

- هر ربات شامل سه بخش :
- ۱- نرم افزار
- ۲- سخت افزار(الکترونیکی)
- ۳- مکانیکی
- است .

نرم افزار

تهیه برنامه ای که برای انجام محاسبات توسط پردازنده استفاده می شود ، جزو وظایف مسئول نرم افزار است. همچنین از دیگر وظایف مسئول نرم افزار می توان به شبیه سازی قسمت های مختلف ربات و تست آن ها اشاره کرد.

سخت افزار

مسئول این قسمت وظیفه دارد مدار های الکترونیکی ربات را طراحی و اجرا کند.

مکانیکی

طراحی قسمت حرکتی و ابزار های عکس العمل ربات (موتور، بازو و ...) توسط مسئول این قسمت انجام می شود.

قسمت سخت افزار یک ربات متناسب با کار هایی که از آن انتظار داریم می تواند شامل :

۱- حسگر و مبدل آنالوگ به دیجیتال (ورودی)

۲ - پردازنده(پردازش)

۳- موتور، بازو و... (خروجی)

باشد .

حسگرها

هر ربات متناسب با کاری که می خواهد انجام دهد به مقادیری نیاز پیدا خواهد کرد که دریافت آن مقادیر باید توسط حسگر ها انجام شود.

هر حسگر کمیت مورد نظر را به ولتاژ، عدد و... تبدیل می کند. ولی در بعضی مواقع حسگر عددی غیر رند به ما می دهد یا اینکه عدد تولید شده توسط حسگر متغیر است در چنین مواقعی اگر پردازنده ما می تواند از منطق فازی (Fuzzy Logic) استفاده کند مشکلی پیش نمی آید ولی اگر پردازنده فقط ۰ یا ۱ بتواند بگیرد باید از یک مبدل آنالوگ به دیجیتال (Analog to Digital Convertor) استفاده کرد. (در بخش آموزش الکترونیک در مورد ADC بیش تر بحث خواهد شد.)

پردازنده ها

وظیفه ی پردازنده ها دریافت ورودی ، انجام محاسبات و دادن خروجی است. محاسبات در پردازنده ها توسط انسان تعریف می شود.

موتور ها، بازوها و ...

خروجی پردازنده ها به صورت ۰ و ۱ است ولی ما از ربات انتظار داریم عملی را انجام دهد. این عمل توسط موتور ها، بازوها و چیزها دیگر انجام می شود.

ربات تعقیب خط

یک ربات تعقیب خط نیاز به سنسور هایی دارد که خط سیاه یا سفید را تشخیص دهند. همچنین به یک ADC نیاز دارد. خروجی های ADC نیز به یک میکرو کنترلر می روند و در میکرو کنترلر توسط برنامه پردازش و خروجی به دو عدد موتور متصل می شود.

۱- سخت افزار(الکترونیکی)

قسمت الکترونیکی این ربات شامل :

۱- سنسورها (فرستنده و گیرنده ها)

۲- تبدیل گر آنالوگ به دیجیتال

۳- میکرو کنترلر

۴- موتورها

۱-۱- سنسور ها

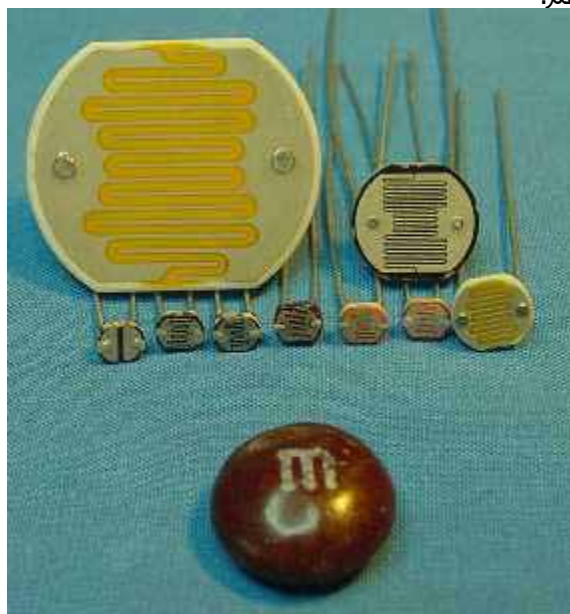
سنسور های این ربات به دو دسته تقسیم می شوند :

۱- حساس به نور(اپتیک)

۲- مادون قرمز

۱-۱-۱- سنسور های نوری

این دسته از سنسور ها از يك LED و يك مقاومت حساس به نور تشكيل مي شود. نور توسط LED تابیده مي شود و نور بازتاب شده باعث تغيير مقاومت و در نتیجه تغيير ولتاژ مي شود(همانطور که مي دانيد بازتاب سطح مشكي کمتر از سطح سفيد است). اینگونه از سنسور ها به دليل اثر پذيري از نور محیط ، درصد خطاي بسياري دارند . به همین دليل خيلي در مورد این سنسور ها توضيح نمي دهيم.



سنسور اپتيك(يك مقاومت حساس به نور)

۱-۱-۲- سنسور هاي مادون قرمز

این گونه از سنسور ها از يك فرستنده و يك گيرنده مادون قرمز تشكيل شده است. پرتو هاي مادون قرمز توسط فرستنده فرستاده شده و به سطح برخورد مي کند و بازتاب آن توسط گيرنده دريافت مي شود. اینگونه از سنسور ها به دليل اینکه از محیط اثر نمي گيرند، درصد خطا بسيار پاييني دارند. بنابر این ما در مورد این سنسور ها صحبت مي کنيم. سنسور هاي مادون قرمز به دو دسته تقسيم مي شوند. دسته اول LED فرستنده و گيرنده جدا از هم است. این گونه از سنسور ها در دو اندازه 5mm و ۳mm موجود است . نوع 3mm به دليل اینکه LED فرستنده و گيرنده هم رنگ هستند تشخيص آن ها سخت است و بايد توسط مالتی متر انجام پذيرد. مشکل ديگر این سنسور ها این است که نوع تقلبي آن نیز بسيار زياد است و امکان دارد از بين ۵ سنسور ۱ يا ۲ تا خراب باشد. گونه ي ديگر سنسور هاي مادون قرمز به صورتي است که فرستنده و گيرنده با هم در يك بسته قرار دارند . این گونه از سنسور ها داراي اندازه ي بسيار کوچک و دقت خوبي هستند.



LED فرستنده مادون قرمز

يکي از گونه هاي قوي سنسور هاي مادون قرمز سري TSL است . سنسور هاي این سري بستگی به نوع خود قدرت نسبتاً زيادي دارند. نوع ديگري از سنسور هاي مادون قرمز سري RS است . این سري داراي اندازه ي کوچکي است مشکل بزرگ این سنسور ها این است که بسيار حساس هستند و با بالا رفتن ولتاژ به سرعت مي سوزند.

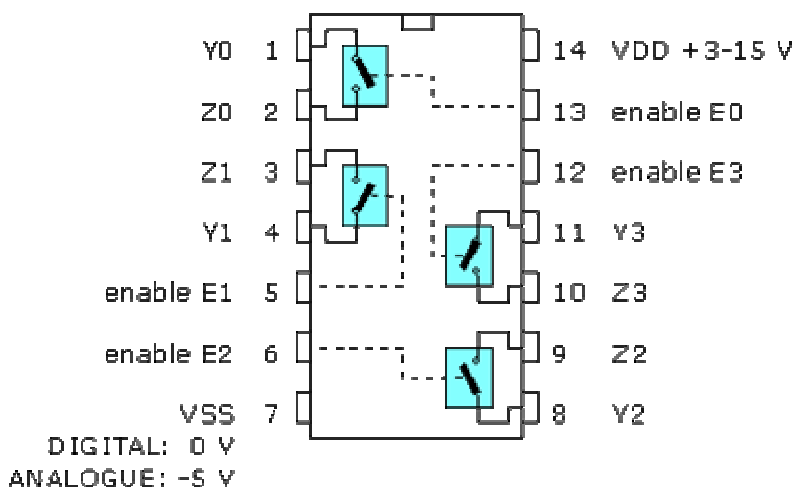
۱-۲- مبدل آنالوگ/ديجیتال

براي تبديل آنالوگ به ديجیتال چند راه وجود دارد :
استفاده از ADC (Analog to Digital Converter)
استفاده از OpAmp ها

اگر از میکرو کنترلر ها PIC يا AVR استفاده مي کنيد مي توانيد از مبدل داخلي این میکروکنترلر ها استفاده کنيد.

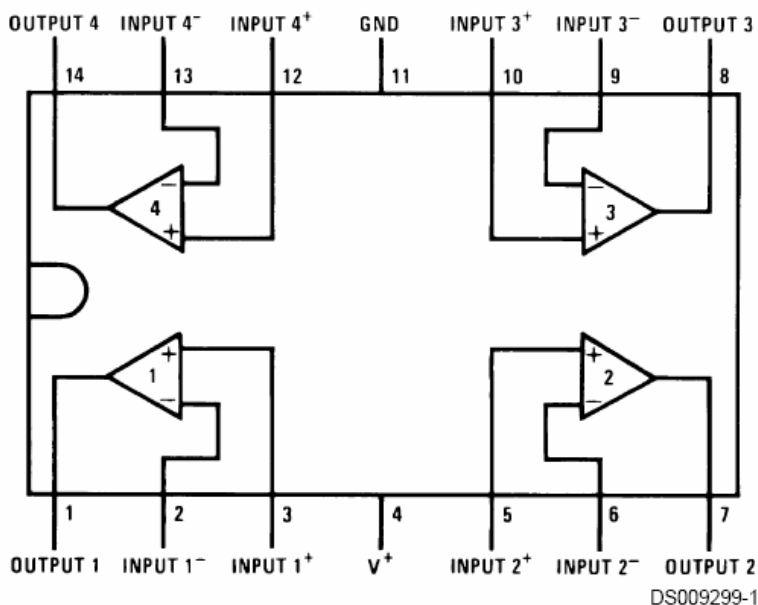
۱-۲-۱ (Analog to Digital Converter) ADC

این مبدل ها می توانند هر گونه موج سینوسی را به موج مربعی تبدیل کنند . برای مثال می توان از CD4016 نام برد.



Op Amp - ۱-۲-۲

این گونه آی سی ها مقایسه گر هستند و عمل تقویت را نیز می توانند انجام دهند . با استفاده از خاصیت مقایسه گری این آی سی ها می توان موج مربعی تولید کرد . به این صورت که ولتاژی را معین می کنیم . اگر ولتاژ ورودی از این ولتاژ کمتر بود ، ولتاژ خروجی صفر و اگر بالا تر بود ولتاژ خروجی ۵ ولت می گردد. متداول ترین OpAmp ، ای سی LM324 است که دارای چهار واحد OpAmp است. نوع دیگر این آی سی ، آی سی LM393 است که دارای دو واحد OpAmp است.



LM324

۱-۲-۱ میکرو کنترلر ها

میکرو کنترلر ها مغز ربات شما هستند . ماک شما برای انتخاب میکروکنترلر باید تعداد پورت های ورودی و خروجی و امکانات جانبی مثل PWM و ADC باشد.

سه خانواده میکرو کنترلر ها عبارتند از :

PIC - ۱

8051 - ۲

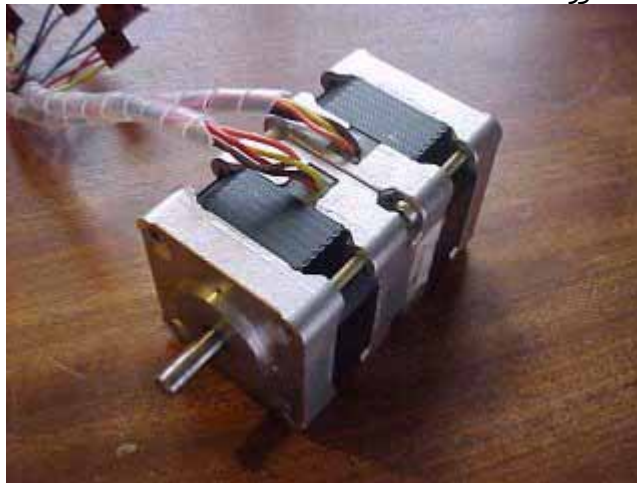
AVR - ۲

من در مورت ۸۰۵۱ توضیح می دهم. چون راحت ترین میکروکنترلر برای شروع است . البته نه در مورد برنامه نویسی. راحت ترین میکروکنترلر از نظر نرم افزاری ، میکروکنترلر PIC است. ولی من با ۸۰۵۱ کار کردم و خیلی در مورد PIC اطلاعات ندارم.

۱-۲-۱- موتور ها
موتور ها بر دو نوع هستند:
۱- Stepper Motor
۲- DC Motor

۱-۲-۱-۱- استپر موتور ها

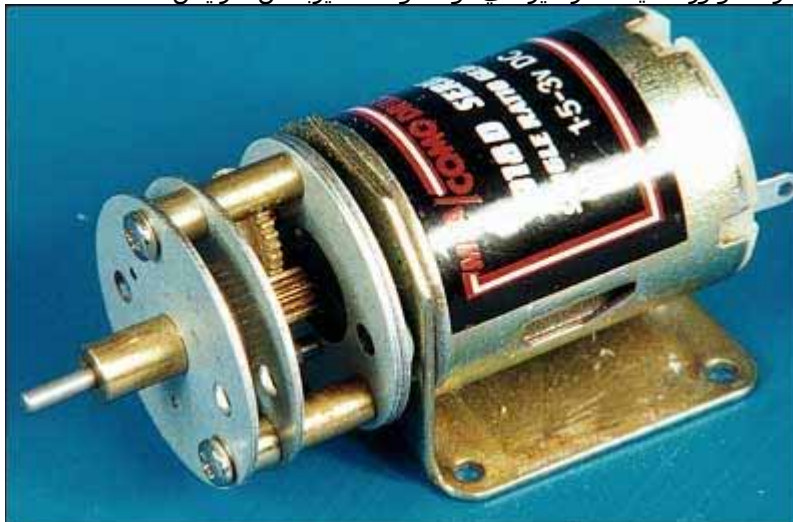
استپر موتور ها خود نیز بر اساس تعداد سیم پیچ های داخل خود و اینکه با هر استپ (هر حرکت موتور) چند درجه می چرخد به انواع متفاوتی تقسیم می شوند. درایو کردن استپر موتور ها سخت تر از موتور های DC است ولی قدرتی و دقت استپر موتور ها بیشتر است. مثلاً می توان با محاسبه تعداد استپ و اندازه هر استپ بر حسب درجه و ضرب آن در محیط چرخ، مقدار حرکت ربات را به دست آورد.



دو عدد استپر موتور

۱-۲-۱-۲- موتور های DC

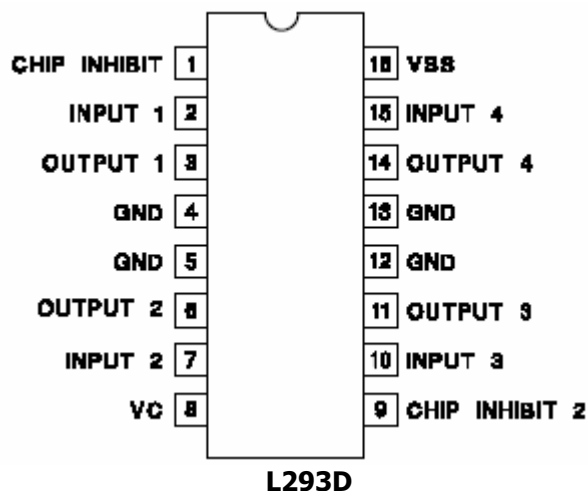
این موتور ها راحت درایو می شوند ولی دقت ندارند ولی در ربات تعقیب خط چون دقت موتور مهم نیست بهتر است از این موتور ها استفاده شود. قدرت موتور های DC را نیز می توان توسط گیربکس افزایش داد.



یک موتور DC با گیر بکس

۱-۲-۱-۳- درایور ها

موتور ها به دلیل جریان کم خروجی میکرو کنترلر نمی توانند به طور مستقیم به میکرو کنترلر متصل شوند. بنابراین خروجی میکرو کنترلر به یک واحد منطقی به نام درایور وارد می شود و خروجی درایور به موتور متصل می شود. راه دیگری که وجود دارد این است که از خاصیت کلید زنی ترانزیستور ها برای کلید زنی بین صفر و ولتاژ مورد نیاز موتور استفاده کرد. چند درایور متداول عبارتند از L293, L293D, L298 تفاوت بین L293 و L293D این است که L293D دارای دیود تخلیه موتور است.



برنامه نویسی میکرو کنترلر های ۸۰۵۱

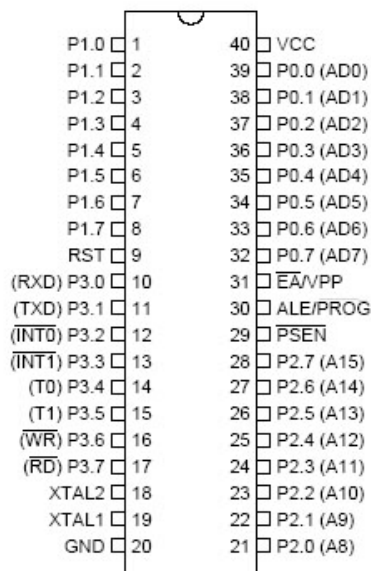
برای این میکرو کنترلر ها می توان با دو زبان C و ASM51 برنامه نوشت. (ASM51 اسمبلی برای میکرو کنترلر ۸۰۵۱ است.) سپس باید فایل برنامه را تبدیل به یک فایل HEX کرده و بر روی میکرو کنترلر اسمبل کرد. البته بعضی از پروگرامر ها با برنامه هایی کار میکنند که خودشان کار تبدیل فایل برنامه به HEX را انجام می دهند. اگر می خواهید هزینه خرید پروگرامر را ندهید لینک زیر می تونه بهتون کمک کنه (پروگرامر+برنامه):

<http://www.kmitl.ac.th/~kswichit/IspPgm30a/ISP-Pgm30a.html>

اینم یک ربات تعقیب خط ساده:

<http://www.kmitl.ac.th/~kswichit/LFrobot/LFrobot.htm>

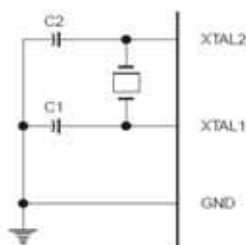
میکرو کنترلر ۸۰۵۱



پایه های میکرو کنترلر ۸۰۵۱

اسیلاتور

هر میکرو کنترلر و به طور کلی هر پروسیسور (پردازنده) نیاز به یک اسیلاتور (پالس دهنده) دارد. مدار پالس دهنده اینگونه است:



در نتیجه برای راه اندازی مدار اسیلاتور نیاز به دو عدد خازن عدسی ۳۰ یا ۳۳ پیکوفاراد و یک کریستال نیاز دارید. کریستال ۸۰۵۱ می تواند تا ۱۲ مگاهرتز باشد. ولی بهتر است از کریستال ۱۱,۰۵۹۲ استفاده شود. زیرا خطای کمتری را در میکروکنترلر به وجود می آورد .

ریست

میکرو کنترلر هم مانند کامپیوتر گه گاه به سرش می زند و نیاز به ریست دار که این مدار این نیاز را رفع می کند. البته ساده ترین راه قرار دادن یک کلید فشاری بین GND (یا همان ولتاژ صفر) و پایه ریست در میکروکنترلر است که با وصل شدن این دو نقطه عمل ریست انجام می شود.

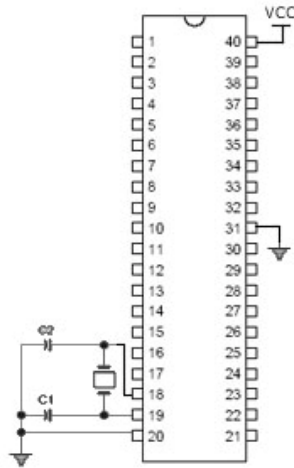
پایه EA

این پایه در صورتی که نخواهیم از حافظه جانبی استفاده کنیم باید به GND متصل باشد .

پورت P0

این پورت به تفاوت جزیی با دیگر پورت ها دارد و آن عدم وجود مقاومت پول آپ در مدار داخلی آن است. اگر بخواهیم از این پورت استفاده کنیم باید به هر یک از پین های آن یک مقاومت ۱۰ اهم متصل کرده و مقاومت ها را به ولتاژ ۵ ولت وصل کنیم.

یک میکرو کنترلر آماده به خدمت!



خب میریم سراغ کار با میکرو کنترلر
اگر می خواهید با زبان اسمبلی میکروکنترلر را برنامه نویسی کنید می توانید از برنامه فرانکلین Franklin استفاده کنید.

اول اصطلاحات:

VCC=+5

GND=0

RST=RESET

خواند/نوشتن پورت

برای این کار باید ابتدا یک بار (در یک سیکل) پورت مورد نظر را یک کنیم:

```
MOV A,#0FFH
MOV P0,A
```

اگر این عمل صورت نگیرد نمی توان از پورت ورودی بخوانیم.

برای اینکه به کم با میکروکنترلر کار کنیم و اثر دستور خواندن و نوشتن را ببینیم می توانیم از برنامه زیر استفاده کنیم:

```
MOV A,#0FFH
MOV P0,A
MOV A,P0
MOV P1,A
```

این دستورات باعث می شود ورودی را از پورت صفر بگیرد و در پورت یک بنویسد . برای دیدن نتیجه باید ۸ LED به پورت یک وصل کرده و ولتاژ مثبت (+۵) یا ولتاژ منفی (۰) به پورت یک اعمال کنید. (ولتاژ ورودی بستگی به این دارد که پایه مثبت یا منفی LED به پورت وصل شده باشد و همونطور که نمی دونید باید پایه دیگر را به GND یا VCC وصل کرد)

تایمر در میکروکنترلر ۸۰۵۱

ثبات های تایمر در میکرو کنترلر ۸۰۵۱ عبارتند از :

TMOD: این ثبات حالت تایمر را مشخص می کند

TCON: این ثبات تایمر را کنترلر می کند

خب می پردازیم به توضیح هر یک از این موارد:

TMOD:

این ثبات ۸ بیتی است (چه جلب!) . بیت های ۰ تا ۳ مربوط به تایمر ۱ و بیت های ۴ تا ۷ برای تایمر ۰ است.

برای مقدار دادن به این بیت باید از دستور MOV استفاده کنیم. (اگه آموزش های اسمبلی رو دانلود کرده باشید می دونید

چه جوری!)

یک مثال:

```
MOV TMOD,#01H
```

این دستور حالت تایمر را ۱۶ بیتی قرار می دهد.

| عملگر تایمر | | |
|-------------|----|----------------------------------|
| M1 | M0 | |
| 0 | 0 | تایمر ۱۳ بیتی |
| 0 | 1 | تایمر ۱۶ بیتی |
| 1 | 0 | تایمر ۸ بیتی با بار شدن اتوماتیک |

| | | |
|---|---|---|
| 1 | 1 | تایمر دو قسمتی: • تایمر ۰ دو قسمت می شود (دو تایمر مجزای ۸ بیتی TLO و TH0) • تایمر ۱ متوقف می شود |
|---|---|---|

: TCON

با این ثابت می توان تایمر های ۰ و ۱ را کنترل کرد مثلا آن ها را راه انداز یا متوقف کرد.
 یک مثال برای راه اندازی تایمر:

SETB TR0

این دستور تایمر ۰ را راه اندازی می کند.

JNB TF0,MOTOR

این دستور منتظر می شود تا تایمر سرریز شود. سپس به برچسب MOTOR پرش می کند.

| شماره بیت | سمبول بیت | عملکرد |
|-----------|-----------|---|
| 6,4 | TR1,TR0 | اگر این بیت ۱ شود تایمر راه اندازی می شود و اگر ۰ شود تایمر متوقف می شود. |
| 7,5 | TF1,TF0 | بیت سرریز تایمر. با پر شدن تایمر ۱ می شود. |
| 3,1 | IE1,IE0 | بیت پرچم حساس به لبه (برای وقفه است فعلا کاری باهاش نداریم) |
| 2,0 | IT1,IT0 | بیت پرچم نوع وقفه (برای وقفه است فعلا کاری باهاش نداریم) |

هر عدد در دارای دو قسمت کم ارزش و با ارزش است. برای مثال:
 FF FF

دو رقم اول کم ارزش و دو رقم بعد پر ارزش هستند.

برای تایمر هم همین حالت وجود دارد یعنی این امکان وجود دارد که فقط بیت های کم ارزش یا بیت های با ارزش را مقدار دهی کنیم.

این عمل نیز با دستور MOV انجام می گیرد و برای مشخص کرد بیت های با ارزش و کم ارزش از L و H استفاده می کنیم .
 برای مثال:

MOV TLO,#0FH
 MOV TH0,#0CH

برای مقدار دهی به تایمر می توان از دستور زیر نی استفاده کرد.

MOV T0,#0FCAH

برای محاسبه ثانیه ها اینگونه باید عمل کرد: ابتدا محاسبه می کنیم که مقدار پالس تایمر چقدر است. برای اینکار پالس میکروکنترلر را تقسیم بر ۱۲ می کنیم (مثلا برای یک میکروکنترلر با کریستال ۱۲ مگاهرتز: $12\text{MHz}/12=1\text{MHz}$). بعد از آن مقدار زمان بر حسب پالس را حساب می کنیم. سپس این عدد (زمان بر حسب پالس) را به مبنای هگزا دسیمال تبدیل کرده (توسط ماشین حساب ویندوز) و از مقدار سرریز تایمر (مثلا FFFF) کم می کنیم. در انتها عدد به دست آمده را در تایمر قرار می دهیم.

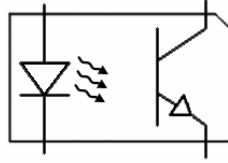
می دونم هیچی نفهمیدید(لا اقل خودم!) خب براتون یه مثال می زنم:

می خواهیم یک صدم ثانیه تاخیر ایجاد کنیم و اسیلاتور مدار هم ۱۲ مگاهرتز است.

در نتیجه هر پالس یه میلیونیم ثانیه طول می کشد(1µs). بنابراین باید ۱۰۰۰۰۰ پالس را تاخیر ایجاد کنیم. معادل این عدد در مبنای هگزا دسیمال : 2710

خب، این مقدار را از مقدار کل یعنی FFFF کم می کنیم. عدد به دست آمده عبارت است از: D8EF
 در انتها این مقدار را در تایمر قرار می دهیم.

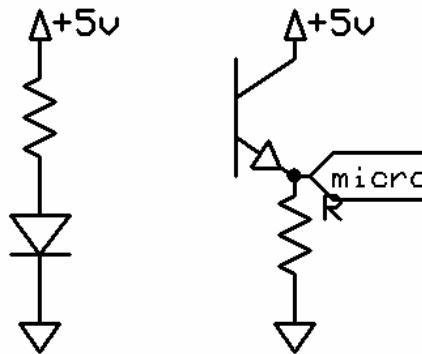
MOV T0,#D8EFH



(سری RS- به صورت یک)

همانطور که در شکل پیدا است سنسور از یک دیود و یک فوتوترانزیستور تشکیل شده است .
چگونگی مدار سنسور

همانگونه که می دانید این یک ترانزیستور NPN (منفی) است. بنابر این باید پایه کلکتور را به ولتاژ، پایه امیتر را نیز توسط یک مقاومت به GND وصل کرد. محل اتصال پایه امیتر و مقاومت ، خروجی ترانزیستور است. پایه مثبت دیود را هم باید توسط یک مقاومت به ولتاژ متصل نمود .



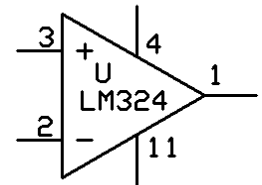
می توان به جای مقاومتی که به امیتر متصل شده است ، یک پتانسیومتر قرار داد، که بتوان حساسیت سنسور را تنظیم کرد.

چرا از ADC استفاده می کنیم؟

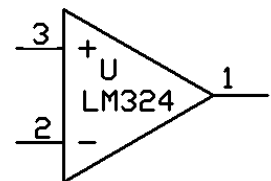
به دلیل اینکه ولتاژ خروجی سنسور ، آنالوگ است (VVVV) و میکروکنترلر می تواند فقط ورودی دیجیتال بپذیرد. بنابر این باید ولتاژ خروجی سنسور را به دیجیتال تبدیل کرد.

برای این کار از ADC ها یا OpAmp ها استفاده می کنیم. ADC ها برای تبدیل آنالوگ به دیجیتال طراحی شده اند ولی OpAmp ها هم برای مقایسه و تقویت طراحی شده اند. از خاصیت مقایسه کنندگی OpAmp برای تبدیل آنالوگ به دیجیتال استفاده می کنیم .

شکل یک واحد OpAmp همراه با پایه های تغذیه آی سی

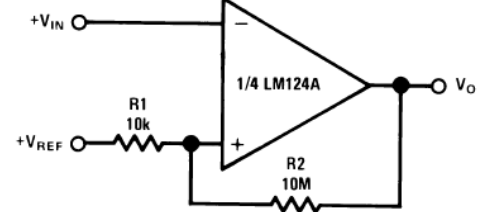


شکل یک واحد OpAmp



پایه ۱ : خروجی
پایه ۲ : ولتاژ ورودی
پایه ۳ : ولتاژ مقایسه

البته یک واح OpAmp برای اینکه بتواند عمل مقایسه را انجام دهد باید به صورت زیر بسته شود.



عمل OpAmp به این صورت است که اگر ولتاژ ورودی کمتر از ولتاژ مقایسه باشد ، ولتاژ پایه خروجی برابر با ولتاژ GND می شود. اگر ولتاژ ورودی بیشتر از ولتاژ مقایسه باشد ، ولتاژ خروجی برابر با ولتاژ تغذیه آی سی (VCC) می شود.