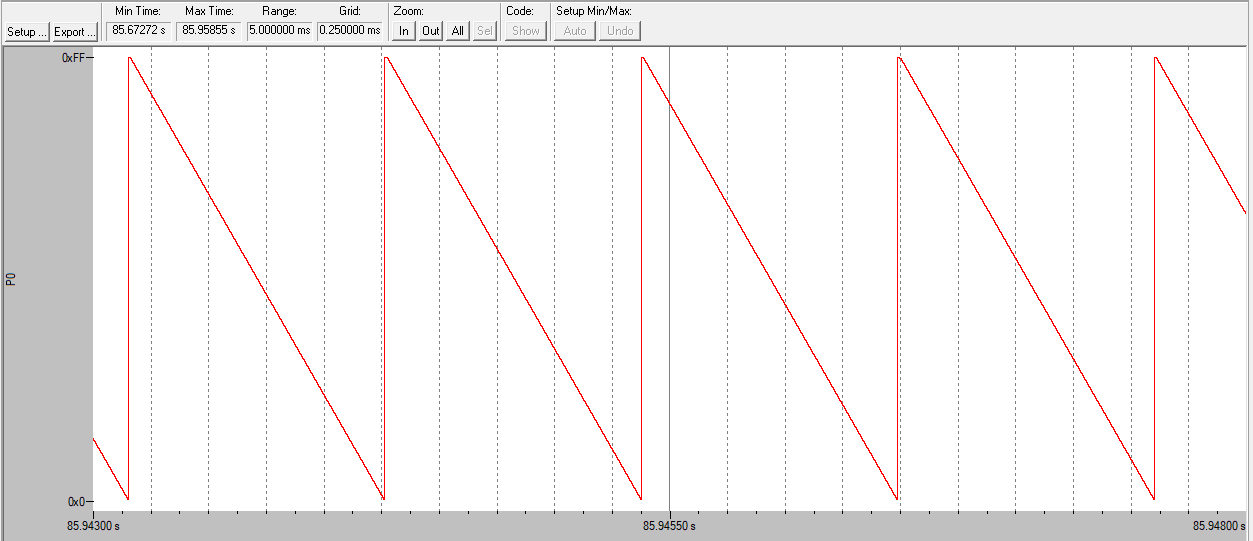
##### Triangular wave



##### Positive ramp



##### Negative ramp



## Experiment-3: Code

NOTE: Add LCD\_ROUTINE header to the source group

#### Case-1 (output on first row)

##### Code

#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] = {"HELLO WORLD"};

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 < 12; i++) {

temp2 = arr[i];

lcd\_data();

}

}

##### Output



#### 

#### Case-2 (output on rows 1&2)

##### Code

#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] = {"Line One"};

unsigned char xdata arr1[16] = {"Line Two"};

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();

}

}

##### Output

## 

## 

## Experiment-5

##### Code

const int buttonPin1 = 13;

int buttonState1 = LOW;

const int ledPin1 = A5;

const int buttonPin2 = 12;

int buttonState2 = LOW;

const int ledPin2 = A4;

const int buttonPin3 = 11;

int buttonState3 = LOW;

const int ledPin3 = A3;

const int buttonPin4 = 10;

int buttonState4 = LOW;

const int ledPin4 = A2;

void setup() {

// put your setup code here, to run once:

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() {

// put your main code here, to run repeatedly:

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);

}

Serial.println(buttonState2);

if (buttonState3 == HIGH) {

digitalWrite(ledPin3, LOW);

}

else {

digitalWrite(ledPin3, HIGH);

}

Serial.println(buttonState3);

if (buttonState4 == HIGH) {

delay(1000);

digitalWrite(ledPin4, LOW);

}

else {

digitalWrite(ledPin4, HIGH);

}

Serial.println(buttonState4);

}

##### 

## Experiment-6

##### Code

#include <dht.h>

#define DHT11\_PIN 4

dht DHT;

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

int chk = DHT.read11(DHT11\_PIN);

Serial.print("Temperature: ");

Serial.println(DHT.temperature);

Serial.print("Humidity: ");

Serial.println(DHT.humidity);

delay(200);

}

##### Output

## Experiment-7

##### Code

int light\_pin = 5;

void setup() {

// put your setup code here, to run once:

pinMode(light\_pin, INPUT);

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

int light\_data = digitalRead(light\_pin);

if (light\_data)

Serial.println("Light not detected!");

else

Serial.println("Light detected!");

delay(1000);

}

##### Output

## Experiment-8

##### Code

int relay\_pin = 8;

void setup() {

// put your setup code here, to run once:

pinMode(relay\_pin, OUTPUT);

Serial.begin(9600);

digitalWrite(relay\_pin, HIGH);

}

void loop() {

// put your main code here, to run repeatedly:

digitalWrite(relay\_pin, LOW);

Serial.println("Relay is OFF");

delay(1000);

digitalWrite(relay\_pin, HIGH);

Serial.println("Relay is ON");

delay(1000);

}