

مکاترونیک و روباتیک

روبات تعقیب خط با میکروکنترلر AVR
(ATmega8)

توسعه کننده: محمدرضا محمدی

رشته سفت افزار

بسم الله الرحمن الرحيم

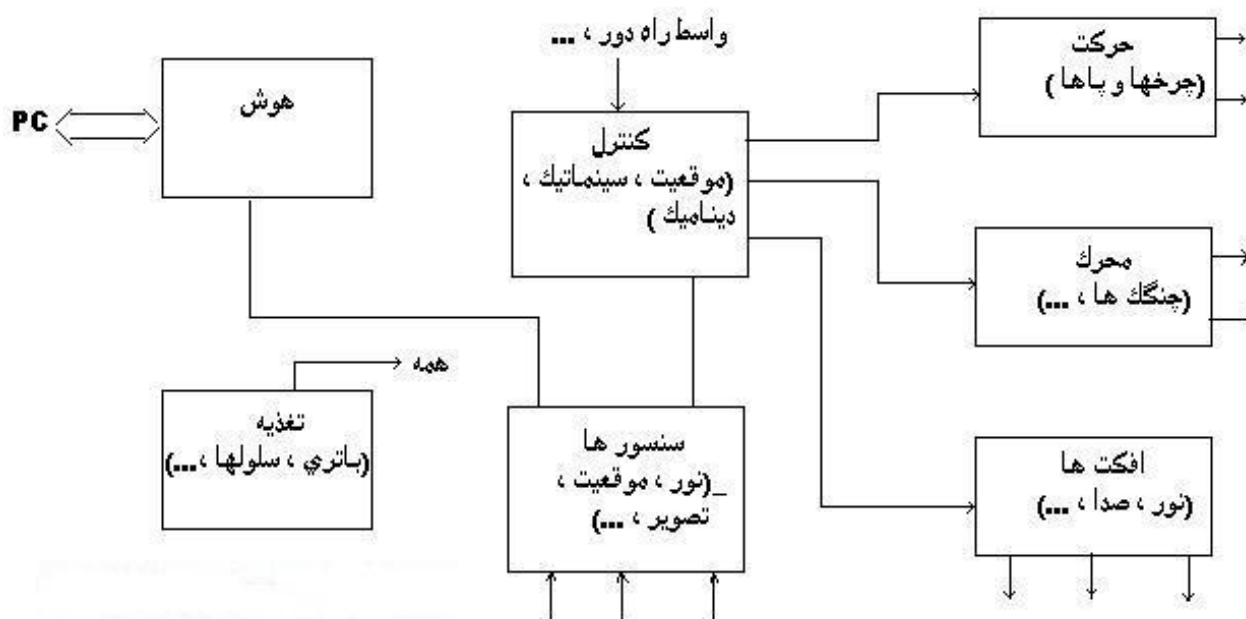
مکاترونیک، روباتیک و هوش مصنوعی:

در واقع به وسایل مکاترونیکی که بتوانند کارهای شبیه انسان را تکرار کنند و بتوانند کارهای تکراری و خسته کننده انسان را انجام دهند روبات گفته می شود .

روبات معادل کلمه کارگر در زبان چک می باشد.

از آنجایی که به هر وسیله ای نمی توان نام روبات نهاد بهتر است به این نکته توجه شود که هوش قسمت اصلی روبات است.

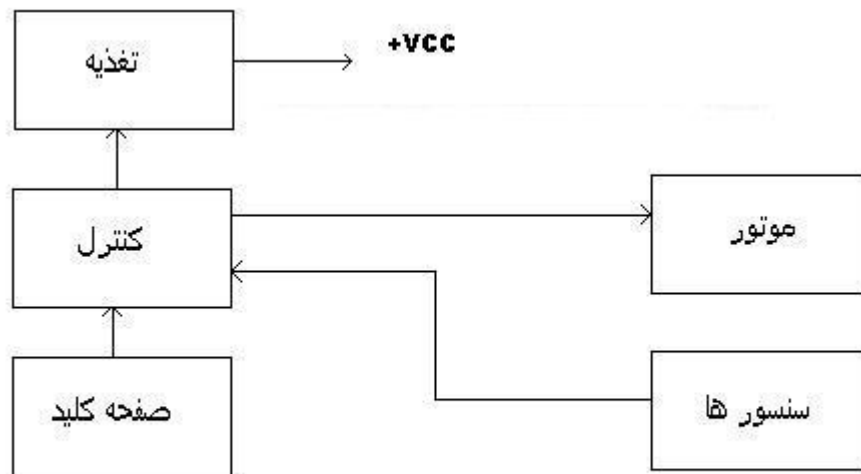
در حالت ساده یک روبات از قسمت های الکترونیک مکانیک و برنامه تشکیل می شود و ما با ترکیبی از آنها بلوک های زیر را تشکیل می دهیم.



ساختار اساسی یک پروژه در مکاترونیک و روباتیک

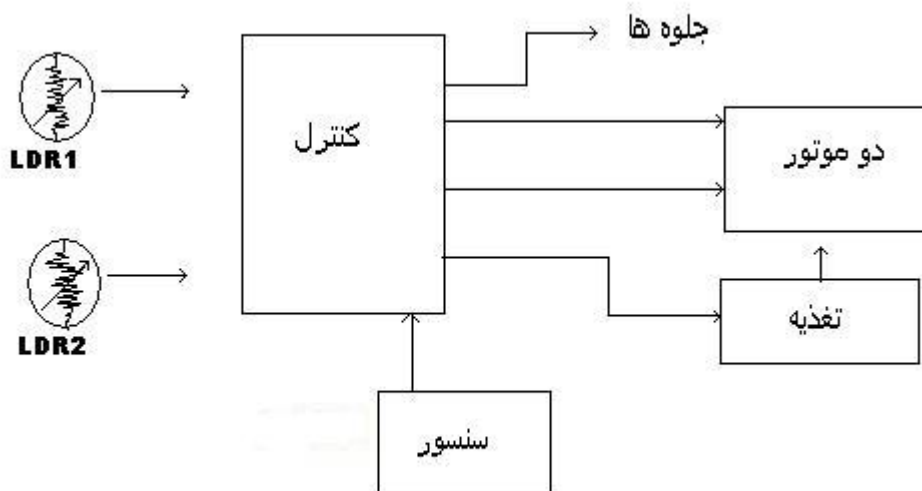
در کل پروژه های روباتیک از قسمت های بالا تشکیل می شوند اگر شما تازه کار هستید لازم نیست تمام این قسمت ها را پیاده کنید و حتی بعضی از روبات ها احتیاج به همه آنها ندارند .

مثلا يك بالا بر اتوماتيك به همه آنها احتياج ندارد , از قسمت های زیر تشكيل شده .



بلوك هاي سازنده آسانسور اتوماتيك

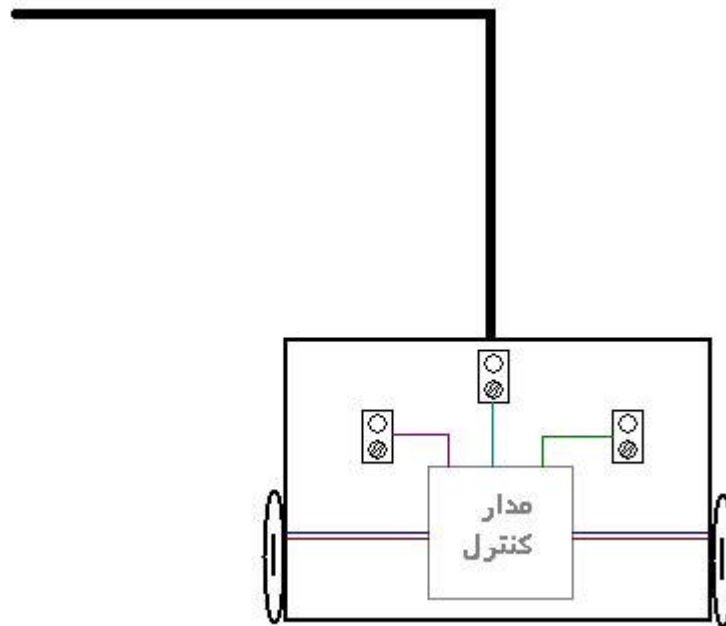
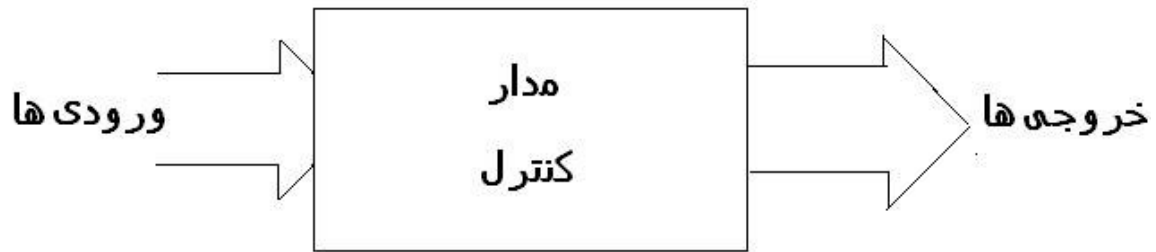
و يا يك روبات تعقيب کننده خط که دو تا موتور و سنسور دارد . فقط به قسمت های زیر احتياج دارد.



ساختار روبات تعقيب کننده خط

در چند سال اخیر مسابقات روباتیک در زمینه های روبات های تعقیب خط - جنگجو - میکروماوس - مین یاب - لابیرنت - آتش نشان و ... برگزار شده است. برای شروع بهتر است از روبات تعقیب خط شروع کنیم. روبات تعقیب خط:

روبات تعقیب باید بتواند مسیری مشکی رنگ به عرض 18 میلیمتر را در زمینه سفید و مسیری سفید رنگ به عرض 18 میلیمتر در زمینه سیاه دنبال کند و از بردگی ها و زاویه های مسیر عبور کند. تشخیص رنگ سیاه از سفید با سنسور های مادون قرمز است که در کف روبات کار گذاشته می شود. روبات با میکروکنترلر AVR و برنامه نویسی با زبان C (CODEVISION)



مسائل پایه:

تبدیل مبناها:

DEC	BINARY	OCTAL	HEX
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

2 → 10

10 → 2

2 → 8

16 → 2

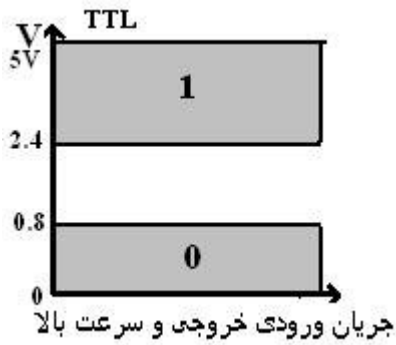
2 → 16

ولتاژ:

دیجیتال - آنالوگ

0 ← 0 ولت

1 ← 5 ولت

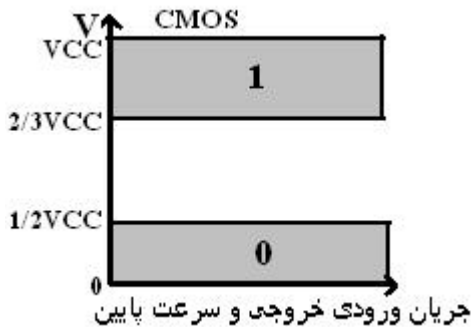


منطق 1 +5V

74***

54***

TTL های IC



3-15 ولت

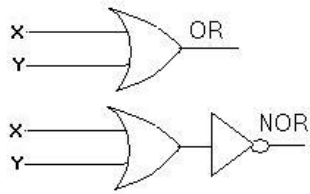
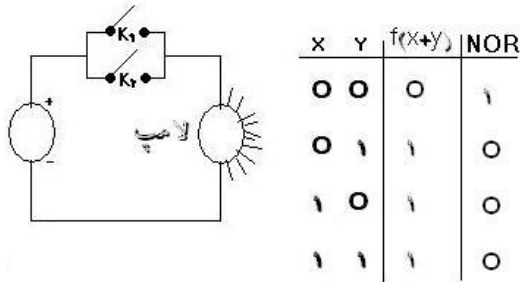
40***

45***

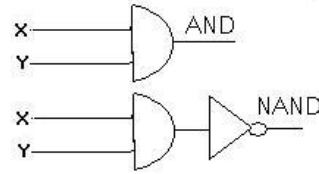
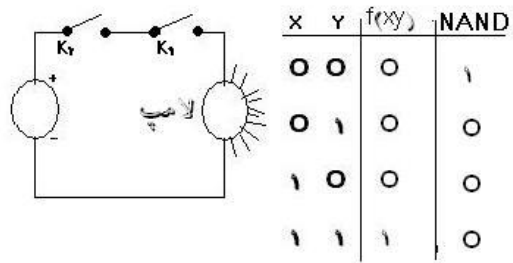
CMOS های IC

گیت های منطقی

گیت NOR, OR

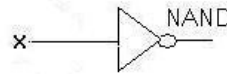
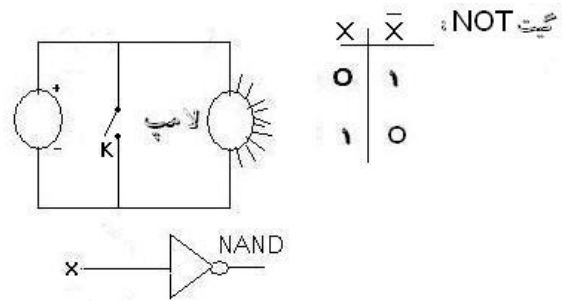
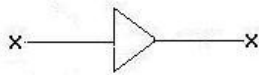


گیت NAND, AND



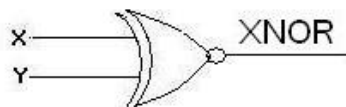
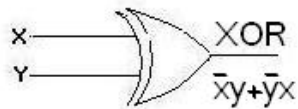
گیت بافر:

ولتاژ ثابت است ولی جریان را افزایش می دهد.
(این گیت منطق را تغییر نمی دهد)

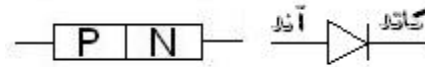


گیت XOR (گیت آشکار ساز نابرابری)

گیت XNOR (آشکار ساز برابری)



x	y	XOR	XNOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1



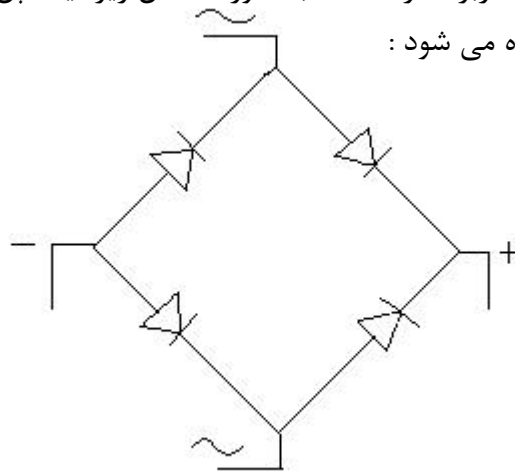
دیود:

دیود به قطعات دو پایایی گفته می شود که نسبت به عبور جریان در یک سمت مقاومت کمی نشان داده و در سمت مخالف مقاومت آنها بالا است. جهتی که جریان عبور داده می شود را معمولاً جهت مستقیم و مخالفش را معکوس می نامند.

زمانی که ولتاژ آند دیود از کاتدش بیشتر باشد اتصال کوتاه شده و جریان را عبور می دهد:



دیود ها بیشتر برای یکسو ساز کاربرد دارند مثلاً به صورت شکل زیر (یک پل دیودی) برای تبدیل ولتاژ متناوب به مستقیم استفاده می شود:

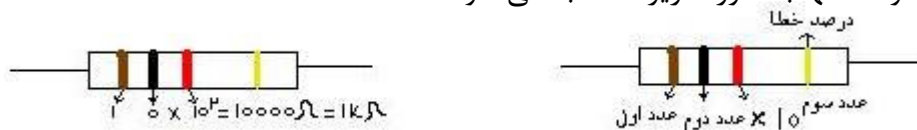


مقاومت:

مقاومت ها امکان کنترل جریان و یا ولتاژ ارائه شده را فراهم می کنند , در هر مدار باید موارد زیر را در مد نظر داشت:

مقاومت: مقدار مقاومت مورد نیاز که بر حسب اهم , کیلو اهم , و یا مگا اهم بیان می شود.
تولرانس (یا میزان دقت), ضریب حرارتی , ایجاد نویز, پایداری , ..
کدهای رنگی :

مقاومت های کربنی و اکسید فلزی همواره به صورت کد های رنگی بر روی بدنه آنها نشان داده می شوند. که مقاومت آنها به صورت زیر محاسبه می شود:



- | | | | | | | | | | | | |
|------|---------|------|--------|-----|-----|-----|------|---------|------|-------|---------|
| ۰ | ۱ | ۲ | ۳ | ۴ | ۵ | ۶ | ۷ | ۸ | ۹ | % | % |
| سیاه | قهوه ای | قرمز | نارنجی | زرد | سبز | آبی | بنفش | خاکستری | سفید | طلایی | نقره ای |

خازن:

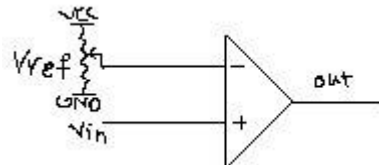
خازن در واقع انرژی الکتریکی را به صورت بار الکتریکی ذخیره می کند.
 ظرفیت خازن: مقدار را مشخص کرده و بر حسب میکروفاراد - نانو فاراد و یا پیکو فاراد بیان می شود.
 ولتاژ کار: معادل حد اکثر اختلاف پتانسیلی است که می توان به صورت مداوم به خازن اعمال کرد ولی اگر ولتاژ اعمال شده از این مقدار بیشتر شود به خازن آسیب می رساند.
 خازن سرامیکی: حجم کم و ظرفیت القایی اندک .
 خازن الکترولیت: ظرفیت های نسبتا بالایی دارند این نوع خازن ها قطبی بوده و باید در مدار هایی با ولتاژ DC و قطبیت مشخص مورد استفاده قرار گیرند.



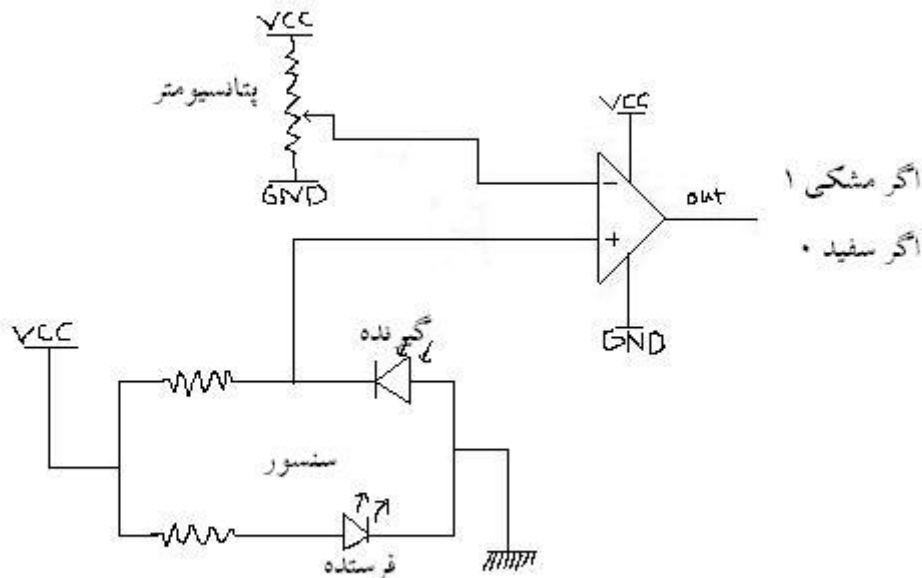
آپ امپ (تقویت کننده های عملیاتی):

حالت خاصی از آپ امپ:

اگر ولتاژ اعمالی به ورودی برابر ولتاژ مرجع باشد - خروجی صفر است.
 اگر ولتاژ اعمالی به ورودی بیش از مرجع باشد - خروجی یک ولتاژ مثبت است.
 اگر ولتاژ اعمالی به ورودی کمتر از مرجع باشد - خروجی یک ولتاژ منفی است.



کاربرد آپ امپ: ولتاژ ورودی را از سنسور دریافت کرده با توجه به ولتاژ Reference (ولتاژی که به وسیله پتانسیومتر تنظیم شده) خروجی را صفر و یک (منطقی) می کند .



سنسور های مادون قرمز :

این گونه از سنسور ها از یک فرستنده و یک گیرنده مادون قرمز تشکیل شده است. پرتوهای مادون قرمز توسط فرستنده فرستاده شده و به سطح برخورد می کند و بازتاب آن توسط گیرنده دریافت می شود .

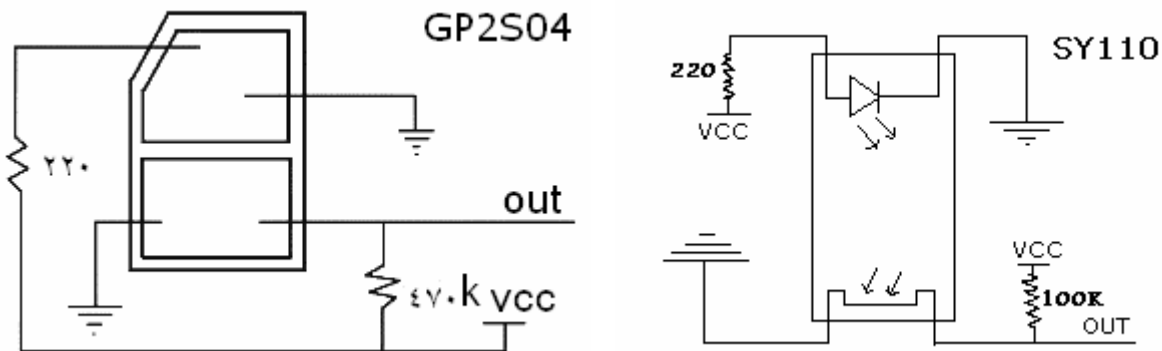
اینگونه از سنسورها به دلیل اینکه از محیط اثر نمی گیرند , درصد خطا بسیار پائینی دارند بنابراین ما در مورد این سنسورها صحبت می کنیم.

سنسور های مادون قرمز به دو دسته تقسیم می شوند دسته اول LED فرستنده و گیرنده جدا از هم است. این گونه از سنسورها در اندازه 5mm و 3mm موجود است. نوع 3 میلیمتری به دلیل اینکه LED فرستنده و گیرنده هم رنگ هستند تشخیص آنها سخت است و باید توسط مالتی متر انجام شود.

مشکل دیگر این سنسور ها این است که نوع تقلبی آن نیز بسیار زیاد است و امکان دارد از بین 5 سنسور 1 یا 2 تا خراب باشد .

گونه ی دیگر سنسور های مادون قرمز به صورتی است که فرستنده و گیرنده با هم در یک قطعه قرار دارند این گونه از سنسور ها دارای اندازه بسیار کوچک و دقت خوبی هستند . برای بایاس کردن سنسور :

مقاومت R2 برای تنظیم جریان Led فرستنده قرار دارد برای فرستنده 20 تا 40 میلی آمپر است. Led گیرنده به صورت معکوس بایاس می شود گیرنده به صورت یک مقاومت متغیر در برابر اشعه مادون قرمز عمل می کند با دریافت اشعه مادون قرمز مقاومت آن شدت افت پیدا می کند ولتاژ پایه مثبت مقایسه گر را کاهش می دهد تا اینکه خروجی آن یک (VCC) شود .



موتور های DC

جهت چرخش موتور به جهت جریان بستگی دارد با معکوس کردن جهت جریان می توانیم جهت حرکت موتور را عوض کنیم.

سرعت یک موتور DC بر حسب دور بر دقیقه (rpm) بیان می شود به جریان و بار موتور بستگی دارد بدون بار 500 rpm تا 1000 rpm

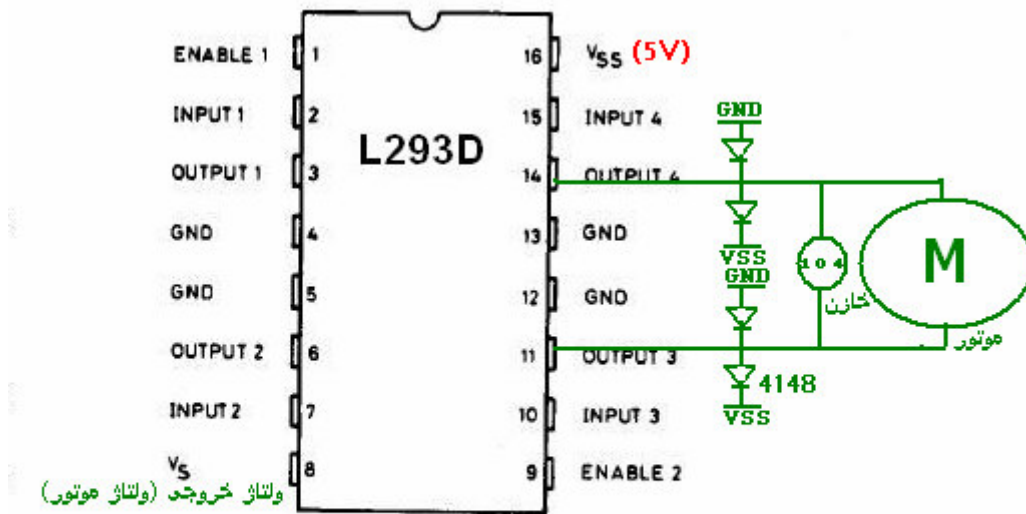
ولتاژ: موتور های DC کوچک را می توان با ولتاژ های نامی در محدوده 1.5 V تا 48 V تهیه نمود ولتاژ مشخص شده بر روی موتور ، ولتاژ نامی موتور را نشان می دهد که برای کارکرد معمولی موتور اعمال می شود در عمل ولتاژ نامی بسیار مهم می باشد ، چرا که این ولتاژ نشان دهنده حد اکثر ولتاژ توصیه شده ای که می توان به موتور اعمال نمود.

جریان: مقدار جریان عبوری از موتور ، زمانی که موتور تحت ولتاژ نامی کار می کند ، به مقدار بار بستگی دارد با افزایش مقدار بار مقدار جریان افزایش می یابد .

باید از کارکرد موتور با بار بیش از حد مجاز جلوگیری کنیم باعث سوختن موتور می شود. موتور های DC متداول دارای جریان های کارکرد در محدوده 50 ma تا بیش از 2 A می باشند.

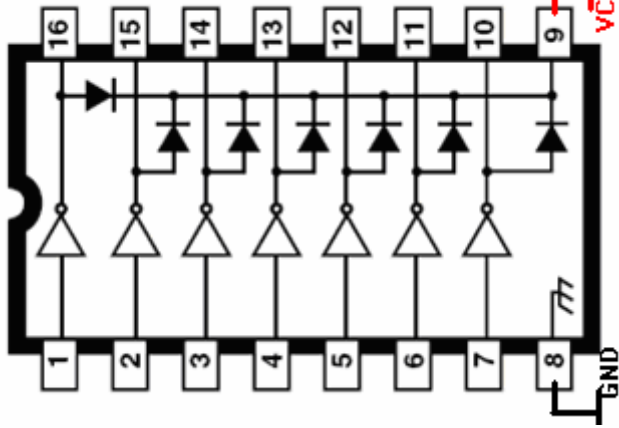
دراپور موتور ها :

(آی سی L293)



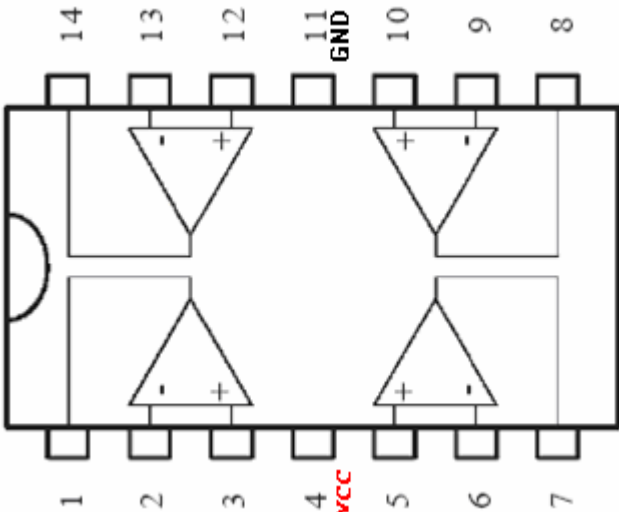
ولتاژ کاری از 5 ولت تا ۳۶ ولت است
 و جریان خروجی در بهترین حالت ۶۰۰ میلی آمپر و بیشترین جریان ۲ آمپر است. برای جریان های بیشتر ، از آی سی L۲۹۸ استفاده می کنیم.

ENABLE : فعال ساز
 INPUT : ورودی
 OUTPUT : خروجی



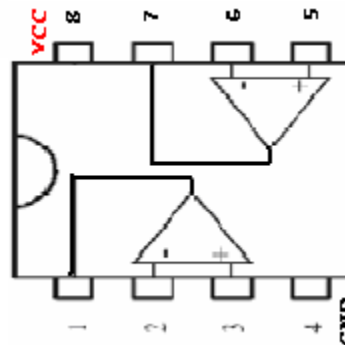
آی سی ULN2003
 آی سی بافر (NOT)

این آی سی یک درایور موتور با 7 کانال هستش
 که هر کانال 600 میلی آمپر جریان

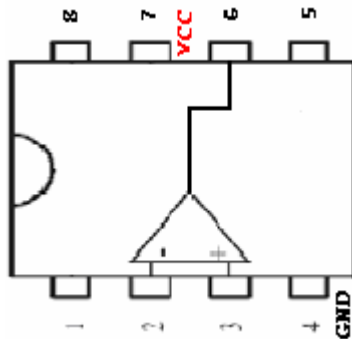


آی سی آپ امپ
 LM324

این آی سی 4 آپ امپ دارد
 در یک حالت خاص: ورودی آنالوگ را از
 سنسور ها را با یک ولتاژ مرجع مقایسه می کند
 و به صفر و یک برای میکرو تبدیل می کند.



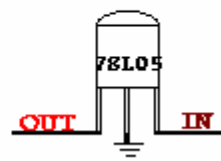
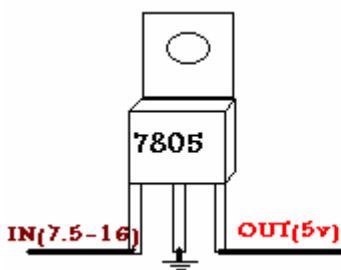
آی سی LM358



آی سی 741

آی سی ریگولاتور 7805 :

آی سی 7805 از 500 تا 1 آمپر جریان می دهد
 آی سی 78L05 در حدود 150 ma جریان می دهد .



میکروکنترلر:

میکروکنترلر های معروف (6811 از موتورولا - 8051 از اینتل - Z8 از زیلوگ - PIC 16x از میکروچیپ و AVR از اینتل)

که در بسته بندی از نوع DIP (چهل پایه) و QFP (بسته مسطح) تهیه شده اند.

هر یک از میکروکنترلر های فوق مجموعه دستورات و مجموعه ثبات های خود را دارد بنابر این با یکدیگر سازگار نیستند. برنامه ای که بر روی یکی از آنها نوشته شود بر روی دیگری قابل اجرا نیست. تفاوت میکروکنترلر و میکروپروسسور های همه منظوره: میکروپروسسور ها فاقد RAM و ROM و پورت های I/O در درون خود تراشه هستند. سرعت میکروپروسسور ها بالا تر از میکروکنترلر ها است میکروکنترلر ها دارای یک RAM و ROM و پورت های I/O از پیش تعیین شده هستند ولی برای میکروپروسسور ها خودمان طراحی می کنیم. انتظاراتی که از میکروکنترلر داریم:

1- برآورده کردن نیاز محاسبات کار به صورت موثر و مقرون به صرفه .

2- در دست داشتن نرم افزار های کمکی مانند کامپایلر ها اسمبلر ها و عیب یاب ها.

3- منابع گسترده و قابل اعتماد برای میکروکنترلر ها .

4- به وسیله Programmer و کامپیوتر می توان میکرو را پروگرام کرد.

میکروکنترلر های AVR به سه دسته تقسیم می شوند :

1-Tiny AVR (ATiny)

2-Classic AVR (AT90s)

3-mega AVR (ATmega)

تفاوت این 3 نوع به امکانات موجود در آنها مربوط می شود . Tiny AVR ها غالباً تراشه هایی با

تعداد پایه و مجموعه دستورات کمتری نسبت به Mega AVR ها می باشند و به عبارتی از

لحاظ پیچیدگی حد اقل امکانات را دارند Mega AVR ها حداکثر امکانات را دارند و Classic AVR

ها جایابی بین این دو نوع قرار می گیرند.البته از آنجایی که از بین سه دسته ذکر شده

Classic AVR ها، قبل از دو گروه دیگر تولید شده اند امروزه در طرح های جدید کمتر از آنها

استفاده می شود و عملاً هر یک از آنها با تراشه ای از گروه Mega AVR یا Tiny AVR

جایگزین شده اند.

در این جزوه ما از میکروکنترلر ATMEGA8 استفاده می کنیم.

خصوصیات Atmega8 , Atmega8L

از معماری AVR RISC استفاده می کند .

- کارایی بالا و توان مصرفی کم.

- دارای 130 دستور العمل با کارایی بالا که اکثرا تنها در یک کلاک سیکل اجرا می شوند
- 32*8 رجیستر کاربردی .
- سرعتی تا 16MIPS در فرکانس 16MHZ
- حافظه , برنامه و داده غیر فرار**
- 8K بایت حافظه FLASH داخلی قابل برنامه ریزی .
- پایداری حافظه FLASH : قابلیت 10000 بار نوشتن و پاک کردن (WRITE/ERASE)
- 1024 بایت حافظه داخلی SRAM
- 512 بایت حافظه EEPROM داخلی قابل برنامه ریزی .
- پایداری حافظه EEPROM : قابلیت 10000 بار نوشتن و پاک کردن (WRITE/ERASE)
- قفل برنامه FLASH و حفاظت داده EEPROM

خصوصیات جانبی

- دو تایمر – کانتر (TIMER/COUNTER) 8 بیتی با PRESCALER مجزا و دارای مد .COMPARE
- یک تایمر – کانتر (TIMER/COUNTER) 16 بیتی با PRESCALER مجزا و دارای مد های CAPTURE , COMPARE
- 3 کانال PWM
- 8 کانال مبدل آنالوگ به دیجیتال در بسته بندی های TQFP , MLF
- 6 کانال با دقت 10 بیتی
- 2 کانال با دقت 8 بیتی
- 6 کانال مبدل آنالوگ به دیجیتال در بسته بندی PDIP
- 4 کانال با دقت 10 بیتی
- 2 کانال با دقت 8 بیتی
- دارای RTC (REAL-TIME CLOCK) با اسیلاتور مجزا.
- یک مقایسه کننده آنالوگ داخلی .
- USART سریال قابل برنامه ریزی
- WATCHDOG قابل برنامه ریزی با اسیلاتور داخلی .
- ارتباط سریال SPI برای برنامه ریزی داخل مدار (IN – SYSTEM PROGRAMING)
- به صورت MASTER یا SLAVE
- قابلیت ارتباط با پروتکل دو سیمه (TWO-WIRE)
- خصوصیات ویژه میکروکنترلر**
- POWER-ON RESET CIRCUIT

- دارای 5 حالت (STANDBY , IDLE , ADC NOISE REDUCTION , SLEEP , POWER – SAVE, POWER – DOWN)
- منابع وقفه (INTERIGHTUPT) داخلی و خارجی
- دارای اسیلاتور RC داخلی کالیبره شده
- عملکرد کاملا ثابت.

- توان مصرفی پایین و سرعت بالا توسط تکنولوژی CMOS
توان مصرفی در 25°C, 3V,4MHZ

- حالت فعال 3.6 میلی آمپر (ACTIVE MODE)
- در حالت بی کاری 0.1 میلی آمپر (IDLE MODE)
- در حالت POWER – DOWN : 5 میکرو آمپر

ولتاژ های عملیاتی (کاری)

- 2.7v تا 5.5 برای (Atmega81)
- 4.7v تا 5.5v برای (Atmega8)

فرکانس های کاری

- 0MHZ تا 8MHZ برای (Atmega81)
- 0MHZ تا 16MHZ برای (Atmega8)

خطوط I/O و انواع بسته بندی

- 23 خط ورودی / خروجی (I/O) قابل برنامه ریزی
- 28 پایه PDIP و 32 پایه TQFP , MLF

(فیوز بیت)

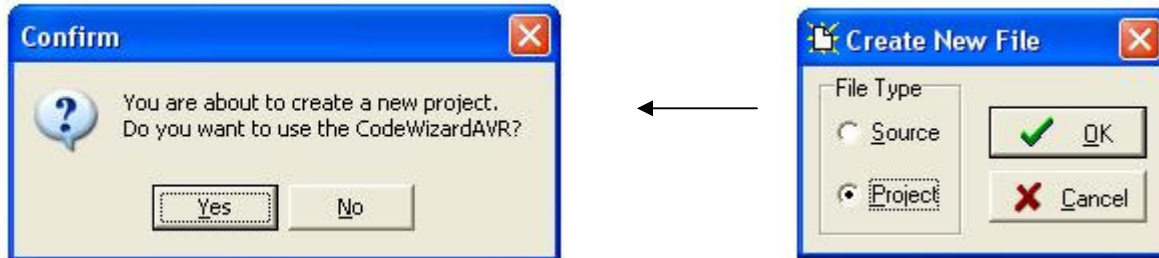
ترکیب پایه ها:

PDIP

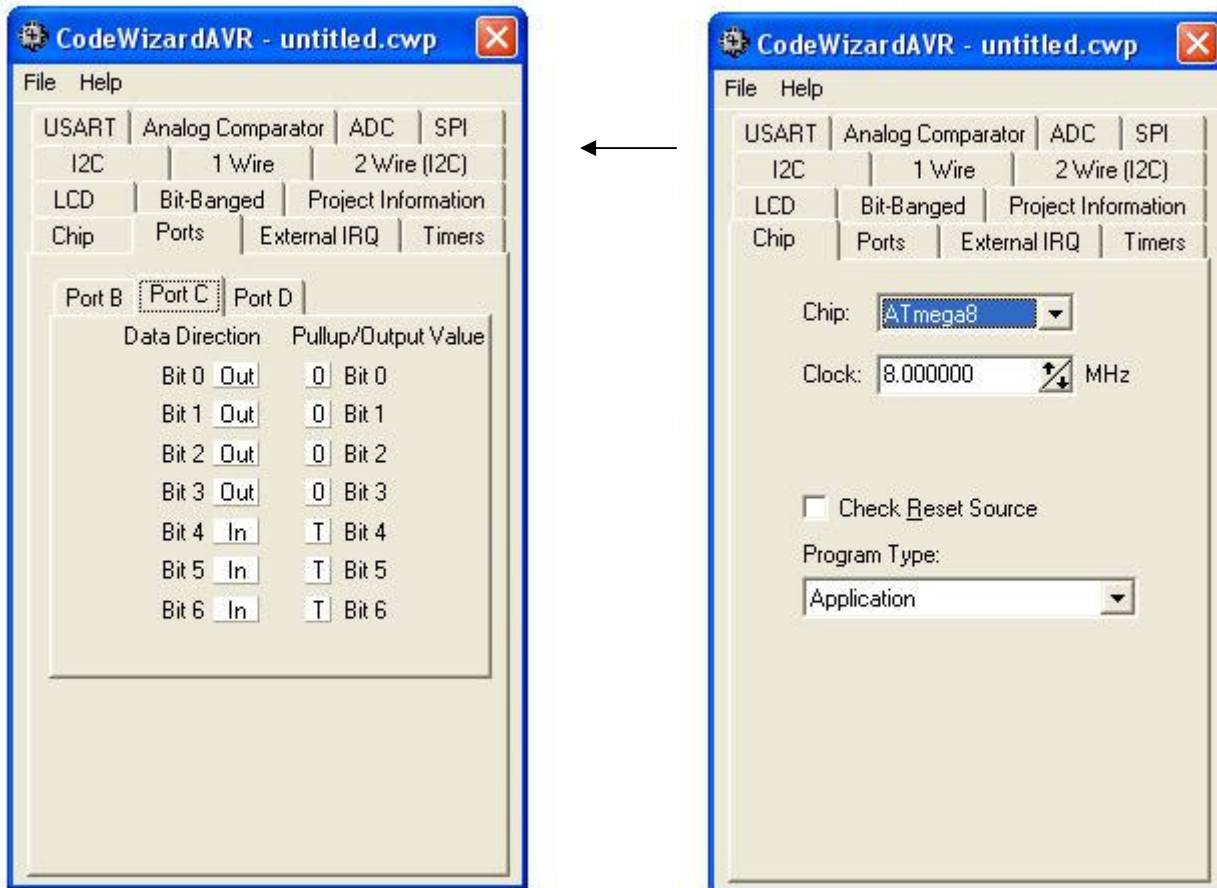
1	(RESET) PC6	28	PC5 (ADC5/SCL)
2	(RXD) PD0	27	PC4 (ADC4/SDA)
3	(TXD) PD1	26	PC3 (ADC3)
4	(INT0) PD2	25	PC2 (ADC2)
5	(INT1) PD3	24	PC1 (ADC1)
6	(XCK/T0) PD4	23	PC0 (ADC0)
7	VCC	22	GND
8	GND	21	AREF
9	(XTAL1/TOSC1) PB6	20	AVCC
10	(XTAL2/TOSC2) PB7	19	PB5
11	(T1) PD5	18	PB4
12	(AIN0) PD6	17	PB3
13	(AIN1) PD7	16	PB2
14	(ICP) PB0	15	PB1

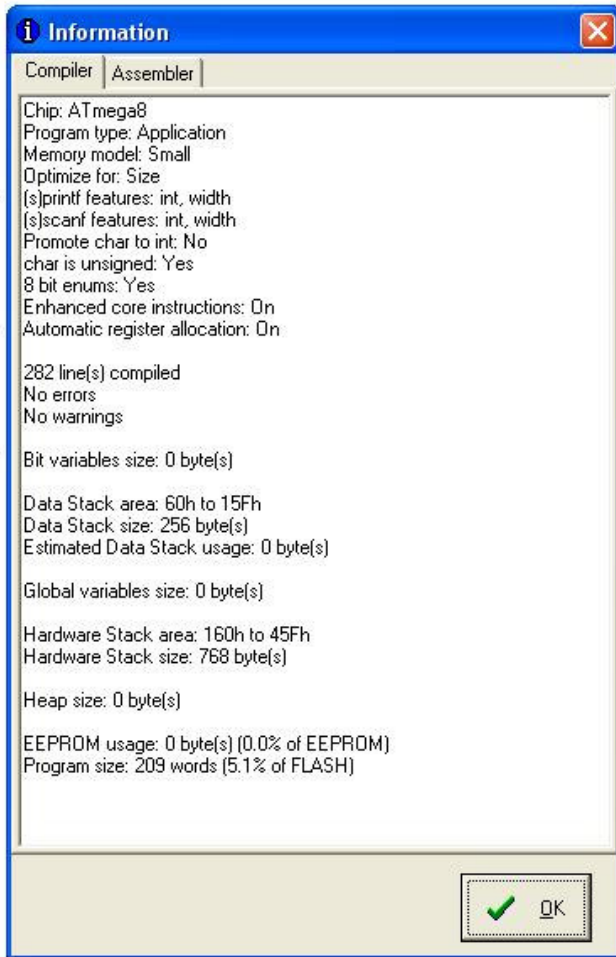
کار با Codewizard AVR و Codevision AVR

بعد از نصب و اجرای برنامه از منوی **File** گزینه **New** را کلیک می کنید و پنجره زیر باز می شود. به همین صورت قسمت **project** را انتخاب کنید.

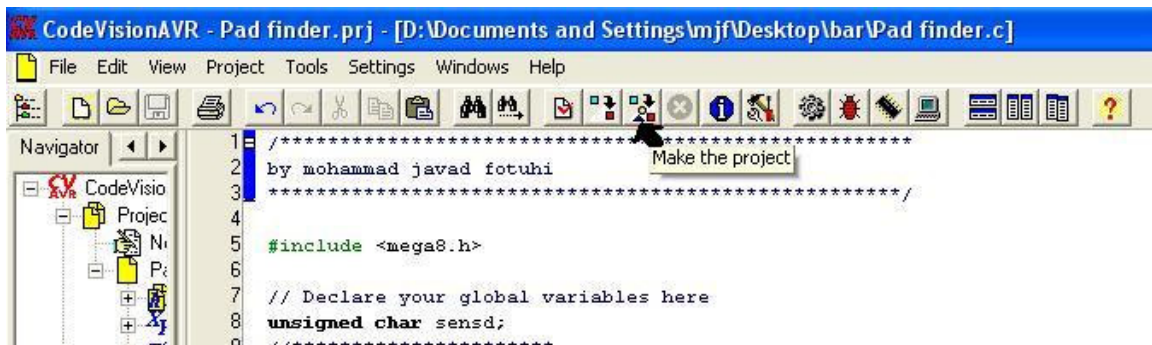


بعد از کلیک روی دکمه **ok** پنجره **confirm** باز می شود و دکمه **yes** را کلیک می کنیم. بعد به ترتیب قسمت های زیر را تنظیم می کنیم

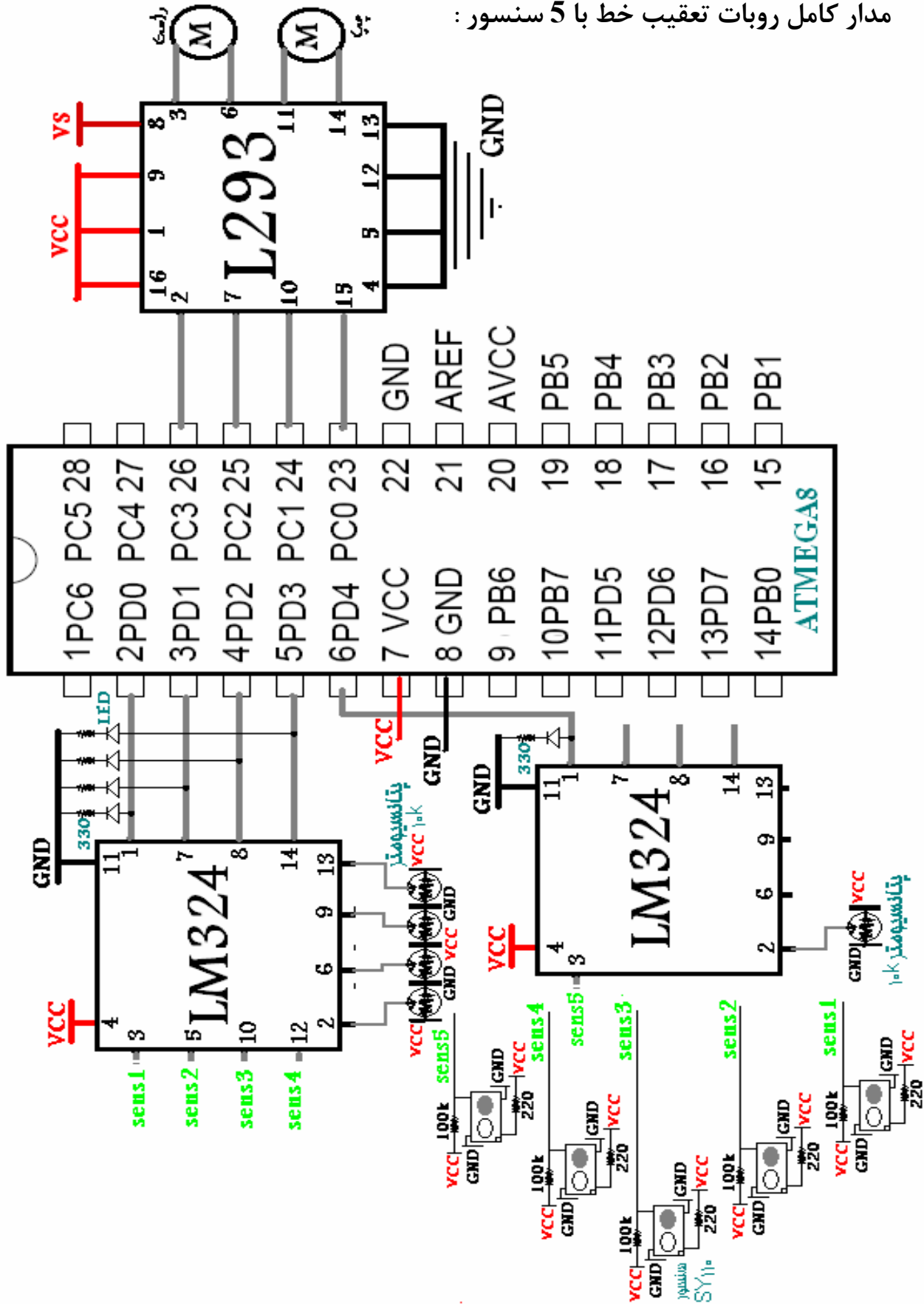




بعد از نوشتن برنامه بر روی دکمه **Make the project** کلیک می کنیم
اگر برنامه **erightor** نداشته باشد **hex** برنامه شما درست شده و آماده پروگرام است.



مدار کامل روبات تعقیب خط با 5 سنسور :



برنامه کامل روبات تعقيب 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;  
DDRD=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 Falling Edge
// Timer 1 Overflow Interrupt: Oforward
// Input Capture Interrupt: Oforward
// Compare A Match Interrupt: Oforward
// Compare B Match Interrupt: 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 Interrupt(s) initialization
// INT0: Oforward
// INT1: Oforward
MCUCR=0x00;

// Timer(s)/Counter(s) Interrupt(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=PIN_D&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();

    };
}
```

تهیه کننده : محمد جواد فتوحی

jfotuhi@gmail.com

www.4robot.ir