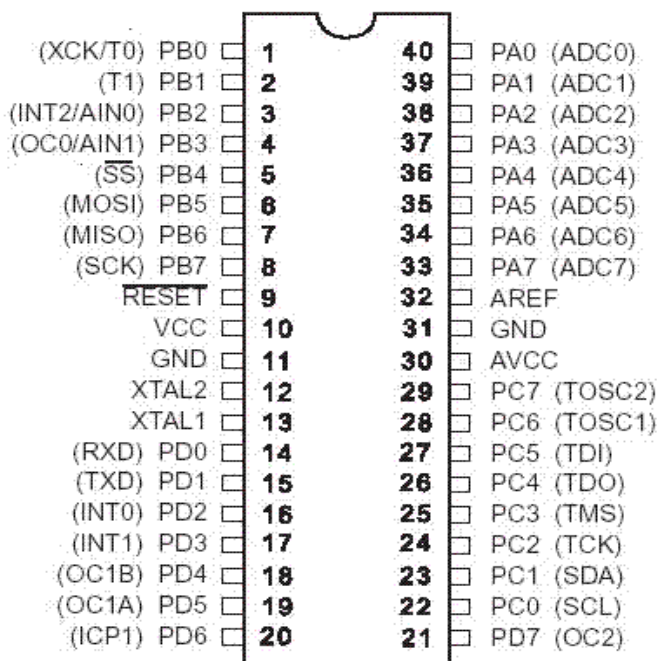


«به نام خدا»

با عرض سلام خدمت همه ی دوستان عزیز  
فوب، بر می گردیم سر میثم میکروکنترلرها....

رجیستر چیست ؟ رجیستری های DDRx , PINx , PORTx ، قسمتی از برنامه ی یک ربات مسیریاب بسیار ساده و ...



رجیسترها نوعی حافظه هستند که به طور  
مستقیم با بفرشش پردازشگر میکروکنترلر  
در ارتباط هستند. هر رجیستر یک بایت یا ۸  
بیت است. یکی از ویژگی های رجیسترها  
این است که به خاطر ارتباط نزدیک با  
پردازنده، سرعت بسیار بالاتری نسبت به  
سایر فانه های حافظه دارند....

قرار بود این جلسه برنامه ی یک مسیریاب بسیار ساده ی ۲ سنسوره را با هم بنویسیم. اما ابتدا باید چندتا نکته  
دیگه هم یاد بگیریم .

همونطور که گفته شد AT Mega16 دارای پایه های متعددی برای تبادل اطلاعات با مدار است. هر ۸ پایه ی مجاور  
که این وظیفه را دارند یک پورت نامیده می شوند (به شکل نگاه کنید) . AT Mega16 دارای ۴ پورت با نام های A  
، B, C و D می باشد. پایه های هر پورت به این شکل نمایش داده می شود :

شماره ی پایه . نام پورت

مثلاً اولین پایه ی پورت D به این صورت نشان داده می شود: D.0

و پایه ی سوم پورت C به صورت : C.2

حال به ترتیب پایه های ATMEGA16L دقت کنید:

(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(SS) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5 (TDI)
(TXD) PD1	15	26	PC4 (TDO)
(INT0) PD2	16	25	PC3 (TMS)
(INT1) PD3	17	24	PC2 (TCK)
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)

دقت کنید که شماره گذاری پایه ها در پورت ها از 0 شروع می شود .

همچنین گفته شد، پایه های میکروکنترلر می توانند به صورت ورودی یا خروجی تنظیم شوند، مثلاً در یک ربات مسیریاب میتوان چند پایه را تنظیم کرد که ورودی باشند و اطلاعات سنسورها را دریافت کنند، یا آنها را تنظیم کرد تا خروجی باشند و موتورها را هدایت کنند. این تنظیم به صورت نرم افزاری و با تنظیم رجیستر DDRX انجام می گیرد. اما ابتدا باید رجیستر را تعریف کنیم .

## رجیستر چیست ؟

رجیسترها نوعی حافظه هستند که به طور مستقیم با بخش پردازشگر میکروکنترلر در ارتباط هستند. هر رجیستر یک بایت یا ۸ بیت است. یکی از ویژگی‌های رجیسترها این است که به خاطر ارتباط نزدیک با پردازنده، سرعت بسیار بالاتری نسبت به سایر خانه‌های حافظه دارند.

## رجیستر DDRx :

رجیستر (Data Direction) DDRx برای تنظیم ورودی یا خروجی بودن پایه‌های میکروکنترلر است. برای تنظیم پایه‌ها در برنامه، باید به جای X باید آدرس پایه‌ی مورد نظر (مثل B.۳) را بنویسیم. اگر بخواهیم آن پایه خروجی باشد باید بیت رجیستر مربوط به آن را ۱ کنیم، و اگر بخواهیم آن پایه ورودی باشد، باید بیت رجیستر مربوط به آن را ۰ کنیم. به عنوان مثال اگر بخواهیم پایه ۱۷ یعنی D.۳ خروجی باشد باید این جمله را بنویسیم:  $DDR.D.3 = 1$ ; و اگر بخواهیم این پایه ورودی باشد:  $DDR.D.3 = 0$ ;

## رجیستر PORTx :

در صورتی که پایه‌ها به صورت خروجی تنظیم شده باشند، هر چه در این رجیستر نوشته شود سطح منطقی پایه متناظر را تعیین می‌کند، مثلاً اگر بنویسیم  $PORT.B.3 = 1$  یعنی پایه ی ۴، ۱ منطقی خواهد شد (یعنی ولتاژ ۵ ولت بر روی این پایه قرار می‌گیرد). و اگر بنویسیم  $PORT.C.1 = 0$ ، پایه ی ۱ C.۱ یعنی پایه ی ۲۳، ۰ منطقی خواهد شد (یعنی ولتاژ این پایه ۰ می‌شود).

## رجیستر PINx :

در صورتی که پایه‌ها به صورت ورودی تنظیم شده باشند، محتویات این رجیستر حاوی اطلاعات دریافتی از پایه‌های میکروکنترلر است. مثلاً اگر  $PIN.B.1 = 0$  باشد، یعنی بر روی پایه شماره ی ۲ یا همان B.۱؛ ۰ منطقی اعمال شده است (مثلاً اگر به سنسوری وصل شده است، خروجی سنسور ۰ منطقی بوده است). در حقیقت این رجیستر برای خواندن وضعیت پایه‌های ورودی مورد استفاده قرار می‌گیرد.

**نکته ی بسیار مهم :** دقت کنید که در زبان C، باید در انتهای هر خط از برنامه یک علامت ";" گذاشته شود. به این علامت در زبان انگلیسی سِمی کالِن می‌گویند.

## نکته ی مهم :

در مقیقت برای هر پورت ۳ رجیستر (حافظه ۱ بایتی) در داخل میکروکنترلر وجود دارد که به مجموع این ۱۲ رجیستر، رجیسترهای I/O (Input/Output) می گویند.

بسیار خوب، حالا نوبت نوشتن برنامه ی ۱ ربات مسیریاب ساده است که فقط ۲ تا سنسور داره !

نرم افزاری کمکی به نام Wizard Code در داخل همان Codevision وجود دارد که کار ما را برای انجام تنظیمات اولیه مانند تنظیم ورودی یا خروجی بودن پایه ها آسان می کند. یعنی دیگه نیازی نیست برای هر پایه تک تک با رجیستری DDR سرو کله بزنیم، و به راحتی با چند تا تیک ساده همه ی پایه ها رو تنظیم می کنیم. البته Code wizard همونطور که از اسمش هم معلومه بسیاری امکانات جادویی دیگری هم داره که در جلسات آینده به تدریج با آن ها آشنا خواهیم شد. Code Wizard در مقیقت برای ساده تر کردن و سریع تر کردن برنامه نویسی در فضای Codevision طراحی شده است و کارش این است که قسمت های زیادی از برنامه را به صورت خود کار و طبق خواسته های ما برای ما می نویسد.

پس با این مساب نیازی نیست تنظیمات رجیستری DDRX رو ما در برنامه خودمون انجام بدیم و این کار رو به wizard Code واگذار می کنیم. با Code wizard در جلسه ی آینده آشنا خواهیم شد.

پس در این جلسه فرض می کنیم تنظیمات اولیه مثل رجیستری DDRX و ... انجام شده است. پایه های B.۰ و B.۱ را به صورت ورودی (برای دریافت اطلاعات سنسورها) ، و پایه های B.۲، B.۳، B.۴ و B.۵ را به صورت خروجی (برای کنترل حرکت موتورها) تنظیم کرده می کنیم.

B۲ و B.۳ برای کنترل موتور سمت راست و B.۰ برای سنسور سمت راست .

B.۴ و B.۵ برای کنترل موتور سمت چپ و B.۱ برای سنسور سمت چپ .

حال مانند ربات قبلی، یک پایه از هر موتور را ۰ می کنیم؛ و روشن و خاموش کردن هر موتور را، با اعمال ۰ یا ۱ منطقی بر روی پایه ی دیگر کنترل می کنیم .

پایه دیگر را هم به صورت هماهنگ با سنسور متناظر آن سمت ۰ و ۱ می کنیم، یعنی اگر خروجی سنسور ۰ بود، پایه موتور را ۰ می کنیم و اگر ۱ بود ، پایه را ۱ کرده و موتور را فعال می کنیم. (به شرطی که از مدار گیرنده ی شماره ۲

استفاده شود (جلسه ی ۱۵))

در زبان C علامت "-" یک عملگر است که عملوند سمت راست خود را فوآنده و در عملوند سمت چپ خود می ریزد.  
مثلاً وقتی می نویسیم :

```
PORTB.۳=PINB.۰;
```

ابتدا مقداری B.۰ فوآنده می شود و سپس بر روی B.۳ ریخته می شود. یعنی مثلاً اگر روی B.۰ ، ۱ منطقی اعمال شده باشد، پایه B.۳ نیز ۱ منطقی می شود.

مال با توضیحات داده شده به برنامه ربات مسیر یاب ساده دقت کنید :

```
PORTB.۲=۰;
```

```
PORTB.۴=۰;
```

```
PORTB.۳=PINB.۰;
```

```
PORTB.۵=PINB.۱;
```

همانطور که می بینید این برنامه بسیار ساده و کوتاه است .

در جلسات آینده سعی می کنیم شما رو با Code wizard بیشتر آشنا کنیم .

آموزشهای رباتیک طبقه بندی شده توسط کمیته مهندسی رباتیک / [nrec.ir](http://nrec.ir) ( طرح ساماندهی آموزش رباتیک در

اینترنت ) برگرفته از سایت رشد مخصوص رده سنی ۱۳ تا ۲۵ سال

گردآوری و ویرایش اولیه : خانم فرناز عطاءاللهی

ویرایش علمی و گرافیکی نهایی : زهره دارابیان



فروشگاه عرضه قطعات الکترونیک ، مکانیک و رباتیک

*RoboChip.ir*