

به نام خدا

عنوان مقاله:

آموزش میکرو AVR به زبان بسیار ساده

قسمت چهارم

نویسنده:

احمد فهیمی

micro-avr@noavar.com



قسمت چهارم

1- سلام خوبی

2- سلام قربانت تو چطوری چکار می کنی با این میکروکنترلر

1- منم خوبم . با این میکروکنترلر داریم حال می کنیم والا قبلا برای طراحی یک مدار ساده دیجیتال

مجبور بودم چند روز روش کار کنم ولی حالا دیگه نه . بعدشم یک چیزو یادت رفته بود بگی

2- چی؟؟

1- از TOGGLE مثال نزدی

2- باشه توی این درسی رو که امروز بهت میدم از این دستور هم استفاده می کنم که یاد بگیری . خوب

شروع کنیم

1- اره

2- خوب قرار بود از کجا شروع کنیم

1- قرار بود ورودی رو بهم بگی . گفته بود قبل از اینکه ورودی رو درس بدی باید چند تا دستور شرط

و ... رو برام بگی

2- اره این جلسه قبل از اینکه ورودی یا همون PIN رو بهت بگم باید . دستور حلقه - دستور پرش رو

اول بگم بعد بریم سر ورودی . چون که این طوری بهتره

1- حرفی نیست بریم

2- خوب اول بریم سراغ حلقه . ببین به یک چیز توی برنامه های قبلی توجه کرده بودی

1- نه چی؟؟

2- ما میومدیم یک برنامه رو می نوشتیم این برنامه هم یک کاری رو برامون انجام میداد و در نهایت برنامه

به END می رسید و برنامه کلا تمام می شد . و برای شروع مجدد مجبور بودیم برنامه رو دوباره اجرا کنیم

. خوب حالا فرض کن ما یک برنامه ای رو میخایم که مثلا PORTB.0 رو هر 40 میلی ثانیه یک بار

روشن و خاموش کنه و این کار ادامه داشته باشه یعنی اینکه 40 میلی ثانیه روشن باشه و 40 میلی ثانیه

خاموش باشه . برای این کار باید از دستور استفاده کنیم که برنامه هیچ وقت به پایان نرسه که این دستور

اسمش هست حلقه

1- خوب حالا این حلقه چکار میکنه

2- این حلقه باعث میشه برنامه توی این قسمت دور بزنه و هیچ وقت به END نرسه یعنی برنامه تمام نشه مگر اینکه ما یک شرط بزاریم که از حلقه بیرون بزاریم

1- چه شرطی

2- اونو تو قسمت دستور شرط برات می گم

حلقه DO - LOOP

برای گذاشتن حلقه DO رو اول می نویسیم و در خط های بعدی برنامه ای که باید در حلقه باشد رو می نویسیم و در نهایت LOOP رو می نویسیم

1- من که چیزی نفهمیدم . یک مثال بزن

همون مثالی رو که PORTB.0 باید SET و RESET بشه رو برات می نویسم . از این به بعد هم دیگه

```
$REGFILE = "M16DEF.DAT"
```

```
$CRYSTAL = 8000000
```

خوب حالا برنامه ای که گفتم:

```
CONFIG PORTA = OUTPUT
```

```
DO
```

```
TOGGLE PORTA.0
```

```
WAITMS 40
```

```
LOOP
```

```
END
```

خوب حالا تحلیل برنامه:

این برنامه هیچ گاه به END نمی رسد تا برنامه تمام شود

خط اول رو که PORTA رو به عنوان خروجی پیکره بندی کردیم

خط دوم DO را گذاشتیم

خط سوم با نوشتن TOGGLE PORTA.0 این را رساندیم که PORTA.0 در هر حالتی از حالت ها

صفر یا یک که بوده باید وضعیتش معکوس شود یعنی اگه صفر بوده حالا باید یک شود . اگر یک بوده

حالا باید صفر شود

خط چهارم هم آخر حلقه را گذاشتیم

خط پنجم هم نوشتیم END البته برنامه هیچ گاه به خاطر این که ما یک حلقه گذاشتیم به END نمی رسد توجه کن که ما برنامه ای رو که می خاستیم رو داخل این حلقه گذاشتیم و برنامه داخل این حلقه دور میزند . بار اول برنامه TOGGLE PORTA.0 رو می خونه و پایه A.0 رو SET یا همون یک می کنه بعد 40 میلی ثانیه صبر می کنه و برنامه از خط بعد شروع می شه خط بعد LOOP هستش که میکرو با خوندن LOOP دوباره بر میگردد برنامه رو از خط پایینی DO شروع به خوندن می کنه یعنی دوباره همون TOGGLE PORTA.0 رو می خونه این بار چون پایه A.0 در حالت SET یا همون یک بوده این بار یاد بشه عکس همین حالت بشه یعنی RESET یا صفر میشه بعد 40 میلی ثانیه صبر میکنه و میکرو برنامه رو از خط بعدی میخونه یعنی دستور LOOP رو می خونه . با خوانده شدن دستور LOOP میکرو برنامه رو از خط زیری DO شروع به خوندن می کنه یعنی دوباره TOGGLE PORTA.0 رو می خونه این بار چون در حالت قبلی پایه A.0 در حالت صفر بوده این بار یک می شه برنامه به همین منوال ادامه داره و ادامه پیدا می کنه و هیچ وقت به پایان نمی رسه

1- گرفتم چی می گی ولی اگه ممکنه یک مثال دیگه هم بزن

2- باشه

مثال دوم:

```
CONFIG PORTB = OUTPUT
DO
SET PORTB.5
WAITMS 60
RESET PORTB.5
WAITMS 60
LOOP
END
```

خوب حالا بریم سر تحلیل برنامه:

برنامه از خط اول شروع میشه و میکرو خط اول رو می خونه

خط دوم که DO هستش که ابتدای حلقه ما هستش

خط سوم SET PORTB.5 یعنی پایه B.5 یک بشه

خط چهارم یک تاخیر در برنامه با مدت زمان 60 میلی ثانیه گذاشتم

خط پنجم RESET PORTB.5 یعنی پایه B.5 صفر بشه

خط ششم با گذاشتن دستور LOOP انتهای حلقه رو مشخص کردیم

خط هفتم هم نوشتیم END که برنامه به خاطر داشتن حلقه هیچ گاه به END نمی رسد

برنامه به این صورت است که از خط اول شروع به خوانده شدن توسط میکرو میشه و پیکره بندی رو می

خونه . بعد میوفته توی حلقه و اولین عبارتی رو که می خونه SET PORTB.5 هستش که در این زمان

میکرو پایه B.5 رو یک می کنه بعد توسط دستور WAITMS 60 حدود 60 میلی ثانیه تاخیر ایجاد می

کنه بعد از سپری شدن این مدت زمان میکرو برنامه رو از خط بعدی شروع به خواندن می کنه یعنی

RESET PORTB.5 که میکرو توسط این دستور پایه B.5 رو صفر می کنه بعد توسط دستور WAITMS

60 میکرو 60 میلی ثانیه در اجرای برنامه دوباره تاخیر ایجاد می کنه بعد از سپری شدن این مدت زمان

میکرو برنامه رو از خط بعدی یعنی LOOP شروع به خواندن می کنه که با خوانده شدن LOOP برنامه می

پره از خط بعدی DO یعنی SET PORTA.0 شروع میشه و دوباره پایه B.5 یک میشه و دوباره 60 میلی

ثانیه تاخیر ایجاد میشه دوباره پایه B.5 صفر میشه و 60 میلی ثانیه تاخیر ایجاد میشه بعد دوباره به

LOOP می رسه و دوباره میکرو میپره و برنامه رو از خط بعدی DO شروع به خواندن میکنه و الی بی

نهایت این برنامه تکرار می شه

1- عالی بود کامل گرفتم

2- خدا رو شکر. جهنم ضرر یک مثال دیگه هم میزنم که LCD هم توش باشه

```
DO
CLS
LCD "AHMAD"
WAITMS 50
CLS
LCD "MICRO"
WAITMS 60
LOOP
END
```

خوب حالا نوبت می رسه به تحلیل برنامه:

برنامه از DO شروع میشه DO یعنی ابتدای حلقه . در خط بعد با دستور CLS ال سی دی پاک می شه و

آماده نوشتن . در خط بعدی به میکرو دستور می دیم که عبارت AHMAD رو روی LCD نمایش بده . در

خط بعد فرمان ایجاد تاخیر به مدت زمان 50 میلی ثانیه را می دهیم پس از سپری شدن این مدت زمان میکرو برنامه را از خط بعد می خواند . خط بعدی با دستور CLS صفحه LCD را پاک کردیم تا برای نوشتن دوباره آماده شود. در خط بعد فرمان دادیم که عبارت MICRO روی صفحه LCD نمایش داده شود. در خط بعد دستور دادیم که 60 میلی ثانیه تاخیر ایجاد شود که پس از سپری شدن این تاخیر برنامه از خط بعد اجرا می شود. به خط بعدی یعنی LOOP که می رسم میکرو تا اینکه این دستور LOOP رو می بیند بی درنگ می پره میره برنامه رو از خط بعدی DO شروع می کنه به اجرا کردن . یعنی از خط CLS و همینطور ادامه می ده

2- رو حساب رفاقت می خام یک مثال دیگه هم بزنم که حالشو بیری

DIM G AS BYTE

G = 0

DO

INCR G

CLS

LCD G

WAITMS 50

LOOP

END

خط اول رو که خودت می دونی یعنی اینکه ما یک متغیر داریم که اسمش هست G و از نوع بایت

خط دوم به متغیر G مقدار دادیم

خط سوم ابتدای حلقه را مشخص کردیم

خط چهارم یک واحد به متغیر G اضافه کردیم

خط پنجم با دستور CLS ال سی دی را برای نوشتن پاک کردیم

خط ششم دستور دادیم که مقدار متغیر G روی LCD نمایش داده شود

خط هفتم یک تاخیر 50 میلی ثانیه ای در برنامه ایجاد کردیم

خط هشتم با دستور LOOP انتهای حلقه را مشخص کردیم

خوب حالا تحلیل اصلی رو گوش کن:

میکرو میاد برنامه رو از خط اول شروع می کنه به خوندن و می فهمه که یک متغیر از نوع بایت ایجاد

شده . بعد خط دوم رو می خونه که نوشته مقدار این متغیر هست صفر . خط سوم ابتدای حلقه مشخص

شده است. خط چهارم با دستور INCR G یک واحد به متغیر G اضافه کردیم یعنی اگر بوده صفر حالا که یک واحد بهش اضافه شده میشه 1. خط بعدی با دستور CLS ال سی دی را پاک کردیم تا برای نوشتن حاضر شود. خط بعدی دستور دادیم که مقدار متغیر G که الان هست 1 روی LCD نمایش داده شود. خط بعدی با دستور WAITMS 50 دستور تاخیر به مدت 50 میلی ثانیه را صادر کردیم. خط بعدی هست LOOP که میکرو وقتی که به این دستور می رسه می پره میره و برنامه رو از خط زیری DO شروع می کنه به خواندن توجه کن که از خط بعدی DO شروع می کنه به خواندن نه اینکه از بالا. داریم به جای جالبش می رسم. خوب حالا باید برنامه از خط بعدی DO خوانده بشه خط بعد از DO نوشته چی نوشته INCR G خوب یعنی اینکه یک واحد به متغیر G اضافه بشه حالا آخرین مقدار متغیر G بوده 1 (به دلیل اینکه دفعه قبل با دستور INCR G یک واحد به ان اضافه کرده بودیم) و حالا که دوباره دستور دادیم که یک واحد دیگر به متغیر G اضافه بشه اینبار میشه 2. خط بعدی با دستور CLS ال سی دی را برای نوشتن دوباره پاک می کنیم. در خط بعدی با دستور LCD G مقدار متغیر G را روی ال سی دی نمایش می دهیم یعنی مقدار جدید که همان 2 است. خط بعدی دوباره یک تاخیر ایجاد می شود و پس از سپری شدن این مدت زمان برنامه از خط بعدی اجرا می شود. خط بعدی هست LOOP باز هم مثل دفعه قبل برنامه از خط زیری DO شروع می شه و روز از نو روزی از نوع همینطور به متغیر G یکسره اضافه می شه تا جایی که متغیر G برسه به 255 وقتی که به این مقدار رسید متغیر G پر می شود (چون از نوع بایت است و با 255 پر می شود و با 256 سرریز می شود). وقتی که این 255 برسه به 256 متغیر سر ریز میشه و دوباره از صفر شروع به شمردن می کنه. دیگه از این ساده تر نمی شد توضیح داد.

1- اقا یک سوال

2- بپرس

1-توی مثالات به یک نکته توجه کردی

2- نه چی

- 1- مثال توی قسمتی که می خاستی توی یک حلقه مثال LCD رو بزنی نوشتی "ALI" LCD ولی وقتی که خاستی مقدار یک متغیر رو نشون بدی این نوشتی LCD G که مقدار متغیر G رو نشون بده حالا سوال اینجاست چرا ALI رو داخل " " گذاشتی ولی متغیر G رو داخل " " نذاشتی
- 2- سوال خوبی بود. بین هر وقت که خاستی یک متغیر رو نشون بدی لازم نیست که اونو داخل " "

قرار بدی یعنی نباید داخل " " قرار بدی ولی هر وقت که خاستی یک عبارت رو روی ال سی دی نمایش بدی باید اونو داخل " " قرار بدی

1- فهمیدم . این حلقه تموم شد

2- اره

1- پس اگه خاستیم از داخل این حلقه بیرون بیایم باید چکار کنیم

2- اینو برات می گم وقتی که دستورات شرط رو گفتم

خوب حالا رسیدیم به دستورات شرط

1- چی هست این دستور شرط

2- گاهی لازم است که اگر یک حالتی که برای ما مد نظر است اتفاق بیفته دستور شرط یک کاری رو مثلا

یک پایه رو برامون یک می کنه

1- بین نفهمیدم دوباره بگو

2- باشه برمیگردیم سر دو تاملال قبل که یک متغیر داخل یک حلقه بود و هر دفعه که دو ر می زد یک

واحد به متغیر اضافه می شد . خوب اگه این کار ادامه پیدا کنه این متغیر هی پرو میشه با 255 و دوباره از

صفر شروع می کنه دوباره پر می شه با 255 و دوباره از صفر شروع می کنه به اضافه شدن حالا ما حال

کردیم که اگر این متغیر G به مثلا 22 رسید یک کاری رو برامون انجام بده حالا هر کاری رو که حال

کردیم رو بره برامون انجام بده مثلا یک پورت رو بیاد برامون SET کنه یا اینکه بره رویه LCD یک چیزی

بنویسه . که ما باید از دستور شرط استفاده کنیم . یک مثال ساده دیگه که می تونم برات بزنم اینه که مثلا

بابای تو بهت می گه که اگر اخر سال معدل تو 20 شد برات دوچرخه می خرم . خوب بابات برای تو یک

شرط گذاشته که اگر این شرط که 20 شدن معدل تو است اجرا بشه . دستور شرط که خریدن دوچرخه

توسط بابات هست اجرا میشه . حالا فهمیدی

1- بابا ما از کوچیکی داریم 20 می گیرم بابامون برامون هیچی نگرفته . ولی مطلبو خوب گرفتم

2- عالیه. خوب بریم سر دستور شرط

IF همان دستور شرط است

THEN یعنی اگر شرط اجرا شود این THEN میاد برای ما یک کاری رو انجام می ده

1-مثال برن

2- همون مثال دوچرخه رو می زنم . خوب بابای فلانی بهش گفته که اگر معدلش 20 بشه یعنی شرط IF ..براش یک دچرخه می خره یعنی یک کاری رو در قبال شرطی رو که گذاشته برای پسرش انجام می ده

یعنی THEN

1- اها گرفتم. از مثال های برنامه هم برام بگو

2- باشه .بریم سر اون مثالی که متغیر G رو توی یک حلقه قرار داده بودیم حالا ما می خایم که اگر این متغیر G به 6 برسه بره و PORTA.0 رو SET کنه:

CONFIG PORTA = OUTPUT

DIM G AS BYTE

G = 0

DO

INCR G

CLS

LCD G

IF G = 6 THEN SET PORTA.0

WAITMS 50

LOOP

END

برنامه بالا رو که تا حدودی می دونی چیه چون قبلا توضیحشو دادم فقط می مونه خط هفتم که نوشته IF G = 6 THEN SET PORTA.0 ما در اینجا با گذاشتن شرط یعنی IF G = 6 شرط گذاشتیم که

اگر متغیر G به عدد 6 رسید . ادامه نوشتیم THEN SET PORTA.0 بروی SET کن . PORTA.0

فهمیدی

1- اره ولی مگه شرط بندی حروم نیست

2- چرا حرومه . ولی باید مواظب باشی که شرط رو نبازی

خوب رسیدیم به دستورات زیر برنامه (GOTO - GOSUB)

1- چی هستن این زیر برنامه ها

2- با یک مثال برات توضیح می دم توی مثال قبل اگر دقت کرده باشی متغیر G وقتی که به عدد 6 می

رسید می رفت و PORTA.0 رو SET می کرد و همینطوری به متغیر G اضافه می شد . حالا اگر که ما

حال کردیم بعد از این که متغیر G به عدد 6 رسید بره همون کاره خودشو یعنی SET کردن PORTA.0

رو انجام بده ولی برنامه دیگه تموم بشه و دیگه شمارش نکنه یا به عبارتی به END برسه . خوب برای این کار باید از زیر برنامه استفاده کنیم به مثال زیر توجه کن:

```
CONFIG PORTA = OUTPUT
DIM G AS BYTE
G = 0
DO
  INCR G
  CLS
  LCD G
IF G = 6 THEN GOTO AHMAD
WAITMS 50
LOOP
AHMAD :
SET PORTA.0
END
```

خوب حالا بریم تحلیل برنامه . برنامه رو که می دونی چجوریه منظورم معنی خط ها هستش البته به غیر از خط هشتم که نوشته

```
IF G = 6 THEN GOTO AHMAD
```

خوب این خط یعنی اینکه اگر متغیر G رسید به عدد 6 دستور شرط را اجرا کن که دستور شرط این است

GOTO AHAD یعنی بپر برو به زیربرنامه ای که استش هست احمد . توجه کن که به جای اسم

AHMAD هر اسم دیگه ای می تونی بزاری هر اسمی که حال کردی . بعد از اینکه متغیر G به عدد 6

رسید دستور شرط اجرا می شود و میکروکنترلر برنامه را از خط بعدی زیربرنامه AHNAD می خواند یعنی

خط SET PORTA.0 را می خونه و پورت A.0 رو ست می کنه بعد از این که ست کرد خط بعدی رو

می خونه که نوشته END به محض اینکه این دستور END رو خوند برنامه متوقف می شه و به پایان می

رسه .

پس فهمیدی که زیر برنامه برای وقتی هستش که ما می خوایم یک دستور شرط رو اجرا کنیم

به یک مثال دیگه توجه کن:

```
CONFIG PORTA = OUTPUT
DIM G AS BYTE
METAL:
```

```

G = 0
RESET PORTA.0
DO
INCR G
CLS
LCD G
IF G = 6 THEN GOTO AHMAD
WAITMS 50
LOOP
AHMAD:
SET PORTA.0
WAITMS 50
GOTO METAL
END

```

خوب حالا تحلیل برنامه قبل . میکرو میاد برنامه رو از خط اول شروع میکنه به خوندن که خط اول مربوط به پیکره بندی PORTA به عنوان خروجی هستش . بعد میاد خط دوم رو می خونه که در این خط ما یک متغیر که اسمش هست G و از نوع بایت می باشد رو تعریف کرده ایم. خط سوم یک زیر برنامه هستش به نام METAL که فعلا بهش کار ندارم تا بعد در ضمن این نکته رو باید یادت باشه که هر وقت یک زیر برنامه رو می نویسی باید جلوش دو نقطه رو بزاری اینطوری . :METAL خط بعدی به متغیر یک مقدار دادیم که صفر است . خط بعدی نوشتیم RESET PORTA.0 که به این هم فعلا کار نداریم . خط بعدی DO هستش که ابتدای حلقه را مشخص می کنه . خط بعدی نوشته INCR G یعنی به مقدار متغیر G یک واحد اضافه کن یعنی صفر که بوده حالا که یک واحد اضافه بشه می شه یک . خط بعدی با دستور CLS ال سی دی را پاک کردیم تا برای نوشتن دوباره آماده شود . خط بعدی با فرمان LCD G این را رساندیم که مقدار متغیر G رو روی ال سی دی نمایش بده که الان باید عدد یک رو روی ال سی دی نمایش بده . در خط بعدی نوشتیم IF G = 6 THEN GOTO AHMAD یعنی اگر متغیر G ما به عدد 6 رسید برو دستور شرط رو که هست THEN GOTO AHMAD اجرا کن یعنی پیر برو توی زیر برنامه AHMAD البته وقتی دستور شرط اجرا میشه که متغیر G به عدد 6 برسه . خط بعدی یک تاخیر در برنامه با مدت زمان 50 میلی ثانیه را گذاشتیم که بعد از سپری شدن این مدت زمان برنامه از خط بعدی شروع می شه

خوب تا این جا که متغیر ما عدد 1 هستش . بعد که میکرو میرسه به دستور LOOP دوباره دور می زنه و خط زیری DO رو می خونه که نوشته INC R G یعنی یک واحد به متغیر G اضافه کن خوب متغیر G الان هستش 1 بعد که به این دستور رسید این دفعه یک واحد بهش اضافه می شه و می شه دو . این عمل انقدر ادامه می یابد تا متغیر G برسه به عدد 6 . همین که به عدد 6 رسید از اونجایی که ما یک شرط با دستور

IF G = 6 THEN GOTO AHMAD اگر داشتیم یعنی اینکه وقتی متغیر G به عدد شش رسید یعنی IF G = 6 برو دستور شرط رو که هستش THEN GOTO AHMAD اجرا کن یعنی اینکه بپر برو توی زیر برنامه AHMAD خوب حالا میکرو میپره میره برنامه رو از خط زیری زیر برنامه AHMAD می خونه که نوشته SET PORTA.0 یعنی پایه A.0 رو یک کن . بعدش نوشته WAITMS 50 یعنی اینکه به مدت زمان 50 میلی ثانیه در برنامه تاخیر ایجاد کن که در طول این مدت PORTA.0 در وضعیت SET قرار دارد بعد از این که این مدت زمان سپری شد برنامه از خط بعدی اجرا می شه خط بعدی یعنی GOTO

METAL یعنی اینکه بپر برو توی زیر برنامه METAL و میکرو هم همین کار رو می کنه و می پره می ره توی زیر برنامه METAL و برنامه از خط بعدی این زیر برنامه که هستش G = 0 ادامه پیدا می کنه این خط به این معنی هستش که متغیر G رو که بوده 6 حالا بیا صفرش کن . در خط بعدی یعنی RESET PORTA.0 یعنی اینکه پایه A.0 رو که ست کرده بودی حالا بیا ریستش کن یعنی صفر کن. خوب حالا هم متغیر G ما صفر هستش هم اینکه PORTA.0 ما صفر یا همون ریست هستش . بعد میکرو می یاد برنامه رو از خط بعدی شروع میکنه به خوندن که نوشته DO یعنی اول زیر برنامه . خط بعدی نوشته

INC R G یعنی یک واحد به متغیر G ما که الان دوباره صفرش کردیم اضافه کن که بشه یک . دوباره روز از نو روزی از نو یعنی اینکه دوباره شروع می کنه به اضافه کردن متغیر تا دوباره برسه به 6 بعد که رسید به 6 دوباره بپره بره به زیر برنامه AHMAD اونجا باز PORTA.0 رو یک می کنه بعدش تاخیر ایجاد می کنه بعد با دستور GOTO METAL می پره میره به زیر برنامه METAL که در بالای برنامه هستش بعد دوباره متغیر رو که 6 هستش رو صفر می کنه به دنبالش PORTA.0 رو هم که یک بوده صفر می کنه باز دوباره.....

حالا رسیدیم به دستور GOSUB

این دستور همان دستور GOTO هستش با این تفاوت که یک دستور دیگه همراهش هست به نام اقای

RETURN که این RETURN رو هر جای برنامه که بنویسیم میکرو میاد برنامه رو از خط بعدی GOSUB شروع می کنه به خوندن . که دربارش وقتی که خواستم ورودی رو بهت درس بدم می گم. ورودی: PIN

ورودی در میکرو کاربردی فراوانی داره مثلا در ساده ترین حالت می خای یک میکروسوییچ به میکروکنترلر وصل کنی که با هر بار زدن این میکروسوییچ یک LED روشن و خاموش بشه در مراحل پیشرفته هم مثلا در ساخت یک ربات کاربرد داره به این صورت که یک سنسور مادون قرمز بهش وصل بشه و مثلا به یک خط سیاه حساس باشه و از این طور کارا خوب بریم سر دستور ورودی . باز هم می خوام با یک مثال برات توضیح بدم فرض کن که می خایم با یک میکرو سوییچ که به میکرو وصل کردیم یک LED رو یک بار روشن و یک بار خاموش کنیم که برنامه به صورت زیر است و خیلی خیلی ساده:

```
CONFIG PINB.0 = INPUT
DIM A AS BYTE
Ali:
DO
DEBOUNCE PINB.0 , 1 , SONY
LOOP
END
SONY:
INCR A
CLS
LCD A
GOTO ALI
```

خوب حالا وقت تحلیل برنامه هستش: در خط اول ما اومدیم با نوشتن دستور CONFIG PINB.0 = INPUT پایه B.0 رو به عنوان ورودی تعریف کردیم . در خط بعدی یک متغیر از نوع بایت که اسمش رو گذاشتیم A تعریف کردیم . در خط بعدی یک زیر برنامه به نام ALI گذاشتیم که فعلا بهش کار نداریم و در ادامه توضیحش می دم. در خط بعدی با نوشتن DO ابتدای حلقه را مشخص کردیم . در خط بعدی نوشتیم که DEBOUNCE PINB.0 , 1 , SONY که هر وقت خاستی که یک ورودی رو تعریف کنی بهتره از این دستور استفاده کنی یا به عبارتی حتما استفاده کنی خوب حالا لابد از خودت می پرسی که این

دستورات چی هست این دستور که همش باید توی یک خط نوشته بشن رو یکی یکی برات بازش می کنم

DRBOUNCE رو همیشه وقتی که می خای یک ورودی رو تعریف کنی اولش باید بزاری و کاربردش

اینه که یک تاخیر خیلی کوچیک که به چشم هم نمیاد ایجاد می کنه و باعث می شه که پایه ورودی

نویز نگیره یک بار دیگه دوباره می گم ما نوشتیم که **DEBOUNCE PINB.0** یعنی پایه **B.0** رو از

نظر نویز محافظتش کن خوب این تا بعدش علامت ، رو گذاشتیم تا جملات از هم تفکیک شوند که

باید حتما علامت ، رو بزاری در ادامه ما نوشتیم که **1** خوب حالا این **1** یعنی چی . این **1** یعنی اگر پایه

B.0 یا یک **1** شد یا به عبارتی **SET** شد با میتونیم بگیریم که یک نوع شرط است که در مقابل یک شدن پایه

B.0 گذاشته ایم و جلوی این **1** دوباره علامت ، گذاشتیم تا از جمله بعدیش تفکیک بشه . در جمله

بعدیش نوشتیم **SONY** که این یعنی اینکه اقا اگر این پایه **B.0** ما **1** شد پیر برو توی زیربرنامه **SONY** که

به جای این **SONY** هر اسم دخواه دیگه ای هم می تونی بزاری . یک بار دیگه برات این جمله رو به

صورت کلی توضیح می دم معنی این جمله دستوری این هستش که اگر **PINB.0** ما **1** شد پیر برو توی

زیربرنامه **SONY** و اگر صفر بود که هیچی . در خط بعدی نوشتیم **LOOP** که انتهای حلقه را با این کلمه

مشخص کردیم . در خط بعدی نوشتیم **END** که برنامه هیچگاه به **END** نمی رسد . در خط بعدی نوشتیم

SONY که **SONY** یک زیربرنامه است که می تونست هر اسم دیگه ای باشه . در خط بعدی نوشتیم

INCR A که به معنی اضافه کردن یک واحد به متغیر **A** می باشد . در خط بعدی نوشتیم که **CLS** یعنی

پاک کن کل صفحه ال سی دی رو تا برای نوشتن آماده بشه. در خط بعدی نوشتیم **GOTO ALI** یعنی پیر

به زیربرنامه . **ALI**

خوب حالا بریم سر طرز کار مدار : این مدار در واقع یک شمارنده هستش . طرز کارش هم اینطوره که

میکرو میاد سه خط اول رو می خونه بعد میره می یوفته توی حلقه **DO - LOOP** و اونجا داخل این حلقه

هی دور میزنه تا کی تا وقتی که **PINB.0** ما یک بشه یعنی شرطی که گذاشته ایم اجرا بشه که شرط ما این

هستش که پایه **B.0** یک بشه بعد از اینکه این شرط اجرا شد میکرو میاد دستور شرط رو اجرا می کنه

دستور شرط کدومه همونی که بهت گفتم یعنی پرش به زیر برنامه . **SONY** و میکرو که گوش به فرمان

برنامه هست می پره می ره داخل زیر برنامه **SONY** و از خط بعدی زیر برنامه **SONY** که نوشته

INCR A رو می خونه و طبق این دستور یک واحد به این متغیر **A** اضافه می کنه یعنی میکش **1** .

بعدش می یاد خط بعدی رو می خونه که نوشته **CLS** یعنی اینکه کل ال سی دی پاک بشه تا برای نوشتن

حاضر بشه . در خط بعدی نوشتیم که

LCD A به این معنی که مقدار متغیر A رو که هستش یک روی LCD نمایش بده . در خط بعدی نوشته
GOTO ALI یعنی اینکه بپر برو توی زیر برنامه ALI و میکرو هم همین کار رو می کنه و می پره می ره
اولین خط بعد از زیر برنامه ALI رو که نوشته DO رو می خونه و دوباره میوفته توی این حلقه و باز هم
دوباره انقدر توی این حلقه هستش و توی این حلقه دور می زنه تا وقتی که دوباره PINB.0 یک بشه یا
به عبارتی SET بشه اونوقت دوباره روز از نو روزی از نو و میپره دوباره توی زیر برنامه SONY و دستورات
داخل این زیر برنامه رو دوباره اجرا می کنه که در خط اول این زیر برنامه نوشته

INCR A یعنی دوباره یک واحد به متغیر A اضافه کن و میکرو هم این کار رو میکنه و متغیر A که
یک بوده الان میشه 2 بعد میاد دستور CLS رو می خونه که لازم به توضیح نیست . بعد می یا دستور
LCD A رو می خونه که مقدار جدید این متغیر A رو که الان هستش 2 رو روی LCD نمایش می ده بعد
میاد دستور GOTO ALI رو می خونه و می پره می ره توی زیر برنامه ALI دوباره میوفته توی حلقه و الی
آخر . با استفاده از این برنامه می تونی یک شما رنده درست کنی که می تونه از عدد 0 تا 255 رو بشماره
و اگر خاستی رنج شمارشش از 255 بیشتر باشه می تونی به جای BYTE توی برنامه بنویسی . WORD
این برنامه در کارخونه هم کاربرد داره البته با کمی تغییرات کوچیک در برنامه و مثلاً بایک سنسور می شه
قطعات عبوری از روی خط تولید رو شمارش کرد.

1- از نظر اتصالات سخت افزاریش هم توضیح بده برام
2- این که خیلی ساده هستش کافیه تغذیه میکرو رو وصل کنی و با یک میکروسویچ که یک سرش رفته
باشه به 5 VCC ولت و سر دیگه میکروسویچ هم رفته باشه به پایه B.0 و اتصالات ال سی دی این مدار
رو کاملش کنی.

1- جالب بود
2- تو به این می گی جالب بزار یکم راه بیوفتی اونوقت این طور مدارا برات پیش پا افتاده هستش . وقتی
که میتونی میکروکنترلر رو از طریق تلفن SMS. و اینترنت کنترلر کنی دیگه این مدارات در نظرت بچه
بازی هستش

1- چطوری از طریق اینایی که گفتی کنترل می کنن
2- دیگه باید یکم اطلاعات در مورد پورتهای کامپیوتر و PC INTERFACES کامل باشه که بتونی از

طریق مثلا اینترنت یک موتور رو کنترل کنی وگرنه برنامه نویسی میکرو که چیز نیست

1- با این PC INTERFACES چه کار میتونی بکنی و اصلا چی هستش

2- اندازه گیری و کنترل با استفاده از پورتهای استاندارد تحت ویندوز رو می گن PC INTERFACES
مثلا شما توی محیط نرم افزار ویرژال بیسیک یک برنامه ای می نویسی که اگر مثلا فلان دکمه کیبرد فشرده شد خروجی فلان پورت کامپیوتر رو یک کن که این خروجی ها از نظر استاندارد با استاندارد میکروکنترلر فرق دارند و باید از ای سی MAX232 استفاده کنی که از نظر استاندارد با هم مطابق شوند .

1- علم چقدر پیشرفت کرده ها

2- بله . بزار یک مثال دیگه هم بزنم ولی توضیحش رو خیلی خلاصه برات می گم

CONFIG PORTA = OUTPUT

CONFIG PINB.0 = INPUT

DIM A AS BYTE

Ali:

DO

DEBOUNCE PINB.0 , 1 , SONY

LOOP

END

SONY:

INCR A

CLS

LCD A

TOGGLE PORTA

GOTO ALI

در برنامه بالا فقط یک چیز اضافه شده اونم دستور TOGGLE PORTA که با وجود این دستور هر وقت که میکرو به زیر برنامه SONY پرش می کنه علاوه براین که یک واحد به متغیر A اضافه می کنه و روی ال سی دی هم نمایش می ده موقعیت PORTA رو هم عوض می کنه یعنی اگر صفر باشه یک می شه و اگه یک باشه صفر می کنه و دفعه بعد که به این زیر برنامه رفت دوباره اگر صفر باشه یکش می کنه و اگه یک باشه صفرش می کنه بزار منظورمو ساده بگم اگر شما یک LED به پایه های A.0 تا A.7 وصل کنی به ازای هر بار زدن میکروسوییچ این LED ها یک بار روشن و باردیگه خاموش می شن.