

یه مثال از حالت تایمر

مثال اول) برنامه ای بنویسید که توسط تایمر صفر يك موج مربعی با چرخه کاری ۵۰٪ روی پایه portb.0 (میکرو AT90S8515) ایجاد نماید (دنوسنساز داخلی 4Mhz در نظر بگیرید).

```
1 #include<90S8515.h>
2 void delay
3 void main
4 {
5   DDRB=0x01 ;
6   PORTB=0x00;
7   TCCR0=0x00;
8   TCNT0=0x00;
9   while(1){
10    PORTB.0 = ~PORTB.0
11    dealy();
12  }
13 }
14 //-----
15 void dealy(){
16   TCCR0=0x02;
17   while(!(TIFR & 0x02 ));
18   TIFR1 =0x02
19 }
```

توضیحات برنامه:

خط ۱ معرفی میکرو
خط ۲ تعریف تابع تاخیر
خط ۳ تعریف تابع اصلی
خط ۵ پورت B به عنوان خروجی تعریف میشود
خط ۶ پورت B صفر می شود.
خط ۷ تایمر صفر را غیر فعال می کنیم.
خط ۸ تایمر با مقدار صفر پر میشود.
خط ۹ حلقه while يك بار اجرا میشود.
خط ۱۰ پورت b تغییر وضعیت میدهد.
خط ۱۱ تابع تاخیر فراخوانده میشود.

دستورات زیر برنامه:

خط ۱۵ معرفی تابع dealy

خط ۱۶ مقسم فرکانس را روی عدد تقسیم بر هشت تنظیم میکند. (فرکانس تایمر از تقسیم فرکانس میکرو بر هشت حاصل می شود).

خط ۱۷ حلقه شرطی while که تازمانی که شرط ان بر قرار باشد در حلقه خود باقی میماند ، همچنین گفتیم که پرچم تایمر را برای پاک کردن با عدد يك پر میکنیم یعنی $TIFR=0 \times 02$ پس تا زمانی پرچم فعال نشده در حلقه میمانیم.

با ایجاد وقفه شرط حلقه باطل و از ان خارج می شویم.

خط ۱۸ بیت پرچم را پاک می کند . حال از تابع تاخیر dealy خارج شده و تابع اصلی خط ۴ برمیگردیم و پورت B معکوس می شود.

اگر فرکانس میکرو ۴ مگا هرتز باشد با تنظیم مقسم روی عدد ۸ فرکانس تایمر ۵۰۰ کیلو هرتز میشود. یعنی هر شمارش در ۲ میکرو ثانیه انجام میشود تایمر صفر ۲۵۶ و اخر می شمارد (۲۵۵ پالس تا پر شدن و يك پالس برای سرریز شدن ان و برگشتن به عدد صفر) پس هر تغییر وضعیت از صفر به يك و يك به صفر ۵۱۲ میکرو ثانیه طول میکشد ($2us \times 256 = 512us$)

اینجا به مثال از حالت مقایسه هم زدم تا باین حالت هم آشنا بشی

مثال ۲) برنامه ای بنویسید که هر 1ms پایه OC0 یعنی portb.4 در میکرو ATmega6 را مکمل کند. (به کمک عملکرد مقایسه تایمر صفر) (فرکانس میکرو 8Mhz)

```
1 #include <mega64>
2 void main(){
3   DDRB=0x10;
4   TCNT0=0x00;
5   OCR0=0x7C;
6   TCCR=0x1C;
7   //start timer 0
8   while(1)
9   }
```

توضیح برنامه:

گفتیم که اگر بخواهیم از پایه خروجی OC0 استفاده کنیم در اولین گام باید به عنوان خروجی تعریف شود این کار در خط ۳ انجام شده.

خط ۴ TCNT0 را برای شروع شمارش با عدد صفر پر میکند.

خط ۵ رجیستر مقایسه خروجی با عدد ۱۲۴ یا همان 7CH

خط ۶ علاوه بر تنظیم فرکانس تایمر بر روی ۱۲۵ کیلو هرتز (مقسم فرکانس روی تقسیم ۶۴ تنظیم

شده $8\text{Mhz} \div 64 = 125\text{Khz}$) ، تنظیم تایمر بر روی حالت مقایسه و پایه OC0 را به صورت مکمل پیکر بندی میکند

در مقاله بالا اشاره شده که بیت های CS00، CS01 و CS02 را اگر به صورت (۱۰۰) پیکربندی کنیم مقسم بر روی

عدد ۶۴ تنظیم میشود. بیت های WGM00 و WGM01 نوع عمل تایمر را تنظیم میکنند که يك تایمر ساده باشد یا مولد

موج PWM یا يك واحد مقایسه که با تنظیم آن بر روی 10 به حالت مقایسه میرود. بیت های COM00 و COM01 نیز

نوع تغییر وضعیت پایه OC0 را تنظیم میکنند که با تنظیم آن بر روی ۰۱ به حالت مکمل میرود یعنی با هر بار برابر

شدن TCNT0 با OCR0 این پایه از يك به صفر و از صفر به يك میرود.

در آخر چون این بیت ها در کنار هم و در رجیستر TCCR0 در کنار هم قرار دارند طبق فرم زیر عدد مورد نیاز ی را

که باید در رجیستر TCCR0 تشکیل می دهند.

Foc0	Wgm00	Com01	Com00	Wgm01	Cs02	Cs01	Cs00
------	-------	-------	-------	-------	------	------	------

00011100b=1CH