دانشکده
فنی و حرفه ای امام حسین (ع)
اصفهان

بسم الله الرحمن الرحیم

میکرو کامپیوتر۱

جلسه۴

مدرس : محمدجواد شاهسوندی

کار با LCD در نرم افزار بسکام

معرفی LCDکاراکتری

LCDهای موجود در بازار عموماً در مدلهای

(بر اساس تعداد کاراکتر)

۲*۱۶ – ۴*۱۶ – ۴*۲۰ – ۲*۲۰ – ۱*۱۶ هستند.

هر LCDدارای ۱۶ پایه است

که در جدول روبرو شرح داده شده اند:

شماره پایه	نام پایه	وظیفه پایه
1	GND	زمین مدار الکتریکی (0 ولت)
2	Vcc	تغذیه +5 ولت
3	VEE	تنظیم مقدار نور صفحه LCD
4	RS	بیت پیغام بین LCD و میکروکنترلر برای ارسال داده (*Register Select*)
5	R/W	مشخص می کنیم که LCD باید بنویسد (اتصال به +5 ولت) یا بخواند (اتصال به زمین الکتریکی)
6	E	بیت پیغام که آمادگی LCD را برای دریافت دیتا اعلام می کند (*Enable*)
7	DB0	پین دریافت دیتا
8	DB1	پین دریافت دیتا
9	DB2	پین دریافت دیتا
10	DB3	پین دریافت دیتا
11	DB4	پین دریافت دیتا (از این پایه استفاده می کنیم)
12	DB5	پین دریافت دیتا (از این پایه استفاده می کنیم)
13	DB6	پین دریافت دیتا (از این پایه استفاده می کنیم)
14	DB7	پین دریافت دیتا (از این پایه استفاده می کنیم)
15	+LED	تغذیه +5 ولت تغذیه LED روشنایی صفحه LCD
16	-LED	تغذیه 0 ولت تغذیه LED روشنایی صفحه LCD

پیکربندی LCD کاراکتری در نرم افزار بسکام Bascom

معرفی LCD به نرم افزار بسکام

برای معرفی LCDبه نرم افزار بسکام به شیوه زیر عمل می کنیم :

CONFIG LCDPIN = PIN , DB4 = PN , DB5 = PN , DB6 = PN , DB7 = PN , E = PN , RS = PN

PN:پایه دلخواه از میکروکنترلر که به پایه LCDمتصل می شود. به طور مثال : PORTB.7

مثال :

CONFIG LCDPIN = PIN , DB4 = PORTB.4 , DB5 = PORTB.5 , DB6 = PORTB.6 , DB7 =-

PORTB.7 , E = PORTC.0 , RS = PORTC.1

باید دقت شود که پیکربندی پایه های LCDباید در یک خط و یا با قرار دادن (-) در انتهای خط در خط بعدی ادامه داشته باشد.

پیکربندی LCD کاراکتری در نرم افزار بسکام Bascom

تعیین نوع LCD

برای معرفی نوع LCDبه نرم افزار بسکام نیز به شیوه زیر عمل می کنیم:

CONFIG LCD = LCDtype

که LCDtype نوع LCD از لحاظ تعداد کاراکتر در سطر و ستون است که انواع آن در بالا اشاره شد.

مثال : Config lcd = LCD 20*4

که در اینجا ما از یک LCD با ۲۰ ستون و ۴ سطر استفاده کردیم.

دستورات و توابع مربوط به LCDکاراکتری :

دستور : LCD

برای نمایش اطلاعات بر روی LCDاز این دستور استفاده می کنیم. و به شکل زیر نوشته می شود:

LCD data

Dataاطلاعاتی است که قصد نمایش آن را داریم و ممکن است یک عدد یا حرف باشد و یک متغییر (var) یا یک ثابت (constant)باشد.

پیکربندی LCD کاراکتری در نرم افزار بسکام Bascom

برای نمایش چند عبارت پشت سر هم از علامت (;) بین عبارات استفاده می کنیم.

مثال:

```
LCD  a ; b1 ; "constant«
```

دستور CLS

این دستور برای پاک کردن تمام صفحه LCD (Clear Screen)استفاده می شود. و در طول برنامه فقط کافی است عبارت CLSبه تنهایی در یک خط به کار رود.

پیکربندی LCD کاراکتری در نرم افزار بسکام Bascom

دستور HOME

از این دستور برای قرار دادن مکان نما (CURSORدر اولین ستون سطرهای LCDاستفاده می شود و به شکل زیر نوشته می شود:

HOME UPPER/LOWER/THIRD/FOURTH

که به طور اختصار به صورت زیر نیز نوشته می شود:

HOME U/L/T/F

که به ترتیب مکان نما در ابتدای سطرهای اول ، دوم ، سوم و چهارم قرار می گیرد.

مثال :

```
Home lower
Lcd "ahmad"
Home u
Lcd "father"
```

که در مثال بالا عبارت ahmadدر ابتدای سطر دوم و عبارت father در ابتدای سطر اول به نمایش درمیاید.

پیکربندی LCD کاراکتری در نرم افزار بسکام Bascom

دستور LOCATE

از این دستور برای انتقال مکان نما به محل دلخواه استفاده می شود. و به شکل زیر نوشته می شود:

```
LOCATE  X , Y
```

X : مشخص کننده شماره سطر

Y : مشخص کننده شماره ستون

مثال :

```
Lcd "ahmad"
Locate 2*13
Lcd "father"
End
```

در مثال بالا عبارت ahmad در ابتدای سطر اول و عبارت father از ستون ۱۳ در سطر دوم شروع می شود.

پیکربندی LCD کاراکتری در نرم افزار بسکام Bascom

دستور SHIFTLCD

از این دستور در مواقعی که نیاز است تا اطلاعات نوشته شده در LCD به سمت چپ یا راست حرکت کنند (مانند تابلو روان) استفاده می شود. و به صورت زیر نوشته می شود:

SHITLCD LEFT/RIGHT

مثال:

```
Lcd "ahmad father"
Shiftlcd left
Wait 1
Shiftlcd right
Wait 2
End
```

در مثال بالا عبارت ahmad father ابتدا به مدت ۱ ثانیه به سمت چپ حرکت می کند و بعد به مدت ۲ ثانیه به سمت راست حرکت می کند.

پیکربندی LCD کاراکتری در نرم افزار بسکام Bascom

تابع DEFLCDCHAR

با این دستور می توانیم حرف یا علامتی که در منوی TOOLSو قسمت LCD DESIGNERمحیط نرم افزار بسکام طراحی کرده ایم را بر روی LCDنمایش دهیم. بعد از طراحی حرف یا علامت دلخواه در LCD DESIGNERو کلیک کردن بر روی دکمه OKخط زیر در محیط برنامه نویسی ظاهر خواهد شد.

```
DEFLCDCHAR ?,r1,r2,r3,r4,r5,r6,r7,r8
```

R1تا r8با توجه به طراحی توسط نرم افزار نوشته می شوند و ما می توانیم به جای ? عددی بین صفر تا ۷ قرار دهیم. بدین صورت ما می توانیم تا ۸ کاراکتر را طراحی کنیم و بر روی LCDنمایش دهیم. نمایش کاراکتر طراحی شده توسط دستور LCD CHR (?)بعد از دستور CLSانجام می گیرد.

مثال :

```
Deflcdchar 0 , 32 , 32 , 17 , 17 , 31 , 32 , 4 , 32
Cls
Lcd chr (0)
End
```

برای نمایش کاراکتر "ب" از مثال بالا استفاده می شود.

نمایش اعداد بر روی 7segment

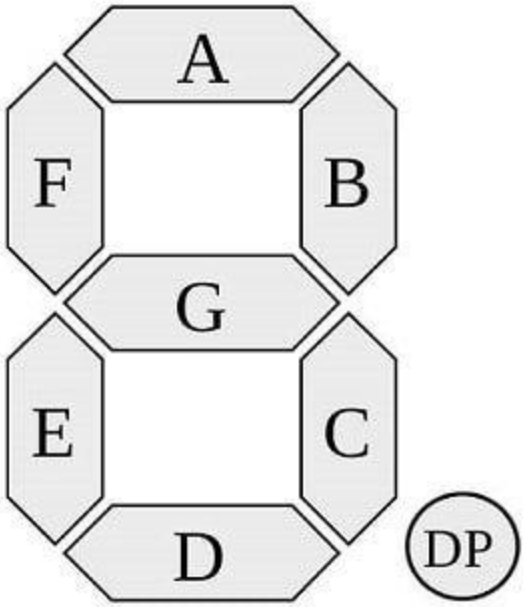
سون سگمنت seven segment

یکی از پرکاربرد ترین قطعات در الکترونیک میباشد که از ان برای نمایش اعداد و مقادیر استفاده می شود. این قطعه از ۸ عدد ledتشکیل شده که ۷ تای ان برای نمایش اعداد ۰ تا ۹ و برخی حروف و یک ledبرای نمایش نقطه (ممیز) استفاده می شود. هفت ledکه برای نمایش اعداد و برخی حروف استفاده می شوند,

مانند تصوی روبرو با حروف Aتا G

نامگذاری می شوند و

نقطه با DP مشخص می شود .



نمایش اعداد بر روی 7segment

یک LEDدارای دو پایه به نام های اند و کاتد می باشد, که برای روشن شدن led باید پایه ی آند به مثبت و کاتد به منفی وصل شود. بر اساس نحوه اتصال این ledها به هم,۲نوع سون سگمنت وجود دارد.

آند مشترک

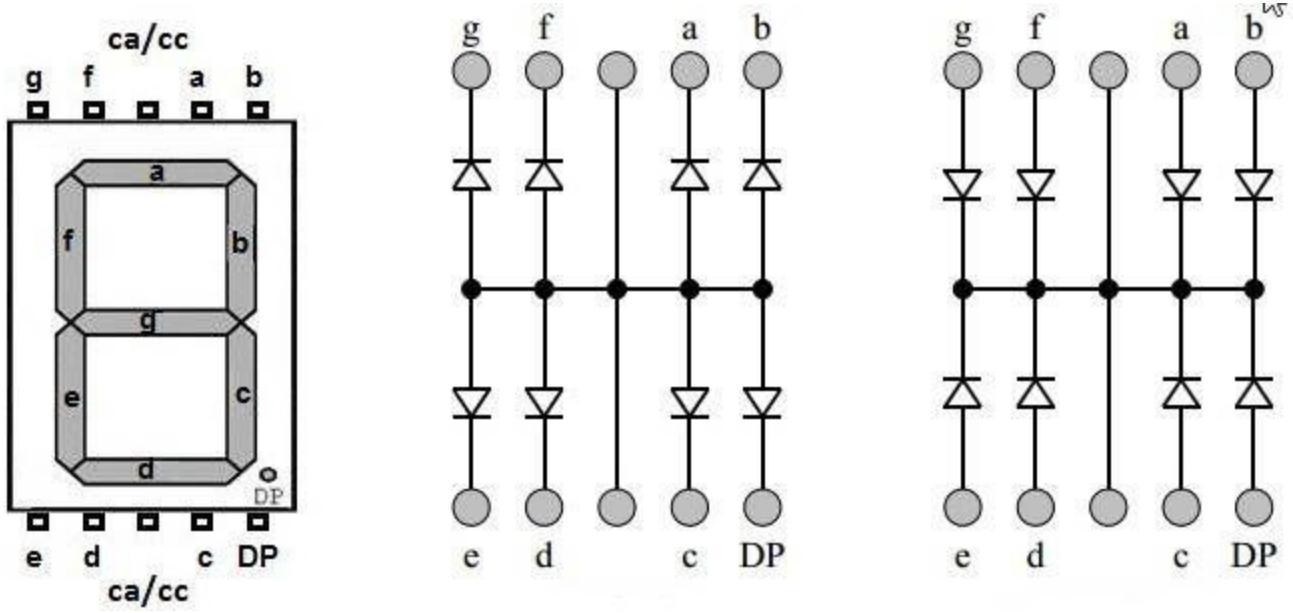
در صورتی که پایه مشترک ledها پایه آند باشد به آن سون سگمنت آند مشترک می گویند, که پایه مشترک باید به مثبت وصل شود و برای روشن شدن هر کدام از ledهای Aتا Gو DPپایه ledمورد نظر باید به منفی وصل شود

کاتد مشترک

در صورتی که پایه مشترک led ها پایه کاتد باشد به آن سون سگمنت کاتد مشترک می گویند, که پایه مشترک باید به منفی وصل شود و برای روشن شدن هر کدام از ledهای Aتا Gو DPپایه led مورد نظر باید به مثبت وصل شود.

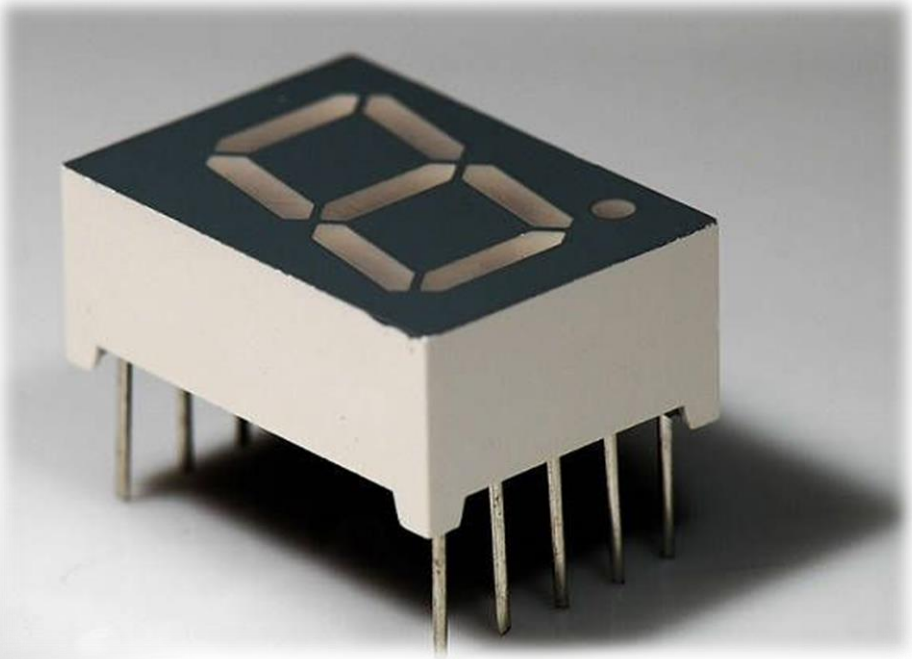
نمایش اعداد بر روی 7segment

در تصویر زیر نحوه اتصال **led**های این دو نوع سون سگمنت مشخص شده است .



نمایش اعداد بر روی 7segment

سون سگمنت ها به صورت تکی , دو تایی , سه تایی , چهار تایی وجود دارند که بسته به نیاز خود می توانید یکی از آنها را تهیه کنید. در تصویر بالا, عکس سمت چپ چیدمان پایه های یک سون سگمنت تکی را نشان می دهد که در هنگام اتصال به مدار باید به ترتیب قرار گیری پایه ها توجه کرد. در تصویر زیر یک سون سگمنت تکی را مشاهده می کنید که می توانید با توجه به تصویر بالا پایه های آن را تشخیص بدهید .



نمایش اعداد بر روی 7segment

```
$regfile = "m8def.dat"
$crystal = 1000000

Config PORTD = Output


Dim A As Byte

Do

PORTD = Lookup(a , 7segment_code)
Waitms 500
Incr A
If A > 15 Then A = 0

Loop

7segment_code:

DATA &B00111111
DATA &B00000110
DATA &B01011011
DATA &B01001111
DATA &B01100110
DATA &B01101101
DATA &B01111101
DATA &B00000111
DATA &B01111111
DATA &B01101111
DATA &B01110111
DATA &B01111100
DATA &B00111001
DATA &B01011110
DATA &B01111001
DATA &B01110001
```

پرسش و تحقیق

برنامه راه اندازی موتور DC وبررنامه ای ورود داده به صورت مستقیم و مبتنی بر بیت STROBE باشد را بنویسید .

تایمرها در میکروکنترلرAVR

یکی از اجزای داخلی میکروکنترلر AVRتایمر است. در میکروکنترلرهای AVR بازه ی زمانی تایمرها می تواند از چند میکروثانیه شروع شود و حتی تا چندین ساعت ادامه یابد. میکروکنترلرهای AVR دارای تایمرهایی با دقت بالا هستند تا جایی که در میکروثانیه ها هم دقت دارند و این موضوع آن ها را برای کاربردهای تایمر بسیار قابل اعتماد و کاربردی ساخته است.

برای درک بهتر تایمرها شاید نیاز باشد تا مثال هایی زده شود. ساده ترین مثال از یک تایمر همان ساعتی است که به دیوار خانه تان آویخته اید یا همان ساعتی است که دور مچ دست بسته اید. هرچیزی در این دنیا با زمان آمیخته است. از برنامه روزانه تان گرفته تا ساعات کاری تان. اما مفهوم تایمرها فقط به روتین روزانه محدود نمی شود. تک تک دستگاه ها و قطعات الکترونیکی بر اساس یک زمان پایه کار می کنند. این زمان پایه کمک می کند تا تمام قطعات با یکدیگر همزمان باشند. بدون زمان پایه هیچ قطعه ای نمی داند که چه موقع باید کار خاصی را انجام دهد .

تایمرها در میکروکنترلرAVR

پس تایمر مفهومی بسیار مهم در حوزه ی الکترونیک است. تولید زمان پایه می تواند از طریق یک مدار تایمر یا یک میکروکنترلر و ... انجام پذیرد. با توجه به این که تمامی میکروکنترلرها دارای فرکانس ساعت از پیش تعریف شده ای هستند، همگی آن ها دارای قوانینی برای تنظیم و راه اندازی تایمر می باشند. همچنین با توجه به دقیق و قابل اعتماد بودن تایمرهای AVR، ویژگی های زیادی را می توان برای این تایمرها برشمرد. و این موضوع خود باعث حجیم شدن مبحث تایمر می شود

تایمرها در میکروکنترلرAVR – تایمرها در نقش ثباتها

اگر بخواهیم به زبان ساده بگوییم یک تایمر در AVRعملا یک ثبات است. البته نه یک ثبات عادی. مقدار این ثبات به طور اتوماتیک کم یا زیاد می شود. در AVRتایمرها دو نوع هستند. ۰ بیتی و ۱۱ بیتی. در تایمر ۰ بیتی طول ثبات ۰ بیت است در حالی که در تایمر ۱۱ بیتی طول آن ۱۱ بیت می باشد. این ۱ به توان ۰(مرحله از صفر تا ۱۵۵ می باشد. (بدین معناست که تایمر ۰ بیتی قادر به شمردن ۱ ۱۵۱ به توان ۱۱ (از صفر تا ۱۵۵۳۵ است. به دلیل وجود این ویژگی است که تایمرها نحت عنوان کانتر (به طور مشابه تایمر ۱۱ بیتی قادر به شمارش ۱۵۵۳۱)شمارنده(نیز شناخته می شوند. حال سوالی که پیش می آید این است که هنگامی که تایمرها به مقدار ماکسیمم خود رسیدند چه اتفاقی می افتد؟ آیا برنامه ی شمارش از کار می افتد؟ جواب خیلی ساده است. تایمر پس از رسیدن به مقدار حداکثر خود به مقدار اولیه ی صفر بازگشته و شمارش را از نو شروع می کند. در صورت رخ دادن این اتفاق می گوییم که تایمر/کانتر سرریز کرده است

تایمرها در میکروکنترلرAVR – تایمرها در نقش ثباتها

در میکروکنترلر ATmega 32سه نوع تایمر مختلف داریم :

- تایمر ۰: یک تایمر ۸ بیتی است.
- تایمر ۱: که یک تایمر ۱۶ بیتی است.
- تایمر ۲: که یک تایمر ۸ بیتی است

بهترین ویژگی ای که تایمر دارا می باشد این است که نسبت به CPUمستقل است. از این رو کاملا موازی با CPUکار می کند و با CPUهیچ گونه تداخلی ندارد. همین امر است که تایمر را کاملا دقیق می سازد.

تایمرها در میکروکنترلر AVR – تایمرها در نقش ثباتها

جدا از عملکرد عادی تایمر، هر سه تایمری که معرفی شدند در سه مد کاری می توانند کار کنند :

مد نرمال،

مد CTC

مد PWM

در مورد این مد ها در مطالب آتی صحبت خواهیم کرد.

تایمرها در میکروکنترلرAVR – مفاهیم پایه ای تایمر

از ابتدای تحصیل همیشه با این فرمول سروکار داشته ایم:

$$\text{دوره‌ی زمانی} = \frac{۱}{\text{فرکانس}}$$

حال فرض کنید که نیاز داریم تا یک LED 18/را هر ۱۸ میلی ثانیه خاموش و روشن کنیم. این یعنی اینکه طبق فرمول بالا باید بنویسیم: ۱ . 100 Hz = msحال بیایید فرض کنید که یک کریستال خارجی به میکروکنترلر وصل است که فرکانس آن برابر با ۴ مگاهرتز باشد. بنابراین فرکانس ساعت CPUبرابر با ۴ مگاهرتز می باشد. همان طور که گفتیم تایمر ۸ از صفر تا ۱۵۵ و تایمر ۱ از صفر تا ۱۵۵۳۵ می شمارند و دوباره سرریز می شوند

تایمرها در میکروکنترلر AVR – مفاهیم پایه ای تایمر

مقدار این شمارش ها با هر پالس ساعت تغییر می کند. بیایید در نظر بگیریم که مقدار اولیه ی تایمرمان صفر باشد. هنگامی که یک پالس ساعت دریافت می کند مقدار آن ۱ می شود، هنگامی که پالس ساعت بعدی را دریافت می کند مقدار آن به ۱ تغییر می کند و به همین صورت با دریافت هر پالس ساعت یکی می شمارد. برای فرکانس ۴ مگاهرتز CPU 4/ هر دوره زمانی برابر است با: ۱ =M

0.00025 ms . بنابراین هر تغییر حالت فقط ۸،۸۸۸۱۵ میلی ثانیه طول می کشد. همان طور که در مثال بالا گفتیم نیاز داریم که یک تاخیر ۱۸ میلی ثانیه ای داشته باشیم. این تاخیر برای ما بسیار کوتاه است، اما برای میکروکنترلری که هر دوره ۸۸۸۱۵ میلی ثانیه است این گونه نیست.

تایمرها در میکروکنترلرAVR – مفاهیم پایه ای تایمر

برای این که محاسبه کنیم که هر با داشتن دوره زمانی پالس میکروکنترلر چه تعداد شمارش نیاز / ی

پالس آن ۸ است، از فرمول زیر استفاده می کنیم :

$$\text{تعداد شمارش تایمر} = \frac{\text{تأخیر مورد نیاز}}{\text{دوره‌ی پالس ساعت}} - ۱$$

تایمرها در میکروکنترلرAVR – مفاهیم پایه ای تایمر

با جایگذاری کردن مقدار تاخیر ۱۰ میلی ثانیه ای که می خواهیم، و همچنین جایگذاری دوره پالس میکروکنترلر که برابر با ۰/۰۰۰۲۵ است، تعدا شمارش مورد نیاز برای تایمر برابر با ۳۹۹۹۹ می شود. یعنی این که اگر فرکانس CPUبرابر با ۴ مگاهرتز باشد و ما نیاز به تولید تاخیر ۱۰ میلی ثانیه ای داشته باشیم، تایمر باید به تعداد ۳۹۹۹۹ بشمارد تا این تاخیر ایجاد شود. مسلم است که برای ایجاد چنین تاخیری نمی توانیم از تایمر ۸ بیتی استفاده کنیم، چرا که این تایمر تا بیشتر از ۲۵۵ نمی تواند بشمارد. بنابراین باید از تایمر ۱۶ بیتی استفاده نمود زیرا این تایمر قادر به شمردن تا مقدار حداکثر ۶۵۵۳۵ می باشد.

تایمرها در میکروکنترلر AVR – تقسیم کننده‌ی فرکانس

موضوعی که در این جا پیش می آید این است که اگر مقدار حداکثر ۶۵۵۳۵ را در فرمول داده شده جایگذاری کنیم به تاخیر ۱۶/۳۸۴ میلی ثانیه ای می رسیم. حال اگر نیاز به ایجاد تاخیر ۲۰ میلی ثانیه ای داشته باشیم باید چه کنیم؟؟؟

تایمرها در میکروکنترلر AVR – تقسیم کننده‌ی فرکانس

موضوعی که در این جا پیش می آید این است که اگر مقدار حداکثر ۶۵۵۳۵ را در فرمول داده شده جایگذاری کنیم به تاخیر ۱۶/۳۸۴ میلی ثانیه ای می رسیم. حال اگر نیاز به ایجاد تاخیر ۲۰ میلی ثانیه ای داشته باشیم باید چه کنیم؟؟؟

تایمرها در میکروکنترلر AVR – تقسیم کننده‌ی فرکانس

خوشبختانه راه حلی وجود دارد. فرض کنید که فرکانس CPU را از ۴ مگاهرتز به ۰/۵ مگاهرتز تقلیل دهیم. در این صورت دوره زمانی هر پالس برابر با ۱/۵۰۰ k = 0.002 ms می شود. حال اگر تاخیر مطلوب ۲۰ میلی ثانیه و دوره زمانی ۰/۰۰۲ میلی ثانیه را در فرمول جایگذاری کنیم به تعداد شمارش ۹۹۹۹ می رسیم و این تعداد شمارش با استفاده از تایمر ۱۶ بیتی قابل دست یابی است. با این فرکانس حداکثر زمان تاخیر قابل دست یابی برابر با حدود ۱۳۱ میلی ثانیه می شود. اما سوالی که در این جا پیش می آید این است که چگونه فرکانس را کاهش دهیم. در پاسخ باید گفت که این تکنیک کاهش فرکانس را تقسیم فرکانس می نامند. در این تکنیک در واقع فرکانس CPU کاهش نمی یابد. بلکه فرکانس واقعی CPU در همان مقدار قبلی باقی می ماند (در این مثال ۴ مگاهرتز). ولی فرکانسی را که تایمر با آن راه اندازی می شود، از فرکانس CPU به دست می آوریم.

تایمرها در میکروکنترلرAVR – تقسیم کننده‌ی فرکانس

برای این کار دستورالعملی برای تنظیم بعضی از بیت ها در AVRوجود دارد که بعدا در مورد آن ها صحبت خواهیم کرد. اما فکر نکنید امکان این وجود دارد که آزادانه از تقسیم کننده ی فرکانس استفاده نمود. بلکه استفاده از مقسم فرکانس هزینه هایی هم در بر دارد. همیشه بین دوره ی زمانی پالس و مدت زمان تاخیر مصالحه ای برقرار است. همان طور که مشاهده کردید در مثال بالا مدت زمان تاخیر از ۱۶ میلی ثانیه به ۱۳۱ میلی ثانیه افزایش یافت. همچنین دوره ی پالس نیز از ۰/۰۰۰۲۵ ثانیه به ۰/۰۰۲ رسید. خب مشکل کجاست؟ مشکل اینجاست که دقت کاهش می یابد. قبلا قادر بودیم دوره ی زمانی ای مانند ۰/۱۱۲۵ را با دقت اندازه بگیریم چون می توانستیم این دوره را با تعداد دقیق ۴۵۰ شمارش (۰،۱۱۲۵/۰،۰۰۰۰۲۵ = ۴۵۰) تولید کنیم ولی اکنون این کار ممکن نیست چون: ۰،۱۱۲۵/۰،۰۰۰۲ = ۵۶،۲۵ تعداد صحیحی از شمارش را به ما نمی دهد. به عبارت دیگر با استفاده از مقسم فرکانس در این مثال می توانیم تاخیر زمانی ۰،۱۱۲ را تولید کنیم و تاخیر بعدی قابل تولید ۰،۱۱۴ است و هیچ تاخیری بین این دو مقدار قابل تولید نیست.

تایمرها در میکروکنترلرAVR – انتخاب تقسیم کننده در تایمر

با یک مثال بحث را پی می گیریم. فرض کنیم که نیاز به تأخیر ۱۸۴ میلی ثانیه ای نیاز داریم در حالی که فرکانس CPUبرابر با ۴ مگاهرتز باشد. مقسم های فرکانسی که AVRدر اختیار ما می گذارد تا یکی را انتخاب کنیم عبارتند از: ۸ ، ۶۴، ۲۵۶ و ۱۰۲۴ . مقسم فرکانس ۸ بدین معنی است که فرکانس موثر پالس تایمر برابر با فرکانس CPUتقسیم بر ۸ می باشد (.(F_CPU/8حال با جایگذاری این مقادیر در فرمولی که گفته شد، مقادیر مختلفی از مقدارهای تایمر به دست می آید. این نتایج در جدول زیر به طور خلاصه ارائه شده اند:

شمارش تایمر	فرکانس ساعت	مقسم
91999	500 kHz	8
11499	62.5 kHz	64
2874	15.625 kHz	256
717.75	3906.25 Hz	1024

تایمرها در میکروکنترلر AVR – انتخاب تقسیم کننده در تایمر

همان طور که مشاهده می شود از بین این ۴ مقسم فرکانس، مقسم ۸ قابل استفاده نیست چون مقدار تایمر با استفاده از مقسم فرکانس ۸ از حد مجاز که ۶۵۵۳۵ می باشد، بالاتر می زند. همچنین به این دلیل که تایمر همیشه مقادیر صحیح اختیار می کند، برای رسیدن به تاخیر دقیق ۱۸۴ میلی ثانیه ای نمی توان از مقسم های ۱۰۲۴ استفاده کرد. حال از بین مقسم های ۶۴ و ۲۵۶ بهتر است که از مقسم ۶۴ استفاده شود چرا که دوره های زمانی کوچک تری از پالس را به ما می دهد و دقتش بالاتر خواهد بود. البته اگر نیاز باشد که در جایی دیگر از تاخیرهای با مدت بیش از ۱۸۴ میلی ثانیه استفاده شود، مقسم ۲۵۶ را بر می‌گزینیم.

تایمرها در میکروکنترلرAVR – انتخاب تقسیم کننده در تایمر

به طور خلاصه: از بین مقسم ها مقسمی را انتخاب می کنیم که مقداری ممکن برای تایمر به ما بدهد (در تایمر ۸بیتی عدد کمتر از ۲۵۵ و در تایمر ۱۶ بیتی عددی کمتر از ۶۵۵۳۵) و مقدار شمارنده همیشه باید مقداری صحیح باشد.

در مورد چگونگی انتخاب و تنظیم مقسم ها در انواع تایمر میکروکنترلر AVRدر مطالب آتی صحبت خواهد شد.

تایمرها در میکروکنترلر AVR – وقفه‌های تایمرمیکروکنترلر AVR

مبحث وقفه ها به طور انحصاری مربوط به تایمرها نیست، ولی با توجه به این که مبحث وقفه ها در جاهای مختلفی استفاده می شود، نیاز است که در این جا نیز در مورد آن صحبت شود.