

به نام خدا

عنوان مقاله:

آموزش میکرو AVR به زبان بسیار ساده

قسمت هشتم

نویسنده:

احمد فهیمی

micro-avr@noavar.com



قسمت هشتم

1- سلام خوبی

2- سلام خوبم تو چطوری

1- عالی راستی امروز چی می خای بگی

2- امروز باید دستور حلقه و پرش رو که قبلا بهت نگفتم بهت بگم که خیلی راحت هست

1- آماده ام بگو

2- اولین دستور GOSUB هستش توسط این دستور شما می تونی بپری توی یک زیر برنامه و دستورات

داخل اون زیر برنامه رو انجام بدی دقیقا همان کاری که دستور GOTO برای ما انجام می داد این دستور

GOSUB هم برای ما انجام می ده با یک تفاوت و اون هم اینکه در دستور GOSUB ما با دستور

RETURN می توانیم دوباره برگردیم GOSUB رو بخونیم کاری که با دستور GOTO نمی تونستیم انجام

بدیم

1- یک مثال بزن که نفهمیدم

2- باشه اینم مثال:

CONFIG PINB.0 = INPUT

CONFIG PORTA = OUTPUT

DO

IF PINB.0 = 1 THEN GOSUB BENZ

LOOP

END

BENZ:

TOGGLE PORTA

WAITMS 50

RETURN

توسط برنامه بالا می تونیم با زدن میکرو سویچ پورت A رو TOGGLE کنیم. یعنی برای بار اول که

میکرو سویچ رو میزنیم پورت A یک می شه برای بار دوم که میزنی صفر میشه برای بار سوم که می زنی

دوباره یک می شه برای بار چهارم که میزنی صفر میشه و الی آخر. در خط اول ما پایه B.0 رو به عنوان

ورودی پیکره بندی کردیم . در خط دوم پورت A رو برای خروجی پیکره بندی کردیم . در خط سوم ما اول حلقه رو با نوشتن دستور DO عنوان کردیم. در خط چهارم ما نوشتیم

`IF PINB.0 = 1 THEN GOSUB BENZ` به معنی که اگر پایه B.0 میکروکنترلر یک شد پیر برو توی زیربرنامه . BENZ در خط پنجم با دستور LOOP انتهای حلقه رو مشخص کردیم . در خط ششم نوشتیم

END که برنامه ما هیچ وقت به اینجا نمی رسه . در خط بعدی ما یک زیربرنامه به اسم BENZ مشخص کردیم . خط هفتم نوشتیم `TOGGLE PORTA` یعنی وضعیت پورت A رو هر چی که هست معکوسش کن یعنی اگه یک هستش بکنش صفر و اگه صفر هستش بکنش یک . در خط هشتم ما یک تاخر 50 میل ثانیه ای گذاشتیم . در خط نهم نوشتیم `RETURN` که در ادامه بهت که توی این برنامه ضیفش چیه .

خوب میکروکنترلر میاد از خط اول برنامه شروع میکنه به خوندن خط اول خط دوم رو می خونه . به خط سوم که می رسه یعنی DO می فهمه که یک حلقه سر راهشه و توی این حلقه یکسره دور میزنه و خارج نمی شه تا وقتی که اون شرطی که گذاشتیم اجرا بشه یعنی `IF PINB.0 = 1 THEN GOSUB BENZ` حالا فرض می کنیم که ما میکروسویچ رو فشار دادیم و با این کارمون PINB.0 رو یک کردیم در اینجا میکرو میاد دستور شرط رو یعنی `THEN GOSUB BENZ` رو اجرا میکنه یعنی میپره می ره توی زیربرنامه BENZ و دستورات داخل این زیر برنامه رو اجرا میکنه .خوب حالا داخل این زیر برنامه ما این دستورات رو نوشتیم `TOGGLE PORTA` یعنی وضعیت پورت A رو هر چی که هستش معکوسش کن و در اینجا میکرو میاد پورت A رو یک میکنه و اگه شما به پورت A وصل کرده باشی می بینی که روشن می شه دستور بعدی که داخل این زیربرنامه هست `WAITMS 50` که میکرو با خوندن این خط به مدت 50 میلی ثانیه در اجرای برنامه تاخیر ایجاد میکنه . دستور بعدی که ما داخل این زیربرنامه نوشتیم `RETURN` هستش که میکرو بعد از گذشت 50 میلی ثانیه که این دستور رو می خونه برمی گرده توی حلقه . و دوباره توی اون حلقه دور می زنه تا وقتی که دوباره شرط اجرا بشه و دوباره پیره بره توی زیربرنامه .

برنامه بالا رو اینطوری هم میشه نوشت که این روش منطقی تره:

3-

`CONFIG PINB.0 = INPUT`
`CONFIG PORTA = OUTPUT`
`DO`

```

DEBOUNCE PINB.0 , 1 , BENZ , SUB
LOOP
END
BENZ:
TOGGLE PORTA
RETURN

```

برنامه بالا رو فقط تاخیر 50 میلی ثانیه ای شو حذف کردم . برنامه بالا رو تنها تغییری که توش انجام دادم خط چهارمش هستش یعنی DEBOUNCE PINB.0 , 1 , BENZ , SUB که ما با نوشتن این خط این رو به میکرو فهمونیدم که اگر یک موقع PINB.0 ما یک شد بپر برو توی زیربرنامه . BENZ اون SUB هم نوشتیم که به میکرو بفهمونیم که ما دستور GOSUB می خایم که این پرش صورت بگیره و اگه ننویسیم میکرو خودش پیش فرض با دستور GOTO میپره توی زیربرنامه . خوب حالا فرض کن که این شرط یعنی پایه B.0 یک شده در اینجا میکرو میپره میره توی زیربرنامه BENZ و دستورات داخل این زیربرنامه رو اجرا می کنه و وقتی که به دستور RETURN میرسه مفهمه که باید بپره بره توی حلقه و خط

```

DEBOUNCE PINB.0 , 1 , BENZ , SUB

```

رو دوباره بخونه و داخل این حلقه دوباره گیر میکنه تا وقتی که پایه B.0 دوباره یک بشه و دوباره بپره بره توی زیربرنامه و روز از نو روزی از نو.

دستور بعدی حلقه FOR –NEXT هستش که با این دستور می تونی یک حلقه شمارشی ایجاد کنی اگه توجه کنی تا حالا من یک حلقه DO-LOOP بهت گفتم که یک حلقه نامحدود بود ولی این حلقه-FOR NEXT یک حلقه محدود هستش یعنی میاد توی یک حلقه ایجاد میکنه البته تایک زمان مشخصی این کار رو میکنه که این زمان دست خودمون هستش که چقدر باشه به مثال زیر توجه کن:

```

DIM A AS BYTE
FOR A = 1 TO 10 STEP 1
CLS
LCD A
WAITMS 30
NEXT A
END

```

خوب بریم سر تحلیل برنامه . خط اول که ما یک متغیر به اسم A و از نوع بایت تعریف کردیم . در خط

دوم ما نوشتیم که

FOR A = 1 TO 10 STEP 2 به این معنی که متغیر A رو افزایش بده از عدد 1 تا عدد 10 . و

دستور STEP 1 که نوشتیم به این معنی هستش که یکی یکی بشمار یا به عبارتی یک پله یک پله بشمار

1- بین نفهمیدم این STEP 1 رو دوباره توضیح بده

2- یعنی این که یکی یکی بشمار 1 و 2 و 3 و 4 و 5 و تا عدد 10 . اگه ما به جای STEP 1 می دادیم

STEP 2 دو تا دوتا می‌شمرد اینطوری 1 و 3 و 5 و 7 و 9

خوب بریم سر ادامه وقتی که میکرو خط FOR A = 1 TO 10 STEP 2 رو می خونه شروع می کنه

به افزایش متغیر . A بعد میاد خط بعدی که نوشته CLS رو می خونه و در این جا LCD رو پاک میکنه .

بعد میاد خط LCD A رو میخونه و می فهمه که باید مقداری که داره افزایش پیدا میکنه رو نشون بده . خط

بعدی نوشتیم WAITMS 30 که یک تاخیر 30 میلی ثانیه ای در اجرای برنامه انجام میده و کارش این

هستش که بیاد مدت زمان افزایش رو تند یا کند کنه یا به عبارتی آهسته یا سریع کنه . خط بعدی نوشتیم

NEXT A که میکرو با خوندن این خط دوباره برمیگرده و خط دوم رو یعنی خط

FOR A = 1 TO 10 STEP 2 رو می خونه . خط بعدی نوشتیم END که برنامه ما فعلا به این

قسمت نمی رسه تا وقتی که شمارش تموم بشه

1- من که چیزی نفهمیدم

2- بزار تحلیلش کنم می فهمی

خوب حالا تحلیل برنامه : میکرو میاد از خط اول شروع میکنه به خوندن و می فهمه که ما یک متغیر به

اسم a و از نوع بایت تعریف کردیم . در خط دوم که ما نوشتیم FOR A = 1 TO 10 STEP 1

میکرو میاد شروع میکنه به شمارش یا همون افزایش متغیر یک نکته رو بزار همیجا بگم که فرقی نداره که

عددها چقدر باشه حتی می تونیم بگیم که از 10 شروع به شمارش کن تا مثال 110 هیچی محدودیتی

وجود نداره البته حداکثر تا 255 میتونی بدی . خوب بری سر دستوری که نوشتیم . خوب بعد میکرو میاد

دفعه اول خود مقدار 1 رو میزازه توی متغیر A بعد میاد با فرمان CLS ال سی دی رو پاک میکنه و با دستور

LCD A میاد مقدار متغیر A رو که الان هست 1 رو روی ال سی دی نمایش می ده بعد بادستور

WAITMS 30 میاد یک تاخیر 30 میلی ثانیه ای در اجرای برنامه ایجاد می کنه و در خط بعد تا که

دستور NEXT A رو می خونه برمی گرده توی اون خط FOR A = 1 TO 10 STEP 1 و دوباره

که این دستور رو بخوند مقدار متغیر A رو یک واحد بیشتر می کنه یعنی 2 و دوباره با دستور CLS ال سی دی رو پاک می کنه و با دستور LCD A میاد مقدار متغیر A رو که الان هست 2 رو روی ال سی دی نمایش می ده . این روال همینطور ادامه داره تا این که مقدار متغیر A بشه 10 یا به عبارتی مساوی بشه با مقداری که در پایان شمارش بهش دادیم (که در اینجا عدد 10 رو دادیم) و وقتی که این اتفاق افتاد میکرو میاد خط زیری NEXT A رو می خونه که ما در اینجا دستور END رو گذاشتیم که برنامه به پایان برسه ولی شما اگه حال کردی که شمارش دوباره آغاز بشه و دوباره بشماره می تونی قبل از دستور FOR یک زیربرنامه تعریف کنی و در بعد دستور NEXT A هم یک دستور پرش به این زیربرنامه رو بزاری که هر وقت که شمارش به پایان رسید میکرو پره بره توی این زیربرنامه و شمارش رو از دوباره آغاز کنه مثل مثال زیر

```
DIM A AS BYTE
MAIN:
FOR A = 1 TO 10 STEP 1
CLS
LCD A
WAITMS 30
NEXT A
GOTO MAIN
END
```

دستور بعدی WHILE – WEND هستش این دستور همون کاری رو می کنه که دستور DO – LOOP انجام می داد ولی با یک تفاوت اون هم اینکه برای داخل شدن به این حلقه بی نهایت باید شرط مورد نظری که جلوی WHILE نوشته میشه اول اجرا بشه تا اجازه داخل شدن به این حلقه داده بشه . یک چیز دیگه هم که هست اینه که WHILE ابتدای حلقه هستش مثل همان DO که ابتدای حلقه بود و WEND هم انتهای حلقه مثل LOOP که انتهای حلقه بود به مثال زیر توجه کن که خیلی هم ساده هست.

- 1- یک سوال فرق بین دستور WHILE – WEND با دستور DO – LOOP چی هستش
- 2- در دستور DO – LOOP شما برای ورود به حلقه نیاز به اجازه نداشتی ولی توی دستور WHILE WEND نیاز به کسب اجازه داری حالا این اجازه کی صادر میشه زمانی که شرطی رو که جلوی WHILE گذاشتیم اجرا بشه . اگه بخای به یک زبان دیگه حالتی کنم باید بگم که فرق این دوتا دستور مثل ابتدایی و

دانشگاه هستش شما توی ابتدایی برای بیرون رفتن از کلاس باید از معلمتون اجازه می گرفتی ولی توی دانشگاه اجازه نمی گیری و سرتو میندازی پایین و می ری بیرون و اگه یک زمانی توی دانشگاه از استاد اجازه بگیری بقیه بهت می خندن توی میکرو هم همینطور ممکنه که متغیرا بهت بخندن. خوب بری یک مثال ساده هم برات بزنیم که حال کنی:

DIM K AS BYTE

K = 3

WHILE K = 3

CLS

LCD "OK"

WEND

END

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

خط دوم به متغیر K مقدار دادیم . مقدار 3 دادیم

خط سوم با دستور WHILE ابتدای حلقه رو مشخص کردیم البته برای ورود به حلقه این حلقه یک شرط گذاشتیم که این شرط هر چیزی میتونه باشه که این شرط رو ما باید جلوی دستور WHILE با یکم فاصله بنویسیم . شرطی که نوشتیم $K = 3$ هستش یعنی اگر که مقدار K عدد 3 بود اجازه ورود به حلقه داده میشه.

خط چهارم که داخل حلقه قرار داره CLS هستش که می دونی چی کار می کنه

خط پنجم هم LCD "OK" هستش که این دستور هم داخل حلقه قرار داره

خط ششم WEND هستش یعنی پایان حلقه

خط هفتم هم END هستش که تا زمانی که شرط برقرار باشه برنامه به این دستور نمی رسه

تحلیل این برنامه هم خیل ساده هستش میکرو میاد از خط اول شروع می کنه به خوندن خط اول خط دوم رو می خونه می رسه به خط سوم این جا رو گوش کن که خیلی مهمه اگه شرط برقرار بود یعنی مقدار K 3= بود که الان هست اجازه ورود به این حلقه داده می شه ولی اگه شرط برقرار نبود میکرو میره و خط بعدی WEND رو می خونه که ما اینجا حال کردیم که اگه شرط برقرار نبود برنامه به END برسه و تموم بشه.

دستور بعدی DELAY هستش که این دستور یک دستور تاخیر در اجرای برنامه هستش همانند دستور های

WAITMS – WAIT – WAITUS که دستور DELAY به مدت 1000 میکروثانیه در اجرای برنامه تاخیر

ایجاد میکنه که نیاز هم به مثال نیست.

فعلا همینا رو داشته باش تا بعد