بسم الله الرحمن الرحیم

تحت عنوان

بررسی میکروکنترلرها و کاربرد آن در ربات

فهرست مطالب

عنوان **صفحه**

فهرست جداول(اشکال)

عنوان **صفحه**

چکیده:

مطالبی که پیش رو دارید میکروکنترلرها را از جنبه های مختلف مورد بررسی قرار داده و امید است خواننده را به سمت کار عملی با این میکروکنترلرها هدایت کند.

در این مجموعه سعی بر این بوده است که از پرداختن به جزئیات در صورت امکان پرهیز شود تا خواننده با سرعت به مفاهیم مسلط شود.

برای راحتی کار و تسهیل در امر خواندن و درک سریع مطالب ، مطالب به چندین فصل تقسیم شده است.

فصل اول:

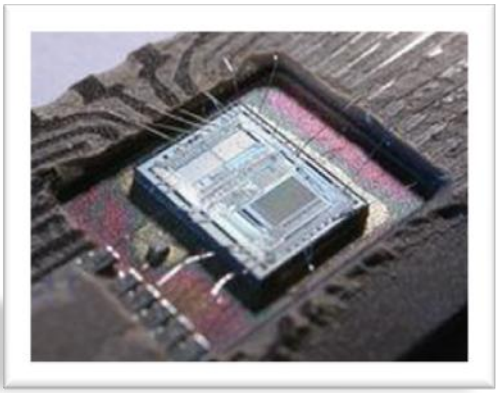
آشنایی با میکروکنترلر

تاریخچه ساخت میکروکنترلر:

در سال 1971 میلادی شرکت اینتل اولین میکروکنترلر را ساخت و اولین میکروکنترلر را با نام 8080 در اوایل سال 1980 روانه بازار کرد (همان شرکت اینتلی که الان در ساخت cpu یکه تاز دنیاست) اما بعدا این امتیاز را به شرکت های دیگری واگذار کرد و شرکت های زیادی در حال حاضر میکروکنترلر های مختلف تولید میکنند.

میکروکنترلر چیست :

میکروکنتلر (با نام های µC ،uC، MCU نیز شناخته می شود) یک رایانه ی کوچک بر روی مدار مجتمع است که شامل یک پردازنده ، حافظه و وسایل ورودی/ خروجی قابل برنامه ریزی است.اغلب حافظه ها از جنس Flash و یا OTPROM (حافظه ی غیر قابل پاک شدن) وRAM بر روی این تراشه ها موجودند
میکروکنتلرها برای سیستم های تعبیه شده طراحی و ساخته شده اند.



شکل 1-1 .تراشه میکروکنترلر

معرفی میکروکنترلرها :

به آی سی هایی که قابل برنامه ریزی می باشد و عملکرد آنها از قبل تعیین شده میکروکنترلرگویند میکرو کنترلر ها دارای ورودی – خروجی و قدرت پردازش می باشد.

کلمه میکروکنترلر :

این کلمه از دو کلمه 1- میکرو 2- کنترلر تشکیل شده است.
میکرو : یک واحد یونانی است و برابر با 10 به توان منفی 6 متر است. یعنی یک ملیونیوم متر واحد(خیلی کوچک است).

کنترلر : یعنی کنترل کننده به تعبیری یعنی "مغز " البته بدون تفکر فقط دستوراتی که به آن داده میشود به نحو احسن انجام میدهد.

نحوه انجام دادن کار میکروکنترلر:

تا حالا همه شما با ماشین حساب کار کردید تا حالا به نحوه کار کردنش فکر کردید! شما اطلاعاتتان را که همان عملیات ریاضی است به وسیله صفحه کلید به ان میدهید بعد ماشین حساب این اطلاعات را بر مبنای دستوراتی که قبلا به ان داده شده پردازش میکند و جواب را روی lcdنمایش میدهد. در واقع یک میکروکنترلر برنامه ریزی شده به عنوان مغز ماشین حساب این اطلاعات یا داده را از صفحه کلید میگیرد و روی انها پردازش انجام میدهدو بعد بر روی lcd نمایش میدهد .

میکرو کنترلر بر مبنای یک سری ورودی که به آن داده میشود مثلا این ورودی از یک سنسور دما باشد که درجه حرارت را میگوید یا از هر چیز دیگر مثل صفحه کلید بر مبنای این ورودی ها و برنامه ای که قبلا ما به آن دادیم خروجی اش را تنظیم میکند که ممکن است خروجیش یک موتور باشد یا یک lcd یا هر چیز دیگری که با الکتریسیته کار بکند. حالت دیگری هم میتواند باشد که فقط میکروکنترلر بر مبنای برنامه ای که به آن دادیم عمل کند و خروجیش را فقط بر اساس برنامه بگیرد.

ساختمان داخلی میکروکنترلر :

کامپیوتری که الان بر روی آن دارید کار انجام میدهید دارای یک پردازنده مرکزی به نام cpu است که از کنار هم قرار گرفتن چندین میلیون ترانزیستور تشکیل شده و بر روی اطلاعات پردازش انجام میدهد. میکرو کنترلر هم عینا دارای یک پردازنده مرکزی به نام cpu است که دقیقا کار cpu کامپیوتر را انجام میدهد با این تفاوت که قدرت و سرعت پردازشش از cpu کمتراست که به آن میکروپرسسور میگویند در بخش بعدی فرق میکرو پرسسور و میکروکنترلر را بررسی میکنیم. میکروکنترلر علاوه بر cpu دارای حافظه است که ما برنامه ای که به آن میدهیم در آن قرار بگیرد در کنار حافظه در میکروکنترلرهای امروزی تایمرها برای تنظیم زمان کانتر ها برای شمردن کانال های آنالوگ به دیجیتال پورت ها برای گرفتن و دادن اطلاعات و امکاناتی دیگر که بعدا مفصل راجع

به هر کدام توضیح داده میشه تشکیل شده و همه اینها در یک چیپ قرار گرفته که تکنولوژی جدید آن را در یک تراشه به اندازه یک سکه قرار داده است.

تفاوت میکروپروسسور و میکروکنترلر :

میکروپروسسور همانطور که گفته شد یک پردازنده است و برای کار باید به آن چیپ ه‌ای حافظه و چیز های دیگری را به آن اضافه کرد این امکان به درد این میخورد که بر حسب کارمان حافظه مناسب و دیگر قطعات را مانند تایمرها و غیره به صورت بیشتری استفاده کنیم ولی مدار خیلی پیچیده میشود و از لحاظ هزینه هم هزینه بیشتر میشود به همین دلیل امروزه از میکروپرسسورها کمتر استفاده میشود اما این روزها میکرو کنترلر های جدید با حافظه های زیاد ،تعداد تایمر زیاد، پورت های زیاد و تنوع بسیار زیاد انها بر حسب این امکانات دست ما را باز گذاشته است تا دیگر میکروپرسسورها را فراموش کنیم.

بخش های مختلف میکروکنترلر :

میکروکنترلر ها از بخشهای زیر تشکیل شده اند

Cpu (واحد پردازش)

Alu (واحد محاسبات)

I /O (ورودی ها و خروجی ها)

Ram (حافظه اصلی میکرو)

Rom (حافظه ای که برنامه روی آن ذخیره می گردد)

Timer (برای کنترل زمان ها)

و . . .

معرفی انواع میکروکنترلر:

تمام میکروکنترلرها جزء این 5 قسمت هستنند :

8051

Pic

Avr

6811

Z8

البته مدل های 6811 ساخت شرکت موتورلا و z8 ساخت شرکت زایلوگ حداقل در ایران خیلی کم استفاده می شوند و رقابت اصلی بین سه نوع دیگر است .

تا به امروز هر میکروکنترلری که ساخته شده زیر مجموعه یکی از این5 نوع است. البته کارخانه های خیلی زیادی با مارک های مختلف میکرو کنترلر تولید میکنند ولی همه اونها زیر مجموعه یکی از این 5 قسمت هستند. شما برای هر کدام از این5 نوع میکروکنترلر میتوانید میکروکنترلر های مختلفی از شرکت های مختلفی را پیدا کنید.

(البته در بازار ایرن کمی با مشکل)

اما خوشبختانه همه میکروکنترلر هایی که جزء هر کدام از 5 نوع بالا باشند از یک برنامه پیروی میکنند. بدین معنا که اگر شما کار با یکی از مدل های آن میکرو را یاد گرفته باشید مثل اینکه کار با تمام میکروکنترلرهای آن نوع را یاد گرفته اید.مثلا شما اگر با یکی از مدل های میکروکنترلر avr مثلا atmega8 را یاد گرفته باشید دیگر با صد ها مدل دیگر میکروکنترلر avr مشکلی ندارید وتقریبا بدون هیچ مشکلی میتوانید با دیگر مدل های این میکرو کار کنید .

اما یه مشکل که در میکروکنترلر ها وجود دارد این است که این5 نوع از لحاظ برنامه نویسی به هیچ وجه با هم دیگر سازگاری ندارند . به طور مثال اگر شما میکروکنترلر های avr و 8051 را کامل یاد گرفتهباشید حتی ساده ترین برنامه رو روی یک میکروکنترلر pic نمیتوانید اجرا کنید. واین یکی از بزرگترین عیب و مشکل برای یاد گیری میکرو است. بنابراین از همون اول باید یک انتخاب درست داشته باشید و میکروکنترلر مناسب را برگزینید تا با یادگیری آن میکروکنترلر بتوانید بعدا به سادگی پروژه های خود را اجرا کنید. البته بسیاری از دوستان هستنند که کار با چند میکروکنترلر را میدونند ولی اگر به صورت خیلی حرفه ای نخواهید وارد این بحث بشوید باید یکی از این میکروکنترلرها را انتخاب کنید و کار با آن را آغاز کنید.

فصل دوم:

میکروکنترلر 8051

مشخصات سخت افزاری 8051

1) 4 کیلوبایت ROM داخلی
2) 128 بایت RAM داخلی
3) چهار درگاه I/O 8 بیتی
4) درگاه ارتباط سری
5) دو زمان سنج و شمارشگر 16 بیتی
6) حداکثر 64k بایت حافظه خارجی برای برنامه‌های بزرگتر
7) حداکثر 64k بایت حافظه خارجی برای داده‌های بزرگتر
8) توانایی پردازش عملیات بولی
9) 210 بیت با امکان دسترسی بیتی
10) وجود دستورات اضافی نظیر ضرب و تقسیم
11) قبول وقفه از اجزای داخلی و سخت افزار خارجی

وضعیت پایه های 8051 و ساختار I/O

32 پایه از 40 پایه تراشه می توانند بصورت خطهای درگاه I/O بکار روند.برای صرفه جویی 24 تا از آنها دارای عملکرد دیگری نیز هستند و در صورت لزوم از عملکرد دوم هر پایه استفاده می شود که به همین دلیل پایه های دو منظوره نام گرفته اند.

درگاه 0: پایه های 32-39 را شامل می شود. در سیستمی که حافظه خارجی نداشته باشد این پایه ها به عنوان درگاه I/O بکار می روند. ولی در سیستمهای با حافظه خارجی این پایه ها به عنوان گذرگاه(آدرس و داده) multiplexed بکار می روند.

درگاه 1: پایه های 1 تا 8 IC را شامل می شود که فقط به عنوان درگاه I/O بکار می رود.

درگاه 2: پایه های 21 تا 28 را شامل می شود این پایه ها دو منظوره طبق برنامه می‌توانند به عنوان درگاه I/O یا در صورت بهره گیری از حافظه کد خارجی یا داده خارجی با حجم 64k ,بایت به عنوان MSB گذرگاه آدرس استفاده شوند.

درگاه 3: پایه های 10 تا 17 تراشه را شامل می شود که به ترتیب دریافت و ارسال اطلاعات به صورت سری، وقفه خارجی 0 , وقفه خارجی 1, ورودی زمان سنج 0 , ورودی زمان سنج 1 و پایه هایی که همزمانی و فرمان نوشتن برای حافظه داده خارجی را فراهم می آورند.

ورودی نوسان ساز داخلی: برای راه اندازی بین پایه های شماره 18 , 19 یک کریستال و دو خازن قرار می دهیم.

اتصال منابع تغذیه : ولتاژ 5+ به پایه 40 متصل می شود و پایه 20 زمین می شود.

PSEN، فعال کننده پایه 29: یک سیگنال خروجی کنترلی است که برای همزمانی و فعال ساختن حافظه کد خارجی در هنگام خواندن استفاده می شود. که غالباً به پایه $\overline{OE}$ یک EPROM به پایه‌های 8051 وصل می شود. هنگامی که برنامه از ROM داخلی 8051 اجرا می شود PSEN در حالت high باقی می‌ماند.

باز نشاندن پایه9: ورودی باز نشاندن IC است . اگر حداقل دو سیکل ماشین high نگه داشته شود IC بازنشانده می شود. (بعد از باز نشاندن IC وضعیت ثباتها مطابق جدول (1) خواهد شد.)

جدول 2_1. مقادیر ثبات پس از بازنشاندن

مقدار ثباتها بعداز بازنشاندن	ثبات
0000 H	PC شمارنده برنامه
00 H	AC انباره
00 H	B ثبات
00 H	PSW کل وضعیت برنامه
07 H	SP اشاره گر پشته
0000 H	DPTR اشاره گر داده
FF H	Port 0-3
XXX0000B	IP اولویت وقفه
0XX00000B	IE فعال ساز وقفه
00 H	Timer ثبات زمان سنج
00 H	SCON ثبات کنترل درگاه سری
00 H	S BUF بافر داده سری
0XXXXXXXB	PCON(HMOS) ثبات کنترل توان
0XXX0000B	PCON(CMOS) ثبات کنترل توان

ALE فعال کننده قفل آدرس (پایه30): هنگامی که این پایه high باشد خطوط خروجی درگاه صفر به عنوان 8 بیت LSB آدرس در ثبات خارجی قفل(لچ) می شوند و وقتی که ALE ، low باشد درگاه به خطوط داده

تبدیل می شود. هنگام استفاده از حافظه خارجی در هر سیکل خواندن یا نوشتن حافظه ابتدا ALE ، high و سپس low می شود.

EA (پایه 31): معمولا بطوردائمی به منبع 5Vیا زمین متصل می شود اگر این پایه high باشد برنامه از ROM داخلی اجرا می شود در غیر این صورت از EPROM خوانده می شود.

درگاهI/O: 0 هر بیت درگاه I/O را می تواند بصورت مجزا برای خواندن ,نوشتن, تغییر یا تست کردن آدرس دهی کرد پایه های I/O نقش ورودی و خروجی را به عهده دارد.

سازمان حافظه : معمولاً در پرسسورها کد برنامه داده روی حافظه RAM جانبی قرار دارد که در ابتدا یک بار از روی دیسک فراخوانی می شود ولی در میکرو کنترلر برنامه کنترل روی ROM و یا EPROM جای می گیرد و میکرو کنترلرها حافظه جداگانه‌ای به داده اختصاص می دهند.

دو ویژگی حافظه داخلی اینست که :

الف) ثباتها و درگاههای I/O ،memory mapped شده‌اند یعنی مانند دیگر مکانه‌ای حافظه قابل آدرس دهی هستند.

ب) برای اختصاص حافظه به پشته فقط از RAM داخلی استفاده می شود. سازمان RAM داخلی نشان داده شده است که از چهار قسمت: 1) بانکهای ثبات. 2) فضای انباشت عمومی. 3) فضای قابلیت آدرس دهی بیتی. 4) ثباتهای ویژه SFR . تشکیل شده است.

ثبات های انتقال سری: شامل دو ثبات SBUF (بافر انتقال سری) و SCON(ثبات کنترل انتقال سری) هستند که به ترتیب با آدرس 98H و99H مشخص می شوند SBUF در واقع در بردارنده بیت ارسالی یا دریافتی در یک عمل ارسال یا دریافت است. SCON برای برنامه ریزی ارتباط سری بکار می‌رود.

ثبات های وقفه: 8051 می تواند به 5 منبع وقفه در دو سطح اولویت پاسخ دهد. میکروکنترلر در صورت نوشتن داده مناسب در IE, به وقفه های دریافتی پاسخ می‌دهد آدرس ثبات فوق A8H است.

برای تعیین سطح اولویت از ثبات IP به آدرس B8H استفاده می شود.

دسترسی به حافظه خارجی:

برای ذخیره کد برنامه در EPROM ، ابتدا خروجی PSEN ، 8051 را به ورودی EPROM,OE متصل می‌کنیم سپس داده‌ها را به آدرس مورد نظر در EPROM می فرستیم. پایه ALE ، عملکرد نگهدارنده 373 را کنترل می کند. 373 داده اول را که 8 بیت LSB آدرس است در خود نگه می دارد سپس داده ها از مسیر بعدی به دو آدرس(A0 . . . A15) حافظه خارجی ارسال می شود.

انواع آدرس دهی

آدرس دهی ثبات: این نوع آدرس دهی یک بایتی است. 5 بیت اول opcade دستور وسه بیت LSB مربوط به شماره ثبات است. بعضی از این دستورات ویژه ثباتی خاص مانند DPTR وAC هستند این گونه دستورات به بیتهای آدرس نیازی ندارند و opcade ثبات را نیزمشخص می کند.

مثال: ADD A,R 7

مثال: INC DPTR

آدرس دهی مستقیم: آدرس دهی 2 بایتی است. که بایت اول آن opcade مربوط به دستور العمل و بایت بعدی آدرس مورد نظر است.

مثال : MOV A,01

آدرس دهی غیر مستقیم: برای غیر مستقیم کردن آدرس دهی یک مکان در RAM داخلی از ثبات R0 یا R1 استفاده می کنند. این نوع دستورات یک بایتی است. بیت آخر مشخص می کند که آدرس دهی غیر مستقیم مربوط به R0 یا R1 است.

مثال:MOV @ R0, #FFH

آدرس دهی فوری (immidiate): زمانی که عملوند یک مقدار ثابت است مثلاً قرار است یک مقدار ثابت با محتوای مکان از حافظه شود از این نوع آدرس دهی استفاده می کنند.

مثال: MOV DPTR,#8000

آدرس دهی نسبی: این نوع آدرس ده‌ی در دستورات پرش استفاده می شود و دستورات دوبایتی هستند که بایت اول opcade دستورالعمل، بایت دوم آدرس نسبی است پس از اجرای دستور به PC دو واحد اضافه می شود و مقدار آدرس نسبی هم به آن اضافه شده و برنامه به آدرس جدید PC جهش می کند.

آدرس دهی مطلق: این نوع آدرس دهی فقط در دستورات AJMP , ACALL استفاده می‌شود. که دستور 2 بایتی است، و آدرس دهی در یک صفحه 2^{10} مکان را ممکن می سازند.

A10	A9	A8	Opcade

A7	A6	...	A0

آدرس دهی 2 بایتی :

زیرا در این نوع آدرس A11 تا A15 تغییر نمی‌کند و می توانیم دستور فوق را روی یک مکان حافظه که در آن صفحه است انجام دهیم

AJMP 021BH

با انجام این دستور بدون اینکه 5 بیت سمت چپ PC تغییر کند 11 بیت سمت راست آن مساوی عدد 021BH می شود.

آدرس دهی بلند: در این نوع آدرس دهی که فقط برای دستورات LJMP , LCALL است, دستور 3 بایتی است. بایت اول برای opcade است و 2 بایت دیگر آدرس، هستند.

مثال: MOVC A,A+DPTR

MOVC A,A+PC

آدرس دهی شاخص دار: در آدرس دهی فوق، مقدار یک ثبات پایه 1 با افزوده شدن به مقدار انباره یک آدرس موثر بوجود می آورد.

انواع دستورالعمل ها:

8051 ، پنج گروه دستورالعمل دارد که در این بخش به هر یک از آنها می پردازیم.

دستورالعمل های حسابی:

این گروه دستورالعملها شامل جمع و تفریق و ضرب و تقسیم است درباره دستورات INC R , DIV AB , MUL AB قبلاً توضیح دادیم. از دستور SUB , ADD می توان در انواع آدرس دهی استفاده کردکه در چند مثال انواع آن نشان داده شده است.

آدرس دهی مستقیم: ADD A,7FH

آدرس دهی غیر مستقیم: ADD A, @ RO

آدرس دهی رجیستری : ADD A,R7

آدرس دهی فوری : ADD A,#35H

دستورالعمل های منطقی و چرخشی:

دستورالعملهای منطقی شامل AND , OR , XOR , NOT است که بیت به بیت روی بایتهای داده انجام می شود و حاصل در انباره ذخیره می شود.

در این دستورالعملها هم انواع آدرس دهی وجود دارد که همانند قسمت قبلی است.

دستورالعمل چرخشی شامل RLC A , RRC A , RLA , RRA است که مانند پروسسورها برای چرخش به راست و چپ بدون رقم نقلی یا همراه رقم نقلی بکار می روند.

دستور SWAP A چهار بیت بالا و پایین انباره را جابجا می کند.

دستورالعمل های انتقال داده:

این دستور العملها برای جابجا کردن داده ها در حافظه بکار می روند. این گروه دستورالعملها را به دیدگاه دیگر سه گروه تقسیم بندی می‌کنند. که بطور مختصر در ذیل آمده است.

RAM داخلی:

دستورالعملهای این گروه در یک یا دو سیکل ماشین اجرا می شوند. و شامل انواع آدرس دهیها است.

RAM خارجی:

همانطور که قبلاً بیان شد برای دستیابی به حافظه خارجی از آدرس دهی غیر مستقیم یک بایتی (@Ri) یا دو بایتی (@DPTR) ازدستور MOVX استفاده می کنیم البته موقع استفاده از RAM خارجی و آدرس دهی 16 بیتی درگاه 2 برای آدرس دهی استفاده می کنیم در صورتی که هر پایه این درگاه برای کنترل یک وسیله خارجی استفاده شود این کار در یک سیستم که با 8051 طراحی شده کار مناسبی نیست.خروجیها $\overline{WR}, \overline{RD}$ به RAM خارجی متصل می شوند تنها موقع استفاده از دستور MOVX فعال می شوند. وعمدتاً غیر فعال (وضعیت بالا) هستند.

جدولهاي جستجو :

دو دستورالعمل انتقال داده MOVC A,A+PC وMOVC A,A+DPTR برای خواندن جداول جستجو واقع در حافظه ROM وجود دارد .

مثال:

```
LOOK – UP :  INC A
MOVE A , @ A+PC
RET
DB       23H
```

DB B2H

با قرار دادن مقدار 0 در A فراخوانی این زیر برنامه عدد 23H درA قرار گیرد.همچنین بازای عدد مقدارB2H باز می گرددو ...INC A. در ابتدا برای رد کردن محل دستور RET در حافظه نوشته شده است.

دستورالعمل های بولی:

8051 یک پردازنده بولی کامل برای عملیات تک بیتی 210 مکان بیت آدرس پذیر دارد این دستورالعملها شامل جابجایی، 1 کردن، 0 کردن، متمم کردن، AND , OR و دستورالعمل های انشعاب است. برای مثال به دستور العملهای زیر نگاه کنید.

```
FLAG1   BIT   05

FLAG1   BIT   05

MOV   C,FLAG1

JNB  FLAG2,SKIP

CPL C

 SKIP:    MOV  A,#2BH
```

فرمان های اسمبلر هستند و دو دستور اول FLAG1 وFLAG2 را به عنوان بیتهای موجود در محلهای 05 و 06 معرفی میکند.

دستورالعمل خط 3 FLAG1 را به رقم نقلی منتقل می کند. دستورالعمل خط 4 در صورت 1 نبودن FLAG2 به SKIP پرش می کند. دستورالعمل خط 5 پر نقلی را NOT می کند.

دستورالعمل انشعاب:

سه نوع دستورالعمل پرش (نسبی، طولانی، مطلق) وجود دارد.

که با نماد AJMP , LJMP , SJMP نشان داده می شود.

تایمرها:

تایمرهای روی تراشه 8051 تایمر 16 بیتی هستند. تایمرها با یک سری فیلیپ فلاپ تقسیم کننده بر دو ساخته می شوند. در تایمرها هر بیت خروجی یکی از فیلیپ فلاپهاست. در 8051 ، تایمر از عدد 0000H تا FFFFH را می شمارد بعد دوباره از 0000 شروع به شمارش می کند و پرچم سرریز 1 می شود. که مجموعاً اعداد 0 تا 65536 را شامل می شود.

یک تایمر16 بیتی سه نوع عملکرد در 8051 دارد:

1) زمان بندی فاصله های زمانی. 2) شمارش اتفاقات. 3) تولید نرخ ارسال بیت برای درگاه سریال داخلی

روش خواندن یک تایمر در حال شمارش: برای نشان دادن تعداد پالسهای ورودی یک شمارگر یا ساعت یک تایمر ابتدا THX , TLX را در دو ثبات ذخیره می کنیم بعد آن ثباتها را می خوانیم واین کار را تکرار می کنیم.

مثال :

```
AGAIN : MOV A , TH1
MOV R6 , TL1
MOV A , TH1 ,A, AGAIN
MOV R7 , A
```

وقفه: با وقفه در مباحث قبل آشنا شده اید هنگام وقوع وقفه یک برنامه به صورت موقت، متوقف می شود و برنامه مربوط به وقفه که یک روال سرویس وقفه ISR (interrupt service routine) نامیده می شود اجرای می شود و پس از اجرای ISR اجرای برنامه اصلی ادامه می یابد. معمولاً برنامه اصلی را سطح پایه و ISR سطح وقفه می گویند یک نمونه از وقفه صفحه کلید است.

در 8051، پنج منبع وقفه وجود دارد: دو وقفه خارجی، دو وقفه تایمر، یک وقفه درگاه سریال.

در بخشهای بعدی به بحث فعال و غیر فعال کردن وقفه ها تقدم و ترتیب اجرا می پردازیم.

فعال و غیر فعال کردن وقفه ها:

برای فعال یا غیر فعال کردن وقفه ها از آدرس 0A8H استفاده می کنیم که بایت فوق بیت آدرس پذیر است. در ضمن یک بیت برای فعال یا غیر فعال کردن همه وقفه ها وجود دارد.

وقتی که این بیت 1 باشد می تواند بعضی از وقفه ها در حالت فعال قرار گیرد.

جدول2_2. فعال کننده وقفه (IE)

فعال/ غیر فعال کلی	AFH	EA	IE.7
	AEH		IE.6
	ADH		IE.5
فعال کردن وقفه درگاه سریال	ACH	ES	IE.4
1 فعال کردن وقفه تایمر	ABH	ET1	IE.3
خارجی1 فعال کردن وقفه	AAH	EX1	IE.2
0 فعال کردن وقفه تایمر	A9H	ET0	IE.1
خارجی0 فعال کردن وقفه	A8H	EX0	IE.0

هر منبع وقفه توسط ثبات IP در یکی از دو سطح در آدرس 0B8H برنامه ریزی می شود. اگر دو وقفه به صورت همزمان اتفاق بیفتد. میکرو کنترلر وقفه با سطح تقدم بالاتر را سرویس دهی می کند . در صورت بازنشاندن میکرو کنترلر همه وقفه ها در سطح تقدم پایین قرار می گیرند.

جدول2_3.تقدم وقفه (IP)

IP.4	PS	تقدم وقفه درگاه سریال
IP.3	PT1	1 تقدم وقفه تایمر
IP.2	PX1	خارجی1 تقدم وقفه
IP.1	PT0	0 تقدم وقفه تایمر
IP.0	PX0	خارجی0 تقدم وقفه

ترتیب اجرا:

فرض کنید دو وقفه با سطح تقدم یکسان همزمان اتفاق بیفتد سپس میکروکنترلر با توجه به ترتیب اجرا، به آنها سرویس دهی می کند ترتیب اجرا چنین است: 0 خارجی، تایمر 0، 1 خارجی ، تایمر 1، درگاه سریال.

وقفه های پردازنده :

پس دریافت وقفه، میکرو کنترلر اعمال زیر را انجام می دهد.

1) اجرای دستور العمل جاری کامل می شود.

2) PC در پشته ذخیره می شود.

3) وضعیت وقفه جاری بطور داخلی ذخیره می شود.

4) وقفه های همسطح با این وقفه متوقف می شود.

5) PC با آدرس ISR بار می شود.

6) ISR اجرا می شود.

7) پس از اجرای ISR با دستور RETI، مقدار قبلی PC از پشته بازیابی شده و اجرای برنامه ادامه می یابد.

بردارهای وقفه:

هنگامی که وقفه مورد پذیرش میکروکنترلر قرار بگیرد محتوای PC را بردار وقفه می نامیم . که این عدد آدرس شروع ISR مربوط به وقفه است. پس از بردار کردن یک وقفه پرچمی که باعث وقفه می شود بطور خودکار توسط سخت افزار پاک می شود.

جدول2_4.بردار وقفه

وقفه	پرچم	آدرس برداری
سیستمReset	RST	0000 H
خارجی0	IE0	0003 H
0تایمر	TF0	000 BH
خارجی1	TE1	001 BH
1تایمر	TF1	000023
درگاه سریال	RI یا TI	002 BH

هستند برای پاک کردن پرچم وقفه باید از نرم افزار استفاده کنیم تشخیص منبع وقفه هم توسط نرم افزار انجام می شود.

برنامه فوق موج مربعی 10KHz را روی P1.0 ایجاد می کند.

```
0000                    ORG     0               ; reset entry point
0000    020030          LIMP    MAIN            ; jump above interrupt vectors
000B                    ORG     000BH           ; Timer 0 interrupt vectop
000B    B290    TOISR:  CLR     P1.0            ; toggle port bit
000D                    RETI                    ;
0030    32              ORG     0030H           ; Main Program entry point
0030    758902  MAIN    MOV     TMOD, #02H      ; timer 0 , mode 2
0033    758CCE          MOV     TH0, #-50       ; 50 us delay
0036    D28C            SETB    TRO             ; start timer
0038    75A882          MOV     IE, #82 H       ; enable timer 0 interrupt
003B    80FE            SIMP    $               ; do nothing
```

فصل سوم:

میکروکنترلر AVR

میکروکنترلر AVR

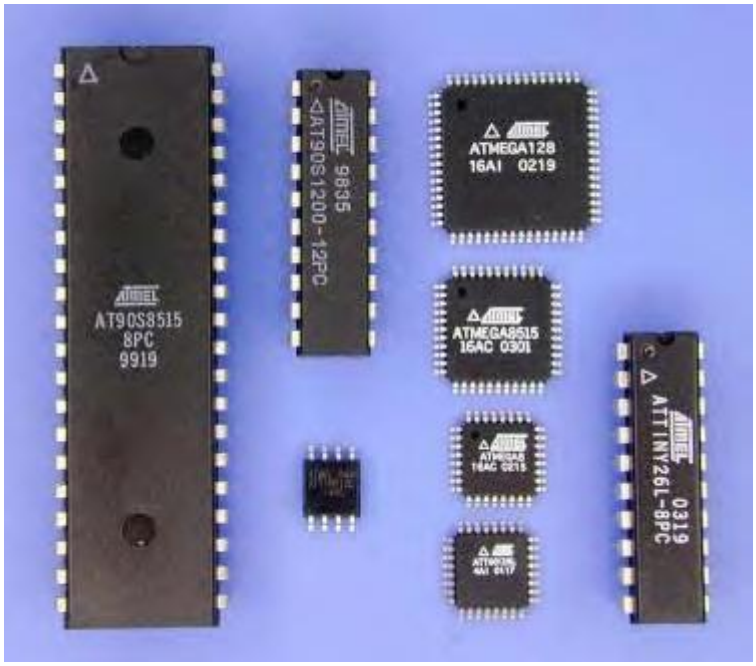
AVR ها میکروکنترولرهای 8 بیتی از نوع CMOS با توان مصرفی پایین هستند که بر اساس ساختار پیشرفته RISC ساخته شدهاند. پس از ساخت اولین نسخه های AVR در سال ۱۹۹۶ ، این سری از میکروکنترولرها توانست نظر علاقه مندان الکترونیک را به خود جذب کند به طوری که امروزه یکی از پرمصرفترین انواع میکروکنترولرها به حساب میآید. همان طور که می دانید نمیتوان هیچ نوع میکروکنترولری را به عنوان بهترین معرفی کرد چرا که هر میکروکنترولر ، کاربرهای خاص خود را دارد و بر اساس خصوصیات داخلی، میتواند تنها برای موارد ویژهای به عنوان بهترین انتخاب گردد، ولی با این حال با مطالعهٔ صفحات بعدی و آشنایی با امکانات و نرم افزارهای جانبی AVR متوجه خواهید شد که در کل استفاده از AVR بر بقیه ترجیح دارد.

انواع میکروکنترولرهای AVR

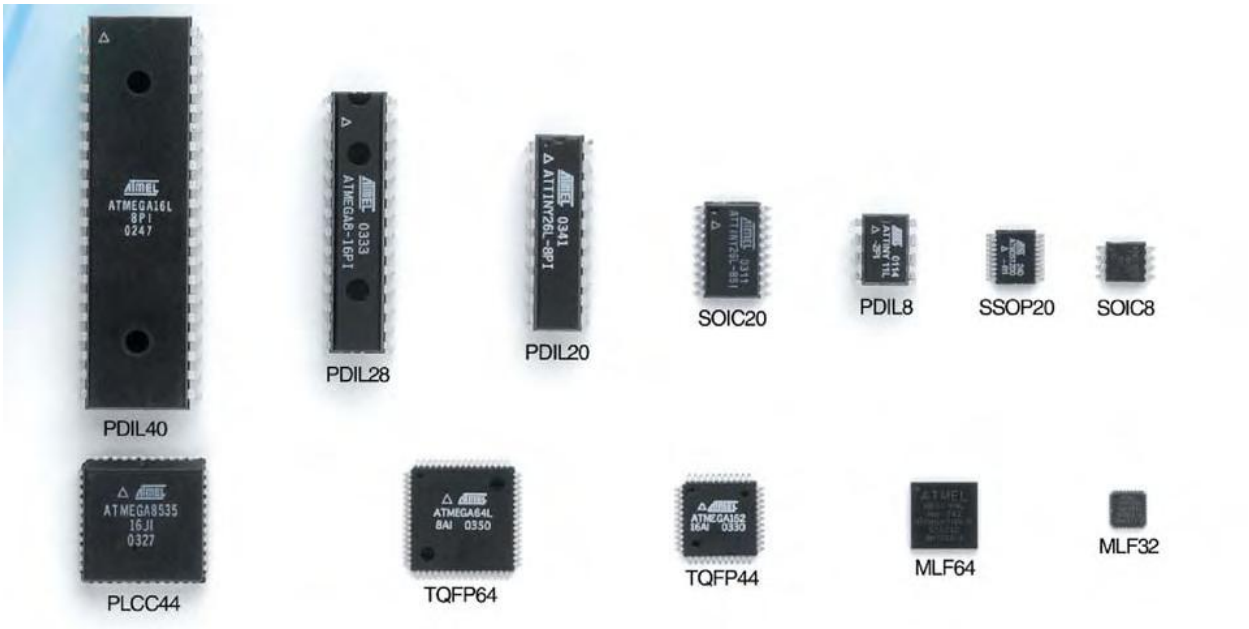
این گروه خود دارای 5 زیر مجموعه میباشد ، زیر مجموعه ها عبارتند از: سری Tiny Avr ، سری 90s ، سری Mega ، سری Lcd Avr و سری Xmega . در هر زیر مجموعه میکروهای زیادی وجود دارد مثلا میکروکنترلر های atmega 8 و atmega 16 از گروه میکروکنترلرهای سری mega هستند.

AVRها با ساختار RISC ، دستورات را تنها در یک پالس ساعت اجرا مینمایند و به این ترتیب میتوانیم تا به ازای هر یک مگاهرتز، یک مگادستور را در ثانیه (MIPS) اجرا کرده و برنامه را از لحاظ سرعت پردازش و نیز مصرف توان بهینه کنیم. AVRها ۳۲ رجیستر همه منظوره (R0...R31) و مجموعه دستورات قدرتمندی را شامل میگردند .تمام این 32 رجیستر مستقیمًا به ALU متصل شدهاند، بنابراین دسترسی به دو رجیستر در یک سیکل ساعت هم امکانپذیر است .این ساختار موجب میگردد تا سرعت آنها نسبت به میکروکنترولرهای CISC بتواند تا ۱۰ برابر هم افزایش یابد.

خانواده میکروکنترولرهای AVR ، تراشه هایی پیشرفته با امکانات جانبی کامل هستند .زمانی که شروع به یادگیری مفاهیم اصلی آنها نمایید، لذت فراگیری تمام جزئیاتشان آغاز میشود.



شکل3_1.انواع میکروکنترلرهای AVR



شکل3_2.انواع مختلف میکروکنترولرهای AVR از 8 پایه تا 16 پایه

روش های برنامه ریزی AVR

دو روش مختلف جهت برنامه ریزی تراشه های AVR وجود دارد که برنامه ریزی موازی و سریال (حالت ISP) نامیده میشوند .در حالت موازی تراشهٔ موردنظر در سوکت پروگرامر قرار داده میشود .در این روش لازم است تا ولتاژ ۱۲ + ولت به پایه RESET اعمال گردد. ارتباط بین پروگرامر و تراشه موردنظر نیز همان طور که از اسمش پیداست به کمک دستورات برنامه نویسی موازی برقرار میشود و به همین دلیل سرعت برنامه ریزی در این روش دو برابر سریعتر از حالت ISP است.

برخلاف روش ISP که بعضی از انواع AVR از آن پشتیبانی نمیکنند، روش برنامه ریزی موازی میتواند برای انواع AVR به کار برده شود بنابراین به طور کلی از این روش در مواردی که تولید انبوه موردنظر است استفاده میگردد.

امکانات کلی یک AVR

- در حدود ۱۳۰ دستور که اکثر آنها در یک سیکل ساعت اجرا میشوند.
- ۳۲ رجیستر ۸ بیتی همه منظوره.
- ضرب کنندهٔ سخت افزاری با زمان اجرای ۲ سیکل ساعت.
- دارای سه نوع حافظهٔ FLASH (برای کدهای برنامه) ، EEPROM ، SRAM .
- برنامهریزی تراشه در داخل مدار موردنظر بدون احتیاج به پروگرامر (ISP).
- حفاظت از کدهای برنامه در مقابل خواندن.
- قابلیت تنظیم نوسانگر برای کار توسط کریستال خارجی، کریستال فرکانس پایین خارجی، نوسانگر RC خارجی، نوسانگر RC داخلی و فرکانس خارجی.
- مجهز به پروتکلJTAG برای انجام عمل دیباگ، تست واسکن کردن وسایل جانبی تراشه و نیز برنامه ریزی حافظه های EEPROM ، FLASH و فیوزها.
- شمارنده و تایمر ۸ بیتی.
- شمارنده و تایمر ۱۶ بیتی.
- RTC با نوسانگر جداگانه.
- کانالهای PWM (با استفاده از تایمرها به صورت ۸ بیتی و ۱۶ بیتی برای تولید PWM)
- امکان تنظیم تایمر به صورت CTC
- ADCهای ۱۰ بیتی با یک ورودی و یا ورودی تفاضلی با بهره ی قابل تنظیم 1،10،200 .

- مجهز به پروتکل I2C یا TWI ، ارتباط دو سیمه از شرکت Philips .
- ارتباط سریال USART با قابلیت برنامه ریزی .
- ارتباط سریال SPI به صورت Master/Slave .
- تایمر نگهبان ، قابل برنامه ریزی با نوسانگر مجزا.
- مقایسه کنندهٔ آنالوگ با امکان تعریف وقفه برای آن.
- RESET شدن در زمان اتصال به برق.
- Brown – out Detector با قابلیت برنامهریزی.
- منابع وقفه داخلی و خارجی.
- با شش حالت مختلف برای کاهش توان مصرفی.
- کار با ولتاژهای 4،5-5،5 در مدل های بدون پسوند L مثل ATMega32 ، و 2،7 – 5،5 در مدلهای با پسوند L مثل ATMega 32(L) .
- ممکن است بعضی از امکانات ذکر شده در بعضی از انواع ساده AVR وجود نداشته باشد و حجم فضاهای Eeprom ، Sram ، Flash هم در آنها متفاوت باشد.

معایب یک AVR

- از آنجا که این میکرو برای مصارف عمومی ساخته شده است (مانیتورینگ ، کنترل غیر صنعتی و...) استفاده از آن در موارد صنعتی مشکلاتی را بوجود میاورد.

فصل چهارم:

میکروکنترلرهای ARM و PIC

میکروکنترلر PIC

واقعا میکروکنترلر خیلی قوی است که بر اساس بعضی آمار ها بیشترین کاربر را به خود اختصاص داده است البته در ایران این آمار به نفع AVR است. این میکروکنترلر ساخت شرکت میکرو چیپ است که PIC رو در مدل های خیلی زیادی با امکانات مختلف برای کارهای مختلف میسازد . این میکروکنترلر با مدل های مختلف PIC16XXX و PIC12XXXX که به جای X دوم از چپ به راست حروف C ,X,E,F قرار میگره که هر کدام مفهوم خاصی داره Xهای بعدی هم اعدادی هستنند که نشان دهنده مدل های مختلف هستنند .

این گروه دارای زیر مجموعه های زیادی میباشد و شما میتوانید با توجه به نیاز خود بهترین نمونه را انتخاب کنید.

ویژگی های PIC:

- دارای امکاناتی همچون adc ، تایمر ، کانتر ، pwm ، نوسان ساز داخلی و...
- پشتیبانی از پروتکل های 1wire ، rs232 ، spi ، i2c
- تنوع در نمونه
- کامپایلرها ،دیباگرها و شبیه سازهای مختلف
- کاربرد گسترده در موارد عمومی و صنعتی (نویز روی این میکرو تاثیر کمی دارد)

معایب یک PIC

- قیمت بالا
- منابع فارسی زیادی ندارند
- بعضی از نمونه ها در ایران موجود نیست.

میکروکنترلر ARM

این خانواده توسط شرکت های زیادی تولید میگردد اما محصولات شرکت اتمل در ایران بیشتر به چشم میخورند

از مزیت های اصلی این میکرو ، سرعت بالا در انجام محاسبات و پشتیبانی از پروتکل های ارتباطی میباشد

ویژگی های ARM

- دارای امکاناتی همچون adc ، تایمر ، کانتر ، pwm ، نوسان ساز داخلی ، pll ، خطوط io زیاد
- پشتیبانی از پروتکل های sam_ba ، i2s ، lan ، can ، usb ، 1wire ، rs232 ، spi ، i2c و...
- تنوع در نمونه
- سرعت بسیار بالا و کار در فرکانس تا 2 گیگا هرتز
- کاربرد گسترده در موارد عمومی و صنعتی و نظامی و...

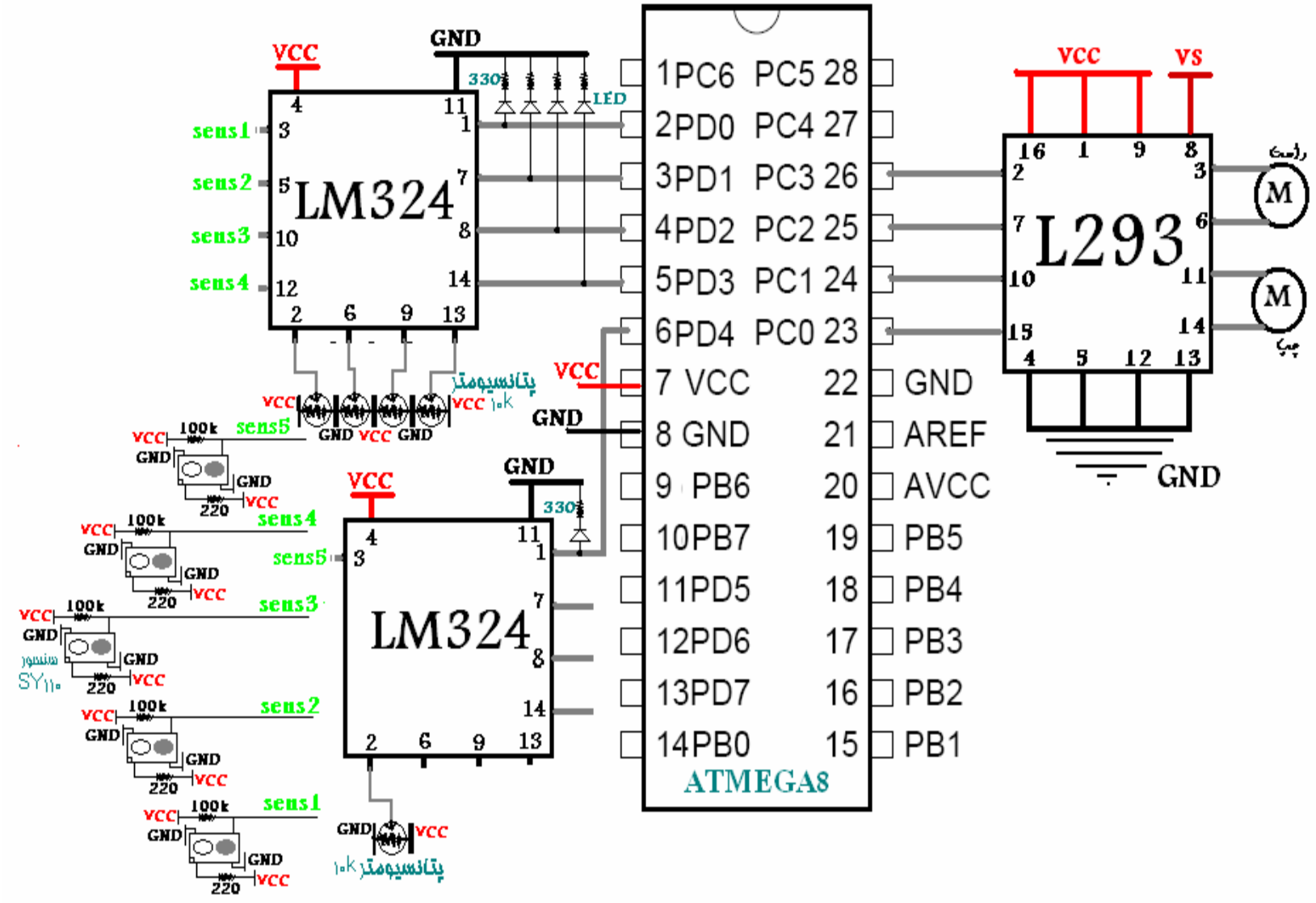
معایب یک ARM

- قیمت بالا
- منابع فارسی زیادی ندارد
- بعضی از نمونه ها در ایران موجود نیست
- کامپایلرها و شبیه سازهای کمی دارد.

فصل پنجم:

کاربرد میکروکنترلر در ربات

کاربرد میکروکنترلر در روبات تعقیب خط


VCC
GND
LM324
sens1
sens2
sens3
sens4
sens5
330
LED
پتانسیومتر
10k
100k
220
سنسور
SY۱۱۰
ATMEGA8
PC6 PC5 PC4 PC3 PC2 PC1 PC0
PD0 PD1 PD2 PD3 PD4
PB6 PB7 PD5 PD6 PD7 PB0
AREF AVCC PB5 PB4 PB3 PB2 PB1
L293
VS
M
راست
چپ

برنامه کامل روبات تعقیب خط با 5 سنسور

```
/*****************************************************
by mohammad javad fotuhi
*****************************************************/
#include <mega8.h>
// Declare your global variables here
unsigned char sensd;
//**********************
forward (void)
{
PORTC=0b001010;
}
//**********************
left (void)
{
PORTC=0b001001;
}
//**********************
right (void)
{
PORTC=0b000110;
}
//**********************
void main(void)
{
```

```
// Declare your local variables here
// Input/Output Ports initialization
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;
// Port C initialization
// Func6=In Func5=In Func4=In Func3=Out Func2=Out Func1=Out Func0=Out
// State6=T State5=T State4=T State3=0 State2=0 State1=0 State0=0
PORTC=0x00;
DDRC=0x0F;
// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
TCCR0=0x00;
TCNT0=0x00;
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FORWARDFORWARDh
```

```
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Oforward
// Input Capture on Falefting Edge
// Timer 1 Overflow Interightupt: Oforward
// Input Capture Interightupt: Oforward
// Compare A Match Interightupt: Oforward
// Compare B Match Interightupt: Oforward
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FORWARDh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
```

```
TCNT2=0x00;
OCR2=0x00;
// External Interightupt(s) initialization
// INT0: Oforward
// INT1: Oforward
MCUCR=0x00;
// Timer(s)/Counter(s) Interightupt(s) initialization
TIMSK=0x00;
// Analog Comparator initialization
// Analog Comparator: Oforward
// Analog Comparator Input Capture by Timer/Counter 1: Oforward
ACSR=0x80;
SFIOR=0x00;
while (1)
{
// Place your code here
sensd=0b00011111;
sensd=PIND&sensd;
if(sensd==0b00000100)
forward();
if(sensd==0b00001110)
forward();
if(sensd==0b00001100)
right();
if(sensd==0b00001000)
```

```
right();
if(sensd==0b00010000)
right();
if(sensd==0b00011000)
right();
if(sensd==0b00011100)
right();
if(sensd==0b00000001)
left();
if(sensd==0b00000011)
left();
if(sensd==0b00000111)
left();
if(sensd==0b00000110)
left();
if(sensd==0b00000010)
left();
//*****************************TAGHIR RANG
if(sensd==0b11111011)
forward();
if(sensd==0b11110001)
forward();
if(sensd==0b11110011)
right();
if(sensd==0b11110111)
```

```
right();
if(sensd==0b11101111)
right();
if(sensd==0b11100111)
right();
if(sensd==0b11100011)
right();
if(sensd==0b11111110)
left();
if(sensd==0b11111100)
left();
if(sensd==0b11111000)
left();
if(sensd==0b11111001)
left();
if(sensd==0b11111101)
left();
};
}
```