

# گزارش کار سوم آزمایشگاه ریزپردازنده

تهیه و تنظیم: مبین خیبری

شماره دانشجویی: 994421017

استاد راهنما: دکتر نیکزاد

## چکیده:

در این جلسه، ابتدا اقدامات انجام شده در جلسه‌ی گذشته مورد بررسی قرار گرفتند و سپس توجه گروه‌های مختلف دانشجویان روی طراحی و پیاده‌سازی سخت‌افزاری و نرم‌افزاری خواسته‌های آزمایش بعدی معطوف گردید.

در این گزارش کار قصد داریم مراحل مختلف لازم جهت طراحی یک تایمر یا شمارنده به کمک یک برد آردوینو و نیز چند قطعه سون سگمنت را قدم به قدم شرح دهیم. به همین منظور، ابتدا به معرفی کلی قطعات سون سگمنت پرداخته و سپس ادامه‌ی گزارش کار را به توضیح مراحل لازم برای پیاده‌سازی شمارنده اختصاص می‌دهیم.

## معرفی سون سگمنت

وقتی LED را با یک ولتاژ خارجی بایاس مستقیم کنیم، از پیوند PN آن جریان می‌گذرد. در الکترونیک، این پدیده را نورتایی الکتریکی (Electroluminescence) می‌نامیم. قطعاتی وجود دارند که از ترکیب LEDها تشکیل شده و برای نمایش اعداد یا حروف به کار می‌روند. این قطعات نمایش گر را سون سگمنت می‌نامند.

رنگ واقعی نور مرئی ساطع شده از یک LED، از آبی تا قرمز و نارنجی وجود دارد. این رنگ‌ها، با طول موج طیفی نور ساطع شده تعیین می‌شوند که خود این طول موج‌ها به ترکیب ناخالصی‌هایی بستگی دارد که به مواد نیمه‌رسانای تشکیل دهنده آن افزوده می‌شود.

دیودهای نورانی یا LEDها، مزایای مختلفی نسبت به لامپ‌های سنتی دارند که از مهم‌ترین آنها می‌توان به اندازه کوچک، طول عمر بالا، داشتن رنگ‌های مختلف، ارزان بودن و به آسانی در دسترس بودن اشاره کرد. این دیودها را می‌توان به آسانی با قطعات الکترونیک دیگر و مدارهای دیجیتال ترکیب کرد.

ویژگی کوچک بودن LEDها سبب می‌شود تعدادی از آنها را در کنار یکدیگر قرار داده و بسته‌هایی تشکیل شوند که نمایشگر هفت قسمتی یا سون سگمنت (7-Segment) نامیده می‌شوند.

نمایشگر سون سگمنت، از هفت LED تشکیل شده که مطابق شکل زیر در کنار یکدیگر قرار گرفته‌اند.

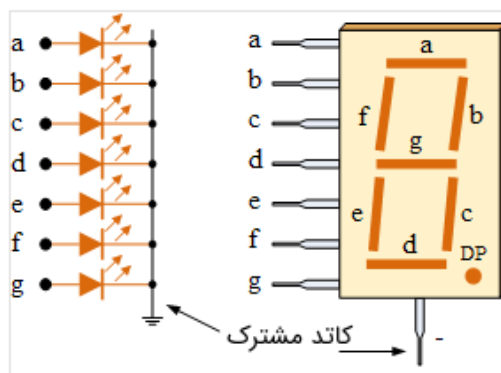


هر یک از این LEDها یک قسمت یا سگمنت نامیده می‌شوند، زیرا وقتی هر یک از آنها روشن می‌شود، بخشی از یک عدد یا حرف را روشن می‌کند. گاهی اوقات یک LED هشتم نیز در سون سگمنت وجود دارد که نقطه دسیمال (DP) را مشخص می‌کند. واضح است که این نقطه وقتی کاربرد دارد که دو سون سگمنت در کنار یکدیگر قرار گیرند.

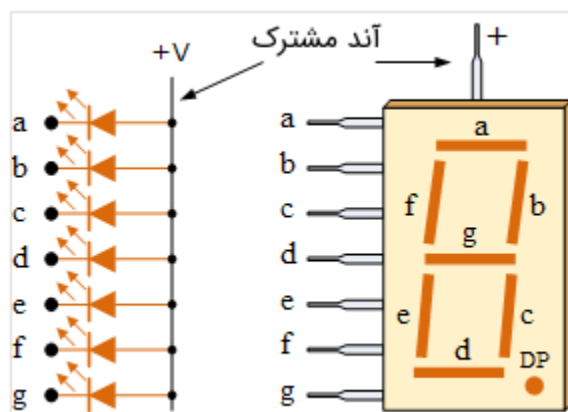
پایه‌های LEDهای سون سگمنت، در کنار یکدیگر و در زیر قطعه چیده شده‌اند. این پایه‌ها به گونه‌ای نام‌گذاری شده‌اند که ارتباط هر کدام از پایه‌ها به LED متناظر با خود مشخص است. یکی از پایه‌ها، نوع سون سگمنت را مشخص می‌کند. همان‌طور که می‌دانیم، هر LED دو پایه دارد که یکی از آنها آند و دیگری کاتد نامیده می‌شود. بر همین اساس، دو نوع سون سگمنت نیز خواهیم داشت: کاتد مشترک (Common Cathode) یا CC و آند مشترک (Common Anode) یا CA.

تفاوت بین این دو نمایشگر، همان‌گونه که از نامشان پیداست، این است که در نوع کاتد مشترک، کاتد دیودها به یکدیگر متصل است و در نوع آند مشترک، آند آنها به یکدیگر وصل شده است.

در سون سگمنت کاتد مشترک، همه کاتدها به یکدیگر متصل شده و با هم به منطق 0 یا زمین وصل می‌شوند. بنابراین، هر سگمنت را می‌توان با اعمال سیگنال HIGH یا 1 منطقی از طریق یک مقاومت محدود کننده به یکی از ترمینال‌های آند (a تا g)، روشن کرد. شکل زیر، نمایشگر سون سگمنت کاتد مشترک را نشان می‌دهد.

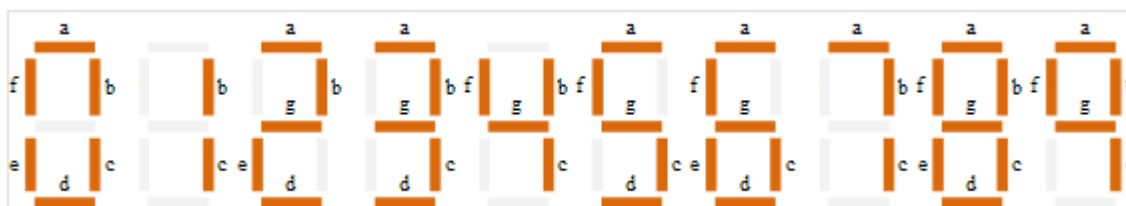


در سون سگمنت آند مشترک، همه آندها به یکدیگر متصل شده و با هم به منطق 1 وصل می‌شوند. بنابراین، هر سگمنت را می‌توان با اعمال سیگنال LOW یا 0 منطقی از طریق یک مقاومت محدود کننده به یکی از ترمینال‌های کاتد (a تا g) روشن کرد. شکل زیر، نمایشگر سون سگمنت آند مشترک را نشان می‌دهد.



در حالت کلی، نمایشگرهای آند مشترک نسبت به انواع کاتد مشترک، محبوب‌تر هستند؛ زیرا تعداد زیادی از مدارهای منطقی مصرف کننده جریان (و نه منبع آن) هستند. لازم به ذکر است که یک سون سگمنت کاتد مشترک را نمی‌توان مستقیماً به جای سون سگمنت آند مشترک در مدار قرار داد و بالعکس؛ زیرا این کار دقیقاً مانند این است که پایه‌های یک ال ای دی را برعکس متصل کنیم؛ واضح است که در این حالت، LED روشن نمی‌شود و نوری تولید نخواهد کرد.

بسته به عدد دسیمالی که می‌خواهیم نشان دهیم، باید مجموعه مشخص متناظری از LEDهای سون سگمنت را با هم بایاس مستقیم و در نتیجه روشن کنیم. برای مثال، برای نمایش رقم 0 باید شش ال ای دی a تا f را روشن کنیم. به همین ترتیب، ارقام 0 تا 9 را می‌توان مطابق شکل زیر نشان داد.



بنابراین، برای یک سون سگمنت می‌توان جدول درستی تشکیل داد و سگمنت‌هایی را مشخص کرد که لازم است برای نمایش یک رقم خاص، از 0 تا 9 روشن شوند.

رقم دسیمال	بخش روشن						
	a	b	c	d	e	f	g
0	x	x	x	x	x	x	
1		x	x				
2	x	x		x	x		x
3	x	x	x	x			x
4		x	x			x	x
5	x		x	x		x	x
6	x		x	x	x	x	x
7	x	x	x				
8	x	x	x	x	x	x	x
9	x	x	x			x	x

برای ساخت و طراحی یک شمارنده، به تعدادی قطعه و برد الکترونیکی نیاز داریم که در تصاویر زیر آورده شده‌اند. البته در کلاس درس برای پیاده‌سازی این آزمایش از کیت‌های کامل و آماده‌ای استفاده شد که تمام این موارد را از قبل شامل بودند.



Arduino Mega 2560

Any arduino should work



4-digit 7-segment display



Push button



Buzzer

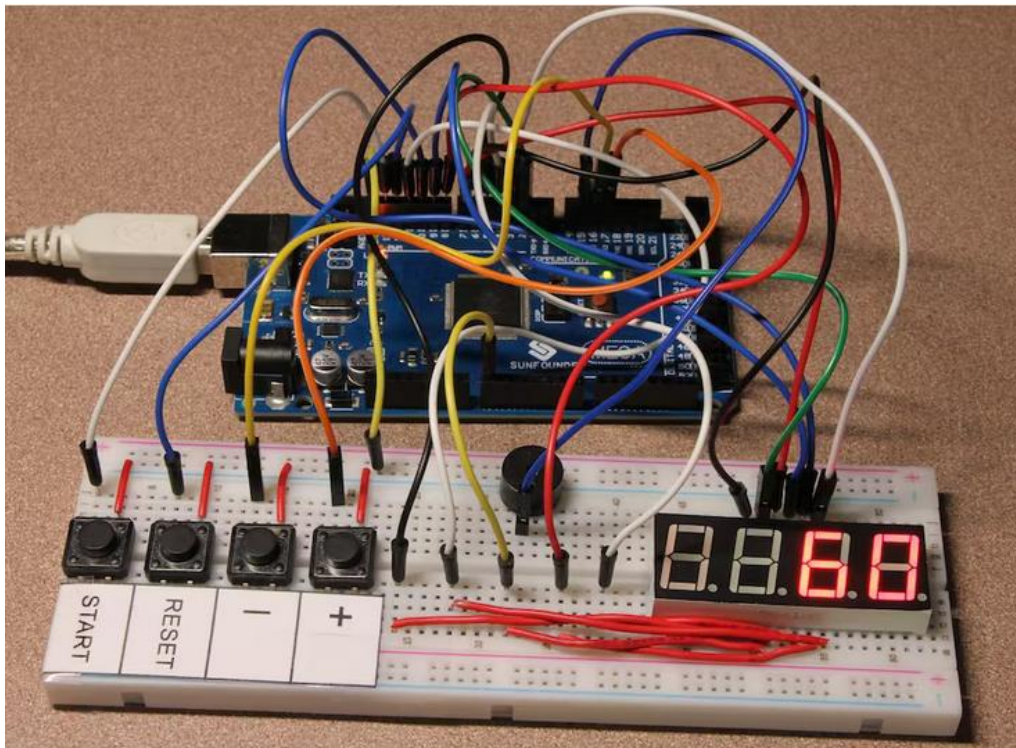


Jumper wires (generic)

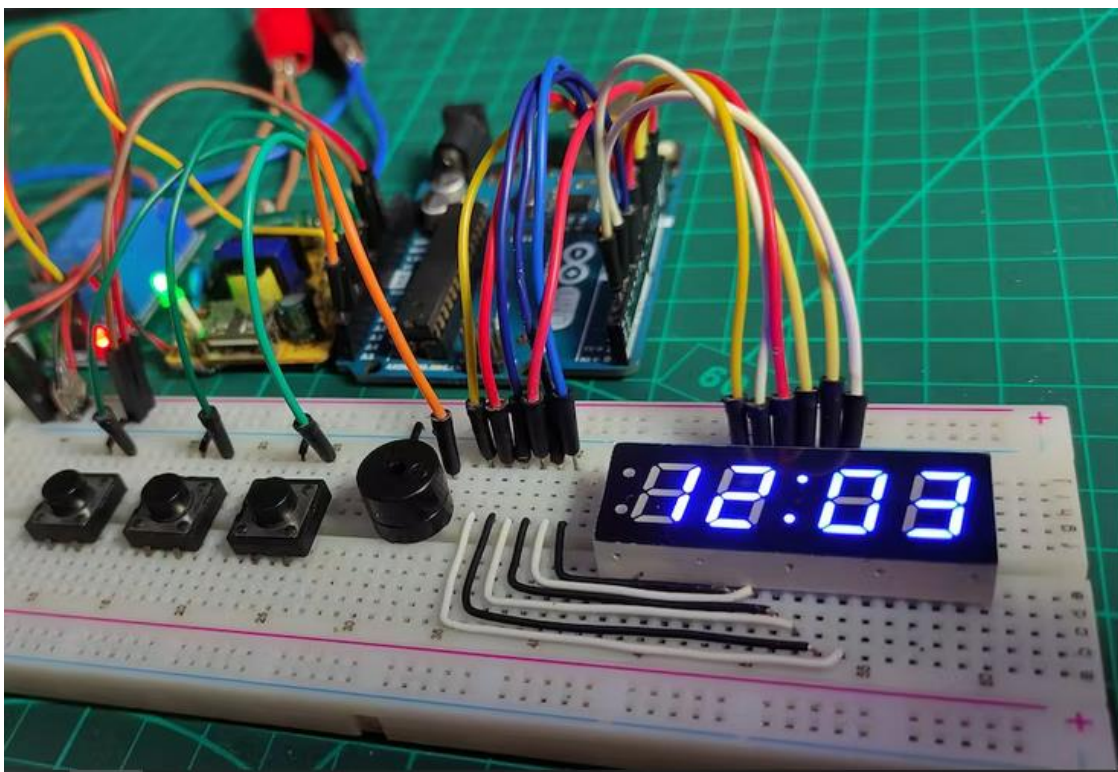


Solderless Breadboard Full Size

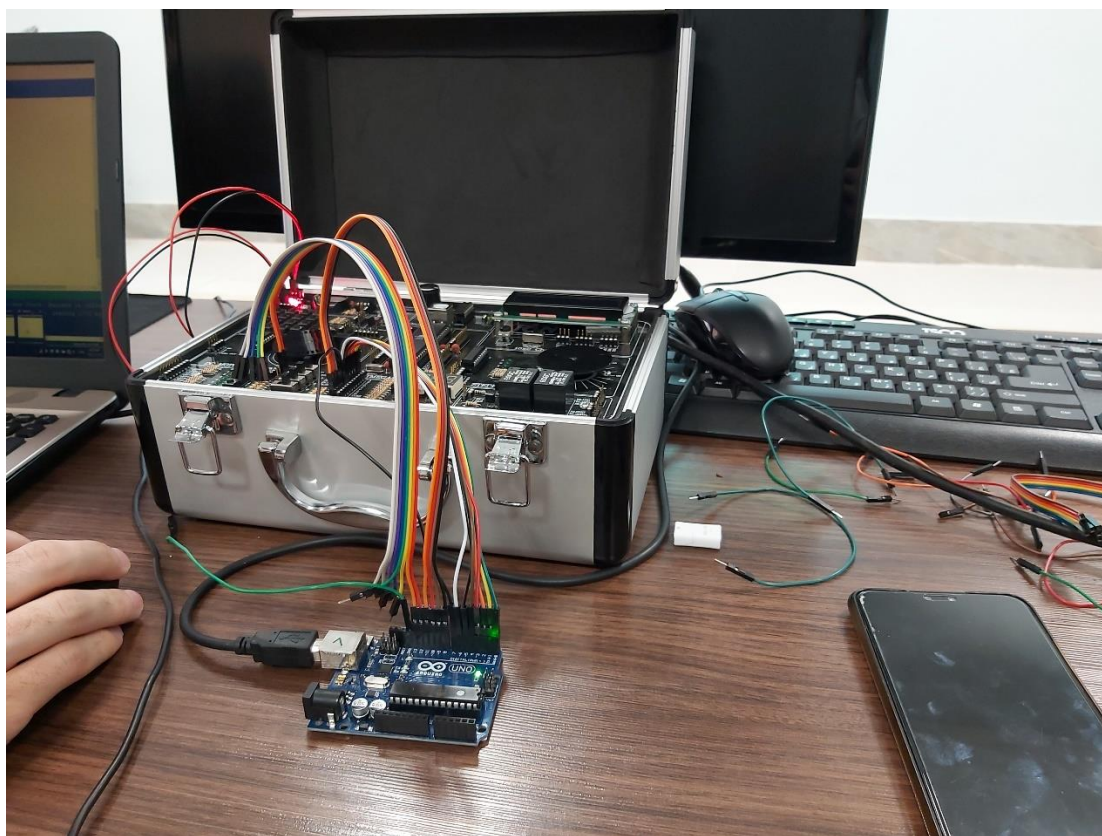
تصاویر زیر به طور خلاصه نتیجه‌ی نهایی اتصالات ایجاد شده در بردها را به کمک اجزای یادشده نشان می‌دهند:







تصویر بعدی، برد پیاده‌سازی شده در آزمایشگاه ریزپردازنده به کمک کیت‌های آماده را نشان می‌دهد:



پس از پایان طراحی اتصالات، نوبت به طراحی نرم‌افزاری و شبیه‌سازی عملکرد تایمر در نرم‌افزارِ Arduino IDE می‌رسد.

کدهای زیر، پیاده‌سازیِ یک شمارشگر با بهره‌گرفتن از کتابخانه‌های زبانِ C را نشان می‌دهند. با کامپایل کردن و ارسال دستوراتِ زیر به بردِ آردوینو، شمارشگر بر اساسِ منطقی طراحی‌شده شروع به کار خواهد کرد.

```
#include <math.h>

int digit_pin[] = {6, 9, 10, 11};

int speakerPin = 15;

#define DIGIT_ON LOW
#define DIGIT_OFF HIGH

int segA = 2;
int segB = 3;
int segC = 4;
int segD = 5;
int segE = A0;
int segF = 7;
int segG = 8;
//int segPD = ;
int button1=13;
int button2=12;
int button3=16;
int button4=17;
int countdown_time = 60;
struct struct_digits {
    int digit[4];
};

void setup() {
    pinMode(segA, OUTPUT);
```

```
pinMode(segB, OUTPUT);
pinMode(segC, OUTPUT);
pinMode(segD, OUTPUT);
pinMode(segE, OUTPUT);
pinMode(segF, OUTPUT);
pinMode(segG, OUTPUT);
for (int i=0; i<4; i++) {
    pinMode(digit_pin[i], OUTPUT);
}
pinMode(speakerPin, OUTPUT);
pinMode(button1, INPUT_PULLUP);
pinMode(button2, INPUT_PULLUP);
pinMode(button3, INPUT_PULLUP);
pinMode(button4, INPUT_PULLUP);
}

void playTone(int tone, int duration) {
    for (long k = 0; k < duration * 1000L; k += tone * 2) {
        digitalWrite(speakerPin, HIGH);
        delayMicroseconds(tone);
        digitalWrite(speakerPin, LOW);
        delayMicroseconds(tone);
    }
}

void lightNumber(int numberToDisplay) {

#define SEGMENT_ON HIGH
#define SEGMENT_OFF LOW
```



```
switch (numberToDisplay){  
case 0:  
    digitalWrite(segA, SEGMENT_ON);  
    digitalWrite(segB, SEGMENT_ON);  
    digitalWrite(segC, SEGMENT_ON);  
    digitalWrite(segD, SEGMENT_ON);  
    digitalWrite(segE, SEGMENT_ON);  
    digitalWrite(segF, SEGMENT_ON);  
    digitalWrite(segG, SEGMENT_OFF);  
    break;
```

```
case 1:  
    digitalWrite(segA, SEGMENT_OFF);  
    digitalWrite(segB, SEGMENT_ON);  
    digitalWrite(segC, SEGMENT_ON);  
    digitalWrite(segD, SEGMENT_OFF);  
    digitalWrite(segE, SEGMENT_OFF);  
    digitalWrite(segF, SEGMENT_OFF);  
    digitalWrite(segG, SEGMENT_OFF);  
    break;
```

```
case 2:  
    digitalWrite(segA, SEGMENT_ON);  
    digitalWrite(segB, SEGMENT_ON);  
    digitalWrite(segC, SEGMENT_OFF);  
    digitalWrite(segD, SEGMENT_ON);  
    digitalWrite(segE, SEGMENT_ON);  
    digitalWrite(segF, SEGMENT_OFF);  
    digitalWrite(segG, SEGMENT_ON);
```

```
break;
```

case 3:

```
digitalWrite(segA, SEGMENT_ON);  
digitalWrite(segB, SEGMENT_ON);  
digitalWrite(segC, SEGMENT_ON);  
digitalWrite(segD, SEGMENT_ON);  
digitalWrite(segE, SEGMENT_OFF);  
digitalWrite(segF, SEGMENT_OFF);  
digitalWrite(segG, SEGMENT_ON);  
break;
```

case 4:

```
digitalWrite(segA, SEGMENT_OFF);  
digitalWrite(segB, SEGMENT_ON);  
digitalWrite(segC, SEGMENT_ON);  
digitalWrite(segD, SEGMENT_OFF);  
digitalWrite(segE, SEGMENT_OFF);  
digitalWrite(segF, SEGMENT_ON);  
digitalWrite(segG, SEGMENT_ON);  
break;
```

case 5:

```
digitalWrite(segA, SEGMENT_ON);  
digitalWrite(segB, SEGMENT_OFF);  
digitalWrite(segC, SEGMENT_ON);  
digitalWrite(segD, SEGMENT_ON);  
digitalWrite(segE, SEGMENT_OFF);  
digitalWrite(segF, SEGMENT_ON);
```

```
digitalWrite(segG, SEGMENT_ON);  
break;
```

case 6:

```
digitalWrite(segA, SEGMENT_ON);  
digitalWrite(segB, SEGMENT_OFF);  
digitalWrite(segC, SEGMENT_ON);  
digitalWrite(segD, SEGMENT_ON);  
digitalWrite(segE, SEGMENT_ON);  
digitalWrite(segF, SEGMENT_ON);  
digitalWrite(segG, SEGMENT_ON);  
break;
```

case 7:

```
digitalWrite(segA, SEGMENT_ON);  
digitalWrite(segB, SEGMENT_ON);  
digitalWrite(segC, SEGMENT_ON);  
digitalWrite(segD, SEGMENT_OFF);  
digitalWrite(segE, SEGMENT_OFF);  
digitalWrite(segF, SEGMENT_OFF);  
digitalWrite(segG, SEGMENT_OFF);  
break;
```

case 8:

```
digitalWrite(segA, SEGMENT_ON);  
digitalWrite(segB, SEGMENT_ON);  
digitalWrite(segC, SEGMENT_ON);  
digitalWrite(segD, SEGMENT_ON);  
digitalWrite(segE, SEGMENT_ON);
```

```
digitalWrite(segF, SEGMENT_ON);  
digitalWrite(segG, SEGMENT_ON);  
break;
```

case 9:

```
digitalWrite(segA, SEGMENT_ON);  
digitalWrite(segB, SEGMENT_ON);  
digitalWrite(segC, SEGMENT_ON);  
digitalWrite(segD, SEGMENT_ON);  
digitalWrite(segE, SEGMENT_OFF);  
digitalWrite(segF, SEGMENT_ON);  
digitalWrite(segG, SEGMENT_ON);  
break;
```

case 10:

```
digitalWrite(segA, SEGMENT_OFF);  
digitalWrite(segB, SEGMENT_OFF);  
digitalWrite(segC, SEGMENT_OFF);  
digitalWrite(segD, SEGMENT_OFF);  
digitalWrite(segE, SEGMENT_OFF);  
digitalWrite(segF, SEGMENT_OFF);  
digitalWrite(segG, SEGMENT_OFF);  
break;
```

```
}
```

```
}
```

```
void SwitchDigit(int digit) {
```

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

```
        if (i == digit) {
```

```
            digitalWrite(digit_pin[i], DIGIT_ON);
```

```

    } else {
        digitalWrite(digit_pin[i], DIGIT_OFF);
    }
}
}

```

```

struct struct_digits IntToDigits(int n){

```

```

    struct struct_digits dig;
    int zeros=0;
    int d;
    for (int i=0; i<4; i++) {
        d=n/pow(10,3-i);
        zeros += d;
        n = n - d*pow(10,3-i);
        if (zeros!=0 || i==3) {
            dig.digit[i]=d;
        } else {
            dig.digit[i]=10;
        }
    }
    return dig;
}

```

```

void PrintNumber(int n, int time) {

```

```

    struct struct_digits dig;
    dig = IntToDigits(n);
    for (int i=0; i<= time/20; i++) {
        if (digitalRead(button2)==LOW) {
            return;
        }
        for (int j=0; j<4; j++) {

```

```

        SwitchDigit(j);

        lightNumber(dig.digit[j]);

        delay(5);
    }
}
}

bool Countdown(int n, int del){
    for (int q=n; q>0; q--){
        PrintNumber(q,del);

        if (digitalRead(button2)==LOW) {
            return false;
        }
    }

    PrintNumber(0,0);

    playTone(1519,1000);

    return true;
}

void reset() {

    int m, zeros, d, pressed3 = 0, pressed4 = 0;

    m=countdown_time;

    struct struct_digits dig;

    dig = IntToDigits(countdown_time);

    while (digitalRead(button1)==HIGH) {

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

            SwitchDigit(j);

            lightNumber(dig.digit[j]);

            delay(5);

        }
    }
}

```



```

if (digitalRead(button3)==LOW) {
    if (pressed3 == 0 || pressed3 > 30) {
        if (countdown_time > 0) {
            countdown_time -= 1 ;
        }
        dig = IntToDigits(countdown_time);
    }
    pressed3 += 1;
}
else if (digitalRead(button4)==LOW) {
    if (pressed4 == 0 || pressed4 > 30) {
        if (countdown_time < 9999) {
            countdown_time += 1 ;
        }
        dig = IntToDigits(countdown_time);
    }
    pressed4 += 1;
}
if (digitalRead(button3)==HIGH) {
    pressed3=0;
}
if (digitalRead(button4)==HIGH) {
    pressed4=0;
}
}

void loop(){
    reset();
    while (!Countdown(countdown_time,962)) {

```

```
    reset();  
}  
while (digitalRead(button2)==1){};  
}
```

پایان.