

به نام خدا

عنوان مقاله:

آموزش میکرو AVR به زبان بسیار ساده

قسمت نهم

نویسنده:

احمد فهیمی

micro-avr@noavar.com



قسمت نهم

1-سلام خوبی

2- خوبم تو چطوری

1- ای بد نیستم خوب قراره امروز چی یادم بدی

2- امروز می خوام کار با حافظه EEPROM رو بهت بگم

1- چی هستش

2- حافظه بلند مدت یا حافظه پایدار میکروکنترلر . اگه یادت باشه برنامه ای رو که ما می نوشتیم و بعد پروگرام می کردیم و با این کار برنامه رو می ریختیم توی قسمت حافظه . FLASH ما در AVR می توانیم برنامه ای رو که نوشتیم یا توی حافظه FLASH بریزم و یا توی حافظه EEPROM بریزیم بستگی داره که دوست داشته باشیم اطلاعات ذخیره بشه یا نه

1- من که نفهمیدم چی شد بالاخره توی کدوم حافظه بریزیم بهتره

2- ببین این دوتا حافظه نسبت به هم یک سری خوبی ها یی دارن و یک سری بدی ها مثلا حافظه FLASH نسبت به حافظه EEPROM از سرعت بالاتری برخورداره و ما برنامه ای رو که قبلا توی این حافظه ریخته بودیم می تونیم از طریق نرم افزار پاکش کنیم بدی حافظه FLASH هم توی همین هستش که اطلاعاتی رو که توی اون ریختیم بعد از این که تغذیه میکروکنترلر قطع بشه از بین می ره

1- منظورت از اطلاعات همون برنامه ای هستش که توش ریختیم

2- نه نه اشتباه نشه منظورم اینه که مثلا شما داری یک متغیر رو افزایش می دی و روی ال سی دی هم نمایش می دی بعد تغذیه میکروکنترلر رو بر می داری بعد دوباره تغذیه رو وصل می کنی ایا مقدار متغیری که قبلا افزایش داده بودی از صفر می یاد یا آخرین عددی که افزایش پیدا کرده بوده خوب مصلما از صفر چون اطلاعاتش پاک شده. خوب بریم سر بحث خودمون حافظه EEPROM هم نسبت به حافظه FLASH یک سری مزیت داره اون هم اینه که می تونی اطلاعات رو در خودش نگه داره یک بدی هم که داره اینه که سرعتش نسبت به حافظه FLASH پایین هستش و برنامه ای رو که شما توی این حافظه می ریزی نمی تونی با نرم افزار پاکش کنی و باید با اشعه پاک بشه . بزار یک مثال بزنم . فرض کن که داری توی برنامه WORD یک متن رو تایپ می کنی و کلی هم تایپ کردی و ناگهان برق

می ره و بعد از دو ساعت برق میاد می ری کامپیوتر رو روشن می کنی ایا اون متنی رو که تایپ کرده بودی هنوز هستش یا اینکه از بین رفته

1- خوب مسلما چون که ذخیره نکرده بودم از بین رفته

- 2- افرین . حالا بیا فرض کنیم که شما بعد از این که این متن رو تایپ کردی رفتی توی یک درایوی ذخیره کردی و بعد از این که ذخیره کردی یک دفعه برق می ره بعد از دو ساعت برق می ره می ری کامپیوترت رو روشن می کنی بعد می ری سراغ اون متنی که قبلا ذخیره کرده بودی ایا پاک شده
- 1- نه چون که ذخیره کرده بودم
- 2- احسن حافظه EEPROM هم همینطور و می تونه آخرین اطلاعاتی رو که بهش دادی تا مدت های نا محدود در خودش نگه داره
- 1- بابا یک مثال هم از نحوه برنامه نویسیش بزن
- 2- چشم این هم یک مثال

```

Dim A As Byte
Readeeprom A , 0
Do
    Debounce Pinb.0 , 1 , Show , Sub
    Debounce Pinb.1 , 1 , Show2 , Sub

Loop
End

show:
    Incr A
    Writeeprom A , 0
    Waitms 5
    Readeeprom A , 0
    Cls
    Lcd A
    Return
show2:
    A = 0
    Writeeprom A , 0
    WAITMS 5
    Cls
    Lcd A
    Return

```

در برنامه بالا ما به وسیله دو میکروسویچ که به پایه های b.0 و b.1 وصل هستند یک متغیر رو زیاد می کنیم و هر وقت که دوست داشتیم صفرش می کنیم میکروسویچی که به پایه B.0 وصل هستش برای زیاد کردن متغیر هستش و به کمک این میکروسویچ می تونیم از صفر تا 225 رو به کمک میکروسویچ زیاد کنیم میکروسویچی که به پایه B.1 وصل می شه برای این هستش که ما در هر جایی که از برنامه باشیم بتوانیم متغیر رو صفر کنیم

جالبی این برنامه این هستش که ما به دلیل این که برنامه رو توی حافظه EEPROM میریزیم اگه یک وقتی تغذیه میکروکنترلر رو قطع کنیم و مجددا وصل کنیم و دوباره میکروسویچی که برای زیاد کردن متغیر هستش رو تحریک کنیم متغیر از آخرین عددی که قبلا زیاد شده بوده شروع می کنه به زیاد شدن

1- همین تیکه رو یک بار دیگه بگو

2- فرض کن که با میکرو سوییچ متغیر رو زیاد کردی و رسوندی به عدد مثلا 10 و روی ال سی دی نمایش دادی بعد تغذیه میکروکنترلر رو قطع کردی و بعد از چند ساعت میای دوباره تغذیه میکروکنترلر رو وصل می کنی و دوباره میکروسویچ رو می زنی حتما فکر میکنی که از صفر دوباره شروع میکنه به زیاد کردن . باید بگم اگه اینطوری فکر می کنی اشتباه کردی چون که بازدن میکروسویچ به دلیل اینکه این مقدار متغیر در حافظه EEPROM ذخیره شده متغیر شروع میکنه به زیاد شدن البته از ادامه کار یعنی ادامه 10 . خوب حالا شاید خواستیم که از صفر شروع کنیم برای این کار میکروسویچی رو که به پایه B.1 وصل هستش رو فشار می دیم و دوباره مقدار متغیر می شه 0

خوب حالا بریم سر تحلیل برنامه:

میکرو میاد برنامه رو از خط اول شروع می کنه به خوندن

خط اول ما یک متغیر از نوع بایت و به اسم A تعریف کردیم

خط دوم نوشتیم `Readeeprom A , 0` یه این معنی که بخوان مقدار متغیر A رو از مکان حافظه 0 EEPROM که به جای 0 هر عدد دیگری میتونی بدی . میکرو که این دستور رو میخونه می فهمه که باید مقدار متغیر A رو که قبلا ذخیره کرده بخونه خوب برای بار اول که ما چیزی رو ذخیره نکردیم . پس مقدرا متغیر A رو در وحله اول صفر در نظر می گیره

بعد میاد میوفته توی حلقه `DO - LOOP` و اون تو دور می زنه تا وقتی که یکی از پایه های B.0 یا B.1 ما یک بشه که این پایه ها مستقیما وصل شده به میکروسویچ ها

فرض کن که میکروسویچ اول که وصل شده به پایه B.0 تحریک بشه در این جا میکروکنترلر میفهمه که پایه B.0 یک شده و می ره دستور شرط رو که پرش به زیر برنامه SHOW هستش رو اجرا می کنه و میپره میره داخل این زیر برنامه خوب حالا باید دستورات داخل این زیربرنامه رو اجرا کنه

اولین دستور داخل این زیربرنامه هستش INC R A و میکرو هم این کار رو میکنه و یک واحد به مقدار متغیر A اضافه می کنه یعنی می شه عدد 1

دستور دوم هستش Writeeprom A , 0 این دستور به این معنی هستش که مقدار متغیر A رو ذخیره کن داخل مکان حافظه 0 و میکرو هم این کار رو میکنه و مقدار عدد 1 رو توی این مکان از حافظه EEPROM ذخیره میکنه . دستور بعدی داخل این زیربرنامه نوشته Waitms 5 یعنی این که یک تاخیر 5 میلی ثانیه ای ایجاد کن این تاخیر رو همیشه بعد از دستور Writeeprom بزار چون که یک تاخیر ایجاد بشه و عملیات ذخیره سازی به درستی انجام بشه . دستور بعدی داخل این زیربرنامه نوشته Readeeprom A , 0 این دستور به این معنی هستش که بخوان مکان حافظه 0 از حافظه EEPROM دقیقه همان مکانی که مقدار متغیر A رو اونجا ذخیره کرده بودیم و میکرو هم این مکان رو می خونه و می فهمه که مقدرا متغیر A هستش 1 . دستور بعدی داخل این زیربرنامه هستش CLS که باید بدونی برای چی هستش . خط بعدی داخل این زیربرنامه نوشته LCD به این معنی که مقداری رو که توسط دستور Readeeprom A , 0 خواندی رو روی ال سی دی نمایش بده و میکرو هم این کار رو می کنه توجه داشته باش که هر وقت خواستی مقدار متغییری رو که قبلا ذخیره کردی رو بخونی و روی ال سی دی نمایش بدی اول باید با دستور Readeeprom مقدار ذخیره شده رو بخونی بعد با دستور LCD مقدرا رو روی ال سی دی نمایش بدی . دستور بعدی داخل این زیربرنامه هستش RETURN به این معنی که بپر برو سر همون خطی که قبلا ازش پرش کرده بودی یعنی خط Show , Sub , 1 , 0 , Pinb و دوباره توی این حلقه دور می زنه تا این که دوباره یکی از دو میکروسوییچ ها تحریک بشه و بپره بره توی زیربرنامه ای که براشون تعریف شده . فرض کن که همیجا یک دفعه دستت میخوره به سیم منبع تغذیه ای که به میکروکنترلر وصل هستش و تغذیه میکروکنترلر قطع میشه و تو دوباره تغذیه رو وصل می کنی و میکروکنترلر هم از خط اول برنامه شروع میکنه به خواندن برنامه و دستور READEEPROM A , 0 رو می خونه و میکروکنترلر با خواندن این دستور می فهمه که باید بره مقدار متغیر A رو از مکان 0 حافظه EEPROM بخونه یعنی دقیقا همانجایی که مقدار متغیر A رو در اونجا ذخیره کرده بود . خوب ما آخرین مقداری رو که با تحریک کردن میکروسوییچ به متغیر A دادیم عدد 1 بود خوب اینجا میکرو می فهمه که ما آخرین مقداری رو که به متغیر A داده بودیم عدد 1 بوده از این به دیگه از صفر شروع نمی کنه به افزایش متغیر بلکه از ادامه عدد 1 شروع میکنه به شمردن . خوب ما یک میکروسوییچ دیگه هم داشتیم که وصل بود به پایه B.1 و این میکرو سوییچ برای این بود که مقدار متغیر ما در هر وضعیتی که بود بیاد صفرش کنه به این ترتیب که با تحریک میکروسوییچ میکروکنترلر می فهمه که پایه B.1 یک شده و باید از داخل حلقه DO - LOOP بپره

بره توی زیربرنامه SHOW2 و دستورات داخل این زیربرنامه رو اجرا کنه خوب حالا داخل این زیربرنامه چی هستش الان می گم اولین دستوری که داخل زیربرنامه SHOW2 هستش $A = 0$ هستش که میکروکنترلر با خوندن این دستور میاد مقدار متغیر A رو صفر می کنه حالا مقدار متغیر A هر چقدر که بوده فرقی نداره و صفرش می کنه . دستور بعدی داخل این زیربرنامه هستش WAITMS 5 که میکروکنترلر با خوندن این دستور یک تاخیر 5 میلی ثانیه ای در اجرای برنامه ایجاد می کنه که این تاخیر رو همونطور که گفتم باید بعد از دستور Writeeprom همیشه بزاری تا اطلاعاتی که می خواد ذخیره بشه به درستی ذخیره بشه . دستور بعدی داخل این زیربرنامه هست CLS که می دونی برای چی هستش . دستور بعدی هستش LCD A و میکرومیاد مقدار متغیر A رو که ما الان صفرش کردیم رو روی ال سی دی نمایش می ده یعنی عدد صفر رو نمایش می ده . دستور بعدی داخل این زیربرنامه هستش RETURN که میکرو با خوندن این دستور میاد میپره کجا همونجایی که قبلا ازش پریده داخل این زیربرنامه یعنی خط Sub , Show2 , 1 , 1 , Debounce Pinb و دوباره می یوفته داخل این حلقه DO - LOOP و روز از نو روزی از نو

- 1- یک سوال اگه ما مقدار متغیر رو با زدن میکروسوییچ رسوندیم به مثلاً عدد 20 بعد توسط میکروسوییچ دوم اومدیم مقدار میکروسوییچ رو صفر کردیم بعد تغذیه میکروکنترلر رو قطع کردیم و دوباره وصل کردیم حالا سوال اینجاست که آخرین عددی که در حافظه EEPROM ذخیره می شه و بعد از برقراری تغذیه مجدداً توسط دستور READEEPROM خونده می شه چه عددی هستش ؟
- 2- خوب این که خیلی راحت شما بعد از این که مقدار متغیر رو رسوندی به عدد 20 اومدی بازدن میکروسوییچ دوم مقدار متغیر رو صفر کردی پس آخرین مقداری که در این متغیر بوده و ذخیره هم شده عدد صفر هستش و با برقراری تغذیه مجدد و خوانده شدن دستور READEEPROM توسط میکروکنترلر میکروکنترلر میفهمه که مقدار متغیر هم اکنون هستش صفر . اینو یادت باشه که آخرین مقداری که به متغیر می دی توی حافظه EEPROM ذخیره میشه و فرقی نداره که چه عددی باشه.