

دستور کار ۴: کار با تایمر صفر میکروکنترلر

در این آزمایش هدف راه اندازی تایمر میکروکنترلر AVR می باشد. با استفاده از تایمرهای AVR می توان تأخیرهای دقیق را تولید نمود. یکی از مزایای استفاده از تایمر در میکروکنترلرها این است که با استفاده از تایمرها یک سری عملیات در زمانهای مشخص می تواند بدون اینکه میکرو درگیر انجام آن شود، صورت پذیرد. تایمر صفر در میکروکنترلرهای AVR یک شمارنده ۸ بیتی می باشد که قابلیت بالا شمار و پایین شمار داشته و در ۴ مود کاری مختلف می تواند عمل نماید. این تایمر دارای دو پرچم وقفه بوده که در صورت فعال بودن وقفه ها، با ۱ شدن پرچم وقفه زیر برنامه وقفه به طور خودکار اجرا خواهد شد. در این آزمایش هدف استفاده از این تایمر در مودهای کاری مختلف می باشد.

تایمر صفر دارای سه رجیستر ۸ بیتی TCNT0، OCR0 و TCCR0 می باشد. برای تنظیمات مودهای کاری تایمر از رجیستر TCCR0 استفاده می شود. رجیستر TCNT0 مقدار لحظه ای تایمر را دارا بوده و با هر بار شمارش عدد داخل این رجیستر تغییر می کند. رجیستر OCR0 برای مقایسه با مقدار TCNT0 استفاده شده و در زمانی که مقدار این دو رجیستر برابر باشد یا به عبارتی نتیجه مقایسه صحیح باشد در این صورت پرچم مربوط به وقفه مقایسه ۱ خواهد شد. در صورتیکه مقدار شمارش تایمر نیز به مقدار نهایی رسیده باشد، پرچم مربوط به وقفه سرریز برابر ۱ خواهد شد.

۱-۲-۱ تایمر صفر در مود Fast PWM

همانگونه که از نام این مود مشخص است، تایمر صفر در این مود برای تولید پالس PWM مورد استفاده قرار می گیرد. پالسهای PWM یکی از موارد بسیار پر کاربرد در صنعت به ویژه صنایع الکترونیک قدرت می باشند که کنترل دور موتورهای مختلف نمونه بارزی از آن می باشد. در این مود تایمر صفر بر روی پایه OC0 که همان پین

شماره ۳ پورت B می‌باشد یک موج PWM تولید خواهد کرد. فرکانس خروجی تایمر براساس رابطه زیر مشخص می‌شود:

$$F_{out} = \frac{F_{Clk}}{N \times 256} \quad \text{معادله (۱-۰)}$$

در رابطه فوق F_{Clk} فرکانس میکروکنترلر بوده و N ضریب Prescaler تایمر صفر می‌باشد. برای تعیین مقدار عرض پالس نیز از رجیستر OCR0 استفاده می‌شود.

یک پروژه جدید در نرم‌افزار ایجاد نموده و برنامه زیر را به آن اضافه نمایید. در برنامه زیر تایمر صفر در مود Fast PWM و حالت Non-Inverting استفاده شده است. با تنظیم رجیسترهای مربوطه، یک پالس مربعی با عرض پالس 50% و فرکانس 500Hz در خروجی OC0 تولید خواهد شد. برنامه را اجرا نموده و پس از پیاده سازی بر روی بردبرد پروگرام نمایید. سپس پالس خروجی را بر روی اسیلوسکوپ مشاهده نمایید.

```
#include <mega32.h>
void main(){
    TCCR0=0x6A;//Timer0 in Non-Inverting FAST PWM mode, Prescaler 8
    OCR0=0x7F;
    DDRB.3=1;// PORTB.3 as Output
    while(1){
        }
    }
```

۱-۲-۲ تایمر صفر در مود Phase Correct PWM

در این مود نیز همانند مود Fast PWM، تایمر صفر برای تولید پالس PWM مورد استفاده قرار می‌گیرد. در این مود تایمر هم به صورت بالاشمار و هم به صورت پایین‌شمار عمل کرده و عرض پالس موج خروجی قابلیت تنظیم بیشتری خواهد داشت. فرکانس خروجی تایمر براساس رابطه زیر مشخص می‌شود:

$$F_{out} = \frac{F_{Clk}}{N \times 510} \quad \text{معادله (۲-۰)}$$

در رابطه فوق F_{Clk} فرکانس میکروکنترلر بوده و N ضریب Prescaler تایمر صفر می‌باشد. برای تعیین مقدار عرض پالس نیز از رجیستر OCR0 استفاده می‌شود.

یک پروژه جدید در نرم افزار ایجاد نموده و برنامه زیر را به آن اضافه نمایید. در برنامه زیر تایمر صفر در مود Phase Correct PWM و حالت Non-Inverting استفاده شده است. با تنظیم رجیسترهای مربوطه، یک پالس مربعی با عرض پالس 50% در خروجی OC0 تولید خواهد شد. برنامه را اجرا نموده و بر روی برد آزمایشگاه پروگرام نمایید. سپس پالس خروجی را بر روی اسیلوسکوپ مشاهده نموده و فرکانس خروجی را از روی پالس خروجی تعیین نمایید.

```
#include <mega32.h>
void main(){
    TCCR0=0x62;//Timer0 in Non-Inverting Phase Correct PWM mode, Prescaler 8
    OCR0=0x7F;
    DDRB.3=1;// PORTB.3 as Output
    while(1){
        }
    }
```

۱-۲-۳ فرکانس متر با استفاده از تایمر صفر

در مود نرمال، تایمر صفر می تواند به عنوان شمارنده کلاک استفاده شود. شمارش تعداد کلاک در زمان مشخص، یکی از روشهای اندازه گیری فرکانس می باشد. یک پروژه جدید ایجاد نموده و برنامه زیر را به پروژه اضافه نمایید.

```
#include <mega32.h>
#include <alcd.h>
#include <delay.h>
#include <stdio.h>
char Buf[16];
unsigned char Counter=0;
unsigned int Data;
interrupt [TIM0_OVF] void timer0_ovf_isr(void) {
    Counter++;}
void main(){
    TCCR0=0x07;//Timer0 in Normal mode, External Clock
    TIMSK=0x01;
    DDRB.3=1;// PORTB.3 as Output
    lcd_init(16);
    lcd_clear();
    lcd_puts("Start");
    #asm("sei")
    while(1){
        TCCR0=0x00;
        TCNT0=0x00;
        Counter = 0;
        TIFR |= 0x01;
        TCCR0=0x07;
        delay_ms(1000);
        TCCR0=0x00;
        Data = (Counter * 256) + TCNT0;
        sprintf(Buf,"Freq = %d",Data);
```

```
lcd_clear();  
lcd_puts(Buf);  
}}
```

در برنامه فوق تایمر صفر در مود نرمال قرار داده شده و کلاک تایمر از پایه T0 دریافت می‌شود. وقفه تایمر صفر فعال شده و با سرریز شدن تایمر به تعداد شمارش یک واحد افزوده می‌شود. تایمر صفر به مدت یک ثانیه فعال بوده و در انتهای یک ثانیه، تایمر صفر خاموش شده و با استفاده از مقدار تایمر صفر و تعداد دفعات سرریز تایمر، فرکانس پالس ورودی محاسبه می‌شود.

تمرین ۱- با استفاده از تایمر صفر یک LED که به یکی از پین‌های میکرو متصل است را با فاصله زمانی ۱ ثانیه روشن و خاموش کنید.