

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

میکرو 6 پرو سسور

تیمو تنظیم
سلان راجی

سرفصل ها:

□ آشنایی با میکرو کنترلر AtMega32

- بررسی پایه های میکرو کنترلر

- بررسی حافظه میکرو کنترلر

□ برنامه نویسی به زبان C

- تعریف متغیر

- ساختارهای شرطی

- ساختارهای تکرار

- آرایه ها

- تفاوت های برنامه نویسی در کامپیوتر و میکرو کنترلر

- آشنایی با محیط CodeVision

- نحوه برنامه ریزی میکرو کنترلر

□ پیاده سازی برنامه های ساده

- تعریف کتابخانه ها

- تعریف فرکانس کاری

- ورودی و خروجی کردن پایه ها

- راه اندازی LCD

- راه اندازی کی بورد

- راه اندازی تابلوروان

- راه اندازی موتورهای پله ای

□ تایمر صفر در میکرو کنترلر

- ایجاد تاخیرهای زمانی

- تولید شکل موج مربعی در خروجی میکرو کنترلر

- آشنایی با ربات تعقیب خط

- تبدیل تایمر به کانتر (فرکانس متر)

□ تایمر یک در میکرو کنترلر

- تولید دو موج مربعی همزمان در میکرو کنترلر

- طراحی کامل ربات تعقیب خط

□ وقفه های خارجی

- راه اندازی سنسور مادون قرمز با وقفه خارجی

- راه اندازی سنسور اولتراسونیک با وقفه خارجی

□ تبدیل آنالوگ به دیجیتال

- راه اندازی سنسور دما

- راه اندازی سنسورهای مقاومتی

- راه اندازی صفحات لمسی

□ ارسال اطلاعات بصورت سریال

- ارتباط دو میکرو کنترلر بصورت سریال

- راه اندازی GPS

- راه اندازی ماژولهای فرستنده و گیرنده

- برقراری ارتباط میکرو کنترلر با کامپیوتر (VB)

□ پروژه های کاربردی

- راه اندازی GLCD

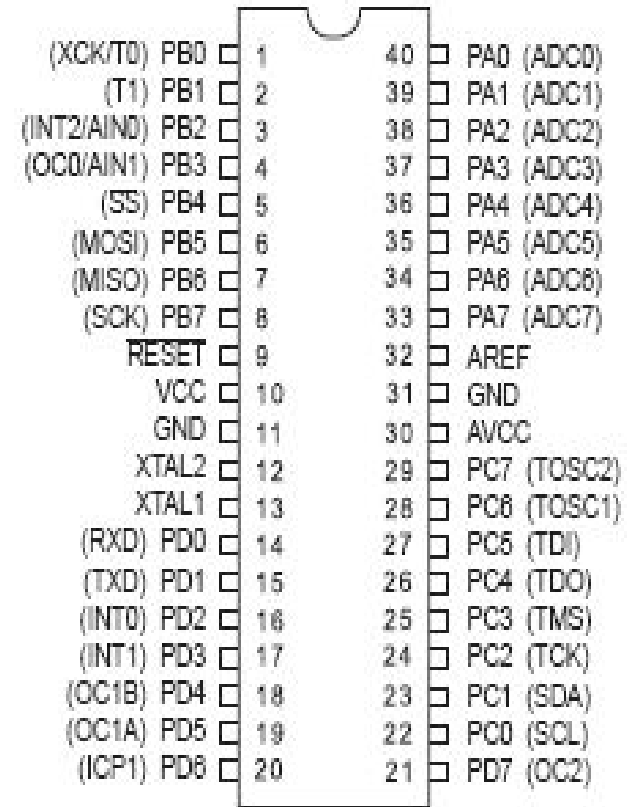
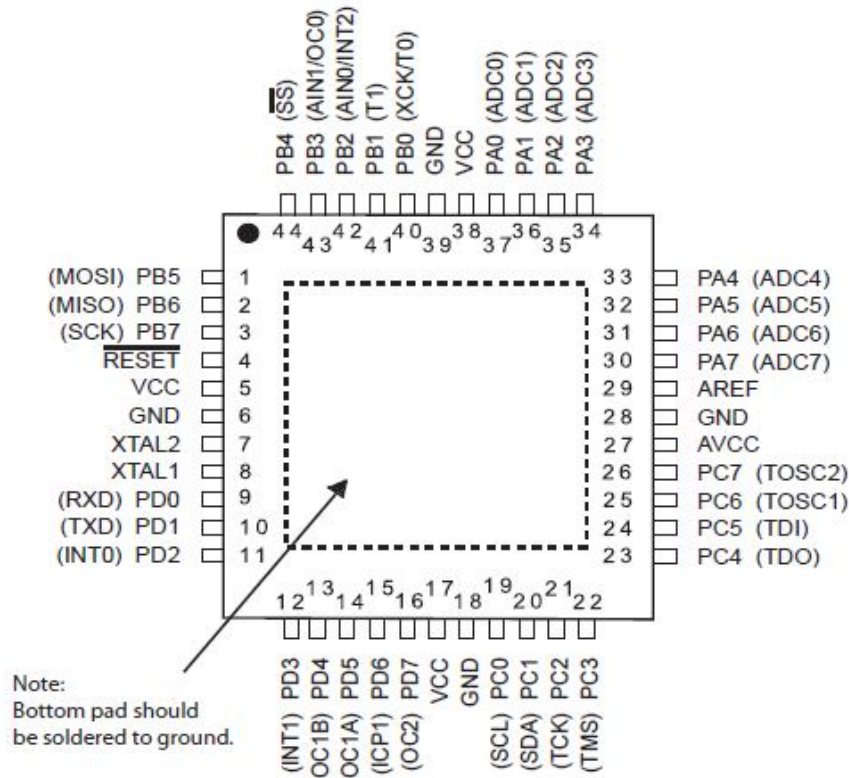
- راه اندازی LCD های رنگی

- راه اندازی Sim900

- راه اندازی میکرو ماوس

- راه اندازی سروو موتورها

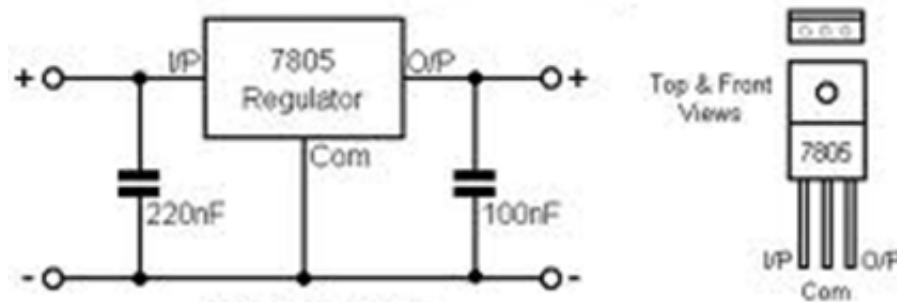
آشنایی با میکروکنترلر ATMEGA32



میکرو پرو سسور

تیم و تنظیم
سلطان راجی

پایه Vcc و Gnd



(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 (SOL)
(ICP1) PD6	20	21	PD7 (OC2)

1. رگولاتورهای ولتاژ خروجی ثابت مثبت: که خروجی آنها یک عدد ثابت و غیر قابل تغییر + می باشد که نام گذاری آنها نیز به صورت 78XX یا L78XX یا M78XX می باشد. 2. رقم سمت راست که به صورت XX نشان داده شده نشان دهنده ولتاژ خروجی است. مثلاً ولتاژ خروجی رگولاتور 7805، 5 ولت می باشد و همچنین L یا M هم نشان دهنده حداکثر جریان دهی آن است (L=1 آمپر، M=1.5 آمپر)

2. رگولاتورهای ولتاژ خروجی ثابت منفی: که خروجی آنها یک عدد ثابت منفی و غیر قابل تغییر - می باشد که نامگذاری آنها به صورت 79XX می باشد.

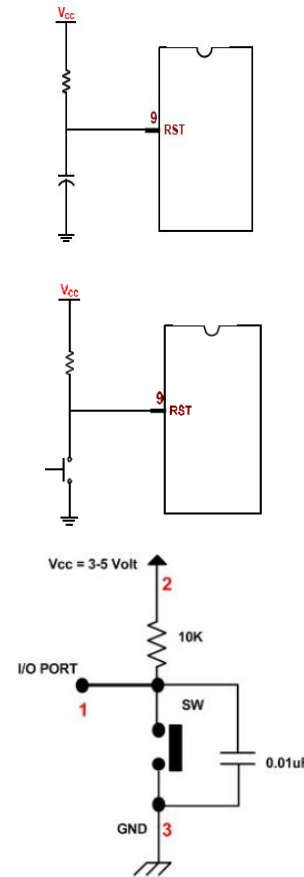
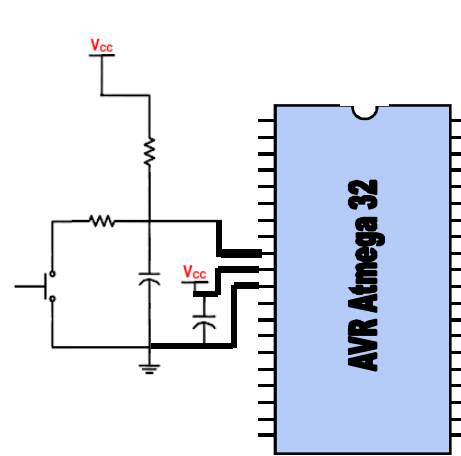
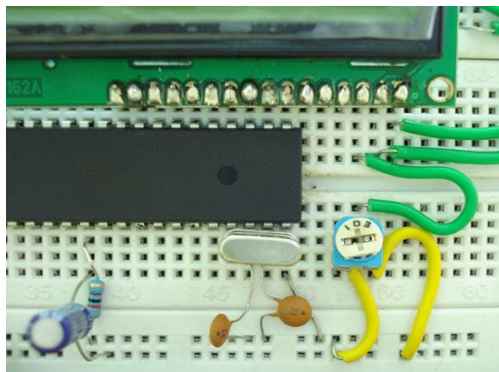
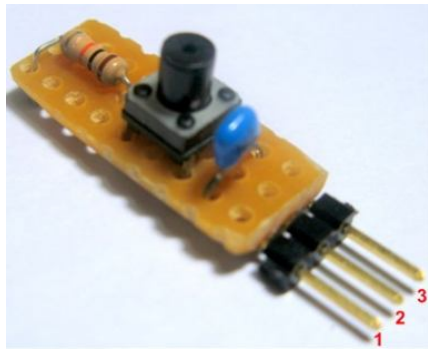
3. رگولاتورهای ولتاژ خروجی متغیر: به وسیله این رگولاتورها می توان ولتاژ خروجی را کنترل کرد. معروف ترین و پرکاربردترین نوع خروجی + آنها LM317 و LM338 و LM337 - خروجی - آنها می باشد. این قطعه برای رده اندازگی نیاز به یک مدار جانبی مختصر دارد.



شماره مدل	ولتاژ ورودی	ولتاژ خروجی
7805	7.3	5
7809	11.5	9
7812	14.6	12
7818	21	18
7824	27.1	21

میکرو پرو سسور

توسعه سیستم
سلمان راجی



(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)

انواع ریست : ۱- بهنگام روشن شدن میکروکنترلر ۲- ریست خارجی ۳- ریست توسط تایمر نگهبان (Watchdog)

تایمر نگهبان : (ریست نمودن اتوماتیک بهنگام هنگ کردن میکرو)

روش فعال نمودن Watchdog:

۱- فعالسازی فیوزبیت WDTON

۲- فعالسازی نرم افزاری با رجیستر WDTCR

-	-	-	WDCE	WDE	WDP2	WDP1	WDP0
---	---	---	------	-----	------	------	------

Whatchdog: on WDTCR:00011110

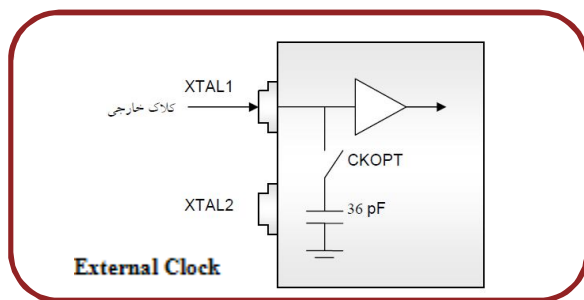
Whatchdog: off WDTCR:00011110 WDTCR:00000000

Low	High	Ext'd
☒ CKDIV8	☐ DWEN	
☐ CKOUT	☐ EESAVE	
☐ SUT1	☒ SPIEN	
☒ SUT0	☐ WDTON	
☒ CKSEL3	☐ BODLEVEL2	
☐ CKSEL2	☐ BODLEVEL1	
☒ CKSEL1	☐ BODLEVEL0	
☒ CKSEL0	☐ RSTDISBL	☐ SELFPRGEN

Low Byte	عملکرد	بیش فرض
CKSEL0	انتخاب منبع کلاک	۱
CKSEL1		۰
CKSEL2		۰
CKSEL3		۰
CKSEL3		۰

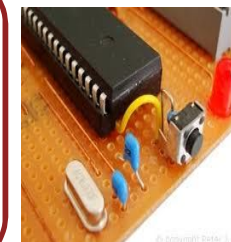
Device Clocking Option	CKSEL3..0
External Crystal/Ceramic Resonator	1111 - 1010
External Low-frequency Crystal	1001
External RC Oscillator	1000 - 0101
Calibrated Internal RC Oscillator	0100 - 0001
External Clock	0000

پایه‌های کریستال



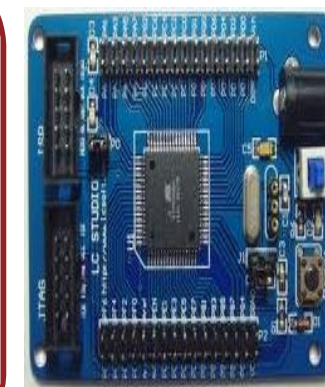
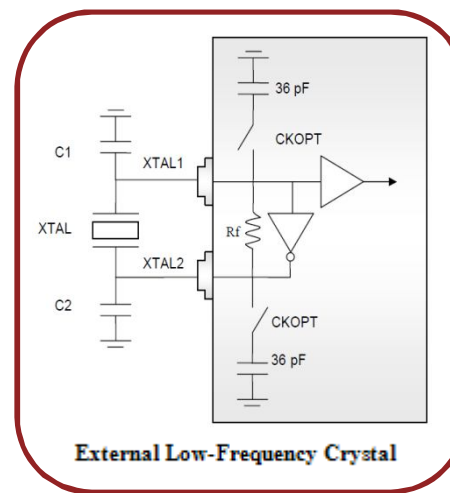
CKSEL3..0	Nominal Frequency (MHz)
0001 ⁽¹⁾	1.0
0010	2.0
0011	4.0
0100	8.0

Internal RC Oscillator

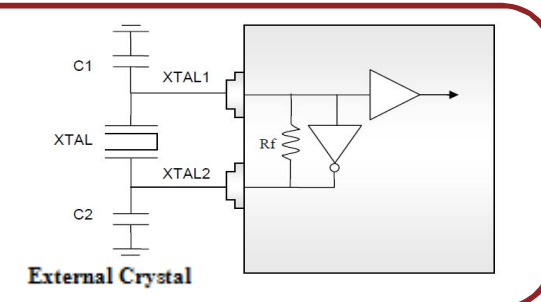


CKSEL3..0	Frequency Range (MHz)
0101	≤ 0.9
0110	0.9 - 3.0
0111	3.0 - 8.0
1000	8.0 - 12.0

External RC Oscillator



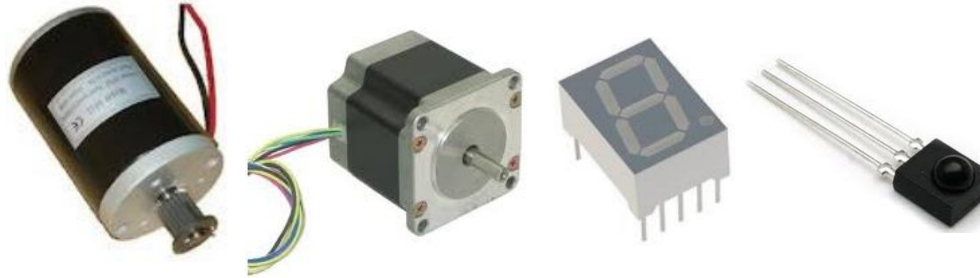
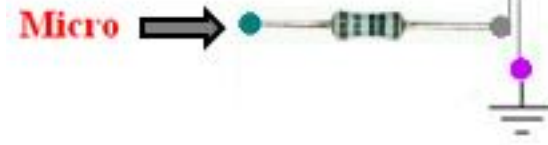
CKOPT	CKSEL3..1	Frequency Range (MHz)	Recommended Range for Capacitors C1 and C2 for Use with Crystals (pF)
1	101 ⁽¹⁾	0.4 - 0.9	—
1	110	0.9 - 3.0	12 - 22
1	111	3.0 - 8.0	12 - 22
0	101, 110, 111	1.0 ≤	12 - 22



میکرو پرو세서

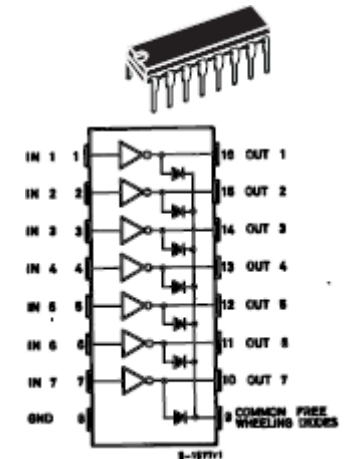
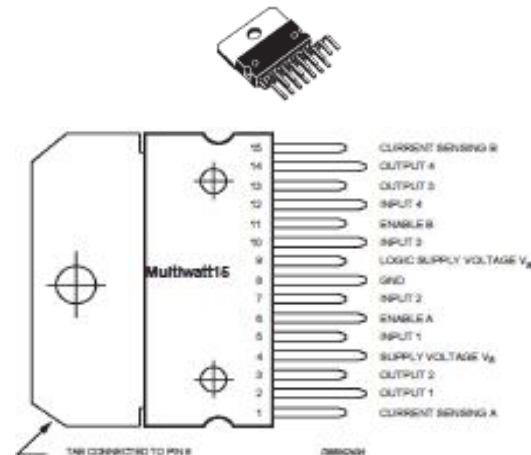
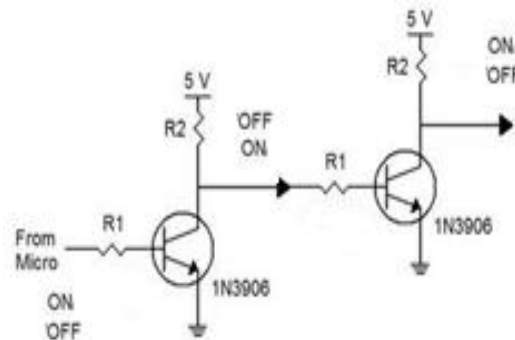
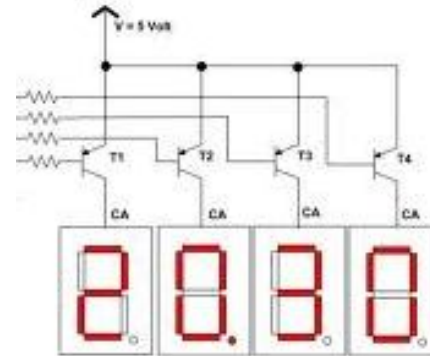
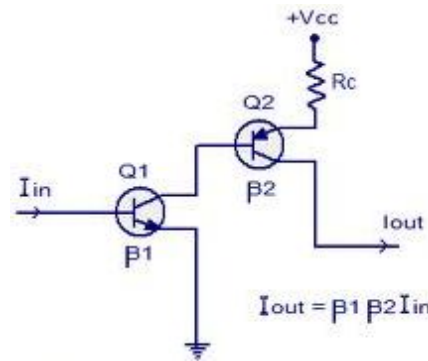
توسعه سیستم
سازمان راهبردی

پایه های ورودی خروجی (I/O)



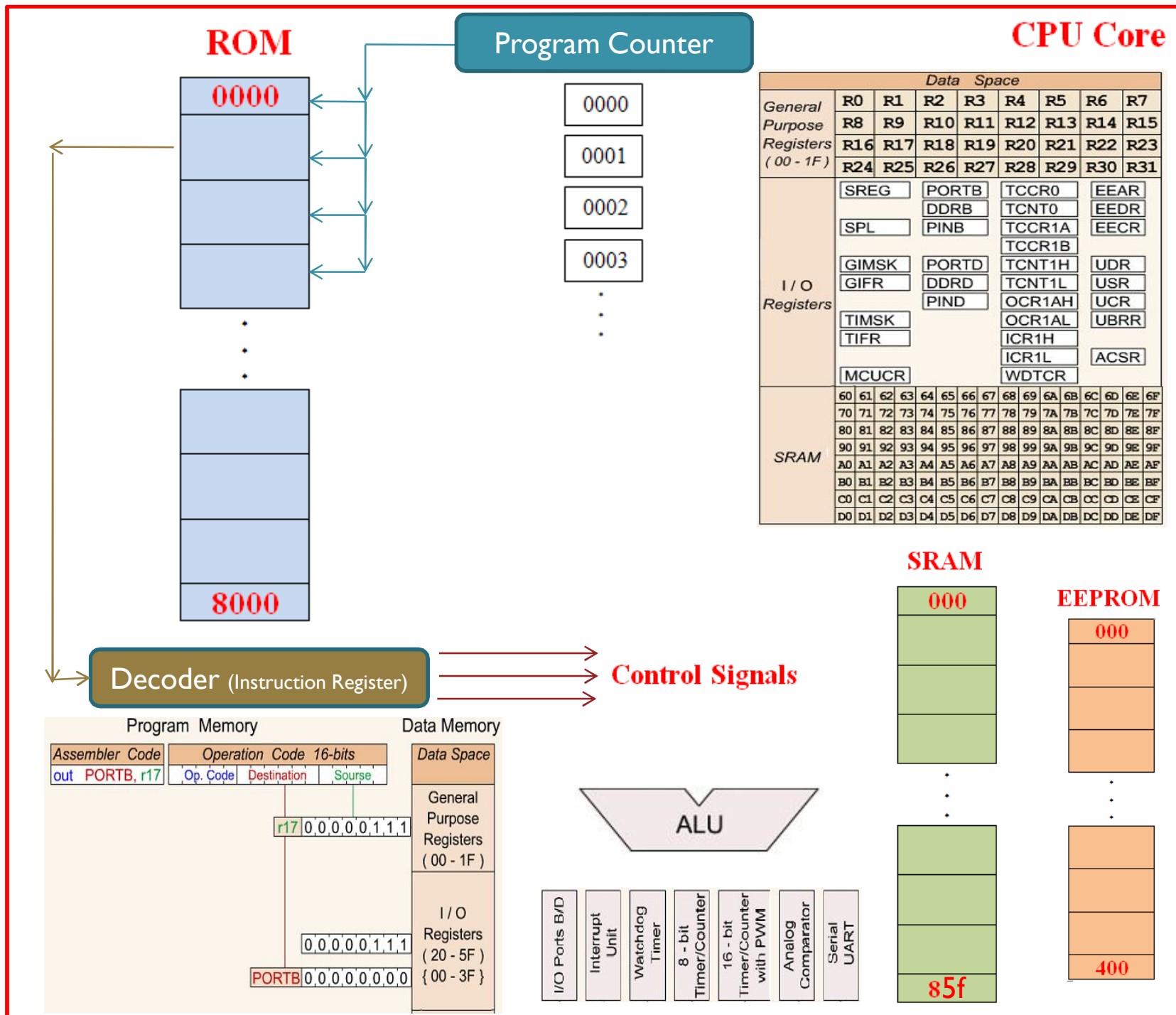
(XCK/TO) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(SS) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	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)

ماکزیم جریان خروجی: ۳۰ الی ۴۰ میلی آمپر



میکرو پرو سسور

توسعه سیستم
سلام راجی



الگوی برنامه در C

تعاریف کلی

الگوی توابع

```
void main()  
{  
    while(1) {  
        .  
        .  
        .  
    }  
}
```

توابع تعریف شده

- خط اول از رهنمود های پیش پردازنده است.

- while(1) ایجاد یک حلقه ی نامتناهی می کند.

- در انتهای هر عبارت یک سمی کالن می آید.

- Brace ابتدا و انتهای یک تابع و همچنین یک بلوک را مشخص می کند.

- از " " برای مشخص کردن ابتدا و انتهای یک رشته ی متنی استفاده می شود.

- از // یا /* ... */ برای نوشتن توضیحات استفاده می شود.

- شناسه ها اسامی متغیرها، ثوابت و یا توابع هستند.

- شناسه ها نمی تواند از کلمات رزرو شده باشند و همچنین نمی توانند با یک کاراکتر عددی شروع شود و طول

آن ها باید کمتر از ۳۱ کاراکتر باشد.

- C یک زبان Case Sensitive است و بین حروف کوچک و بزرگ تفاوت قائل می شود.

- کلمات رزرو شده حتما باید با حرف کوچک استفاده شوند. (مثل if, char, while,...)

```
#include<stdio.h>
```

→ تعریف کتابخانه

```
void main()
```

→ بدنه اصلی برنامه

```
{
```

```
    printf("Hello World!");
```

→ چاپ متن بر روی نمایشگر

```
    while(1);
```

→ انتظار بر روی آخرین خط برنامه

```
}
```

میکرو پروسیسور

توسعه سیستم
سلطان راجی

عملگرها در زبان C

عملگر	عملکرد	مثال	نتیجه
=	انتساب	a=b	a ⇐ b
=	ضرب و انتساب	a=b	a=a*b
/=	تقسیم و انتساب	a/=b	a=a/b
+=	جمع و انتساب	a+=b	a=a+b
-=	تفریق و انتساب	a-=b	a=a-b
a=(value)?x:y	انتساب شرطی		اگر value مقدار صحیح باشد a برابر x و در غیر این صورت برابر y می شود.

عملگر	عملکرد	مثال	نتیجه
>	بزرگتر	2>3	False
<	کوچکتر	'm'>'e'	True
>=	بزرگتر یا مساوی	5>=5	True
<=	کوچکتر یا مساوی	2.5<=4	True
==	تساوی	'A'=='B'	False
!=	نامساوی	2!=3	True
عملگر	عملکرد	مثال	نتیجه
-	قرینه	-a	a = a × -1
++	افزایش یک واحد	a++	a = a + 1
--	کاهش یک واحد	a--	a = a - 1

عملگر	عملکرد	مثال	نتیجه
*	ضرب	۳*۲	۶
/	تقسیم	۵/۲	۲.۵
+	جمع	۳+۶	۹
-	تفریق	۸-۳	۵
%	باقیمانده	۱۰٪۳	۱
&	AND	0xF0 & 0x0F	0x00
	OR	0x00 0x03	0x03
^	XOR	0x0F ^ 0xFF	0xF0
~	مکمل یک	~(0xF0)	0x0F
>>	شیفت به راست	0xF0 >> 4	0x0F
<<	شیفت به چپ	0x0F << 4	0xF0

A	1	0	1	0	1	1	1	0
B	0	1	0	1	1	0	1	0
A B	1	1	1	1	1	1	1	0

A	1	0	1	0	1	1	1	0
B	0	1	1	0	1	1	0	1
A & B	0	0	1	0	1	1	0	0

Number Formats:

- **0b** for binary numbers (base = 2)
- **0x** for hexadecimal numbers (base = 16)
- **No prefix** is needed for decimal numbers (base = 10)

-0b00110101 is an 8-bit binary number
 -0x7F is a hexadecimal number
 -126 is a decimal number

عملگر	عملکرد	مثال	نتیجه
&&	AND منطقی	(2>3)&&(1!=3)	False
	OR منطقی	('a'<3) !(10)	True
!	نقیض	!(7>5)	False

میکرو پروسیسور

توسعه سیستم
 سلمان راجی

تعریف متغیر

نام متغیر نوع متغیر

مثال:

char a ;

Type	Size (Bits)	Range
bit	1	0 , 1
char	8	-128 to 127
unsigned char	8	0 to 255
signed char	8	-128 to 127
int	16	-32768 to 32767
short int	16	-32768 to 32767
unsigned int	16	0 to 65535
signed int	16	-32768 to 32767
long int	32	-2147483648 to 2147483647
unsigned long int	32	0 to 4294967295
signed long int	32	-2147483648 to 2147483647
float	32	±1.175e-38 to ±3.402e38
double	32	±1.175e-38 to ±3.402e38

● متغیرهای local متغیرهایی هستند که در داخل توابع تعریف شده اند.

این متغیرها دارای ویژگی های زیر هستند.

- تنها در داخل تابعی که تعریف شده اند معتبرند.
- هنگام اجرای تابع ایجاد می شوند و هنگام خاتمه اجرا از بین می روند.

● به متغیرهایی که خارج از توابع تعریف می شوند متغیرهای سراسری یا global می گویند.

دو ویژگی متغیرهای global:

- برخلاف متغیرهای local، از نقطه ای که تعریف می شوند تا انتهای برنامه معتبرند.
- اگر هنگام تعریف مقداری به آنها داده نشود مقدارشان صفر در نظر گرفته می شوند.

```
#include <stdio.h>
int num=4;
int func1 (int x)
{
    num+=x;
    return num;
}
int func2 (int x)
{
    num-=x;
    return num;
}
void main ( )
{
    printf("%d",
    func1(3));
    printf("\n%d",
    func2(3));
}
```

● برای تعیین محل تعریف متغیر از عملگر @ استفاده می کنیم. مثال: int b@0xA3;

● برای تعریف متغیر در حافظه ی EEPROM از کلمه ی کلیدی eeprom قبل از نام متغیر استفاده می کنیم

مثال: eeprom int code;

● می توان در زمان تعریف به متغیر مقدار اولیه داد. مثال: char s = 'a';

● تعریف ثابت با کلمه ی کلیدی const یا flash انجام می شود. مثال: const float pi = 3.14 ;

● با رهنمود #define می توان ماکرو تعریف نمود، در این حالت مقداری که برای ثابت تعریف می شود نوع

داده را تعیین کرده و مقادیر تعریف شده، توسط پیش پردازنده با مقدار ثابت جایگزین می شود.

#define code 100 ;

مثال:

میکرو پرو سسور

تیم و تنظیم
سلطان راجی

آرایه‌ها و رشته‌ها

آرایه‌ها

آرایه‌ها مجموعه‌ای از متغیرهای هم‌نوع هستند.

- فرم تعریف: `[تعداد عناصر] اسم متغیر نوع متغیرهای آرایه`

- مثال: `int a[5];`

با مقدار اولیه: `int a[5] = {1,2,3,4,5};`

- فرم تعریف آرایه‌های دو بعدی:

`[تعداد عناصر ستون][تعداد عناصر سطر] اسم متغیر نوع متغیرها`

مثال: `int a[2][3]={{1,2,3},{4,5,6}};`

- در زبان C اندیس آرایه‌ها از صفر شروع می‌شود.

رشته‌ها

رشته‌ها آرایه‌ای کاراکترها هستند.

- مثال: `char name[ ] = "Test";`

یا: `char name[5]={'T','e','s','t','\0'};`

- رشته‌ها همواره به یک کاراکتر Null ختم می‌شوند.

- ✓ یادآوری: با دستور تعریف متغیر مثل `int x;` مکانی در حافظه به نام `x` به برنامه تخصیص داده می‌شود.
- ✓ در زبان C اگر بنویسیم `int x[5];`، پنج مکان مجاور هم در حافظه به برنامه تخصیص داده می‌شود این مکان‌ها به صورت `x[0]`، `x[1]`، `x[2]` و `x[4]` نامیده می‌شوند.
- ✓ اصطلاحاً می‌گوییم آرایه‌ای به نام `x`، از نوع `int` و با ۵ عنصر تعریف کرده ایم.
- ✓ به 0، 1، 2، 3 و 4 اصطلاحاً اندیس آرایه گفته می‌شود.
- ✓ به همین ترتیب می‌توان آرایه‌هایی از انواع دیگر مثل `float`، `char`، `short` و ... داشت.

```
#include <stdio.h>
void main()
{
    char str[20];
    int i;
    printf("Enter a string:");
    scanf("%s",str);
    i=0;
    while (str[i] != '\0')
    {
        if (str[i]>=97 && str[i]<=122)
            printf("%c",str[i++]-32);
    }
}
```

- ✓ نحوه مقداردهی: به دو صورت می‌توان به یک رشته مقدار داد:
- ✓ با استفاده از دستورات ورودی: مثل `scanf` با استفاده از کاراکتر کنترلی `%s`
- ✓ `scanf("%s",&st);` یا `scanf("%s",st)`
- ✓ به صورت مستقیم هنگام تعریف رشته
- ✓ `char st[20]="Ali";`
- ✓ نحوه قرار گرفتن رشته در حافظه به این صورت است:
- ✓ C در انتهای هر رشته به طور خودکار کاراکتر `NULL` (پوچ-تهی) (که با علامت `\0` نشان داده می‌شود) را قرار می‌دهد.
- ✓ در نتیجه طول واقعی یک رشته یکی بیشتر از تعداد کاراکترهایش است.

ساختار Switch Case

switch (مقدار)

```
{
    case مقدار ۱:
        دستورات ۱
        break;
    case مقدار ۲:
        دستورات ۲
        break;
    .
    .
    default:
        دستورات n
}
```

```
char x='b';
switch(x)
{
    case 'a':
        printf("%d",'a');
        break;
    case 'b':
        printf("%d",'b');
        break;
}
```

- ✓ متغیر یا مقدار صحیح با مقادیر موجود در case ها مقایسه می شود.
- ✓ اگر با مقدار یکی از case ها مطابقت داشت دستورات داخل آن اجرا می شود. در صورت عدم تطابق با هیچ یک از case ها دستورات بخش default اجرا می شود.
- ✓ آوردن بخش default اختیاری است .
- ✓ نیازی به قرار دادن {} ما بین دستورات case ها نیست.
- ✓ نوشتن break در انتهای دستورات هر case الزامی است.

دستور goto

پرش بدون شرط به یک برچسب انجام می شود.

ساختار if-else

<pre>if (x > 100) { y = 20; z ++; }</pre>	<pre>if (x > 100) { y = 20; z ++; } else { y = x - 1 }</pre>	<pre>if (شرط) { دستورات ۱ } else { دستورات ۲ }</pre>
--	---	--

در زبان C به عبارت مقایسه ای درست ارزش 1 و به عبارت مقایسه ای نادرست ارزش 0 نسبت داده می شود.

- ارزش $3 \geq 4$ ، 0 است.

- ارزش $(3 > 2) \parallel (4 \geq 5)$ ، 1 است.

- ✓ در صورتی که دستورات یک خطی باشند می توان brace را حذف نمود.
- ✓ استفاده از بلاک else اختیاری است.

ساختارهای تکرار

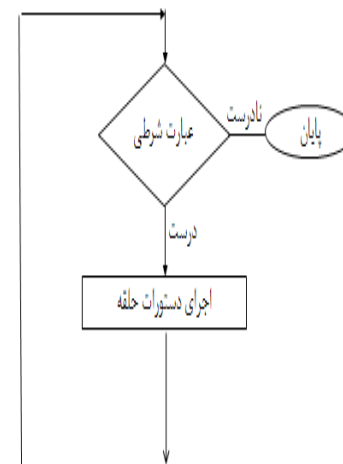
```
#include <stdio.h>
void main()
{
    int i;
    i=0;
    i++;
    printf("%d",i);
    i++;
    printf("%d",i);
    ....
    ....
    ...
}
```

با استفاده از دستورات تکرار (حلقه ها) می توانیم اجرای بخشی از برنامه را چند بار تکرار کنیم.

ساختار While

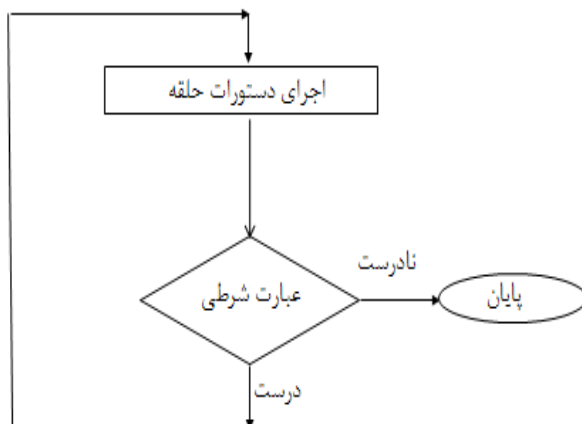
while (شرط)

```
{
    دستورات
}
```



ساختار Do/While

```
do{
    دستورات
} while (شرط)
```



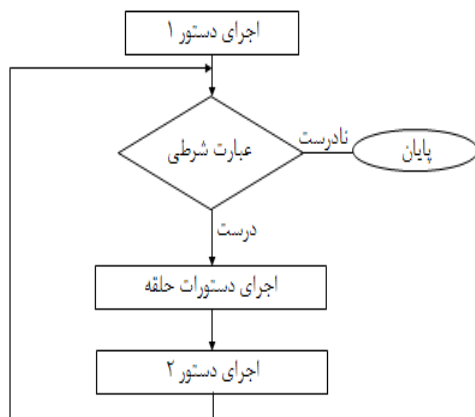
✓ در ساختار Do/While بر خلاف While شرط در انتهای حلقه آزمایش می شود، بنابراین دستورات داخل حلقه، حداقل یکبار اجرا می شوند.

ساختار for

(دستور ۲ عبارت شرطی دستور ۱)

(گام ۱ شرط پایان ۲ مقدار ابتدای حلقه)

```
{
    دستورات
}
```



✓ دستور break باعث خروج بدون شرط از هر حلقه ای می شود.

✓ دستور continue باعث می شود اجرای ادامه دستورات متوقف شده و حلقه از ابتدا آغاز شود.


```
#include <stdio.h>
void main()
{
    int i;
    float sum;
    float x;
    sum=0;
    for (i=0;i<100; i++)
    {
        scanf("%f", &x);
        sum+=x;
    }
    printf("The result=%f",sum);
}
```

```
#include <stdio.h>
void main()
{
    int i,j;
    for (i=1;i<=10;++i)
    {
        printf("\n");
        for (j=1;j<=10;++j)
            printf("%d\t",(i*j));
    }
}
```

```
#include <stdio.h>
void main()
{
    char ch;
    int num,sum;
    sum=0;
    do
    {
        printf("\nEnter a number: ");
        scanf("%d",&num);
        sum+=num;
        printf("\ncontinue(enter y/n):");
        scanf("\n%c",&ch);
    }
    while (ch=='y');
    printf("The result= %d",sum);
}
```

```
#include <stdio.h>
void main()
{
    char ch;
    int num,sum;
    ch='y';
    sum=0;
    while (ch=='y')
    {
        printf("\nEnter a number: ");
        scanf("%d",&num);
        sum+=num;
        printf("\ncontinue(enter y/n):");
        scanf("\n%c",&ch);
    }
    printf("The result= %d",sum);
}
```

تابع (زیر برنامه)

$$y=f(x)=x^2$$

- نام: (f)
- ورودی (x) که متعلق به مجموعه ای به نام دامنه است.
- مقدار برگشتی (y) که متعلق به مجموعه ای به نام برد است.
- ضابطه تابع (x^2) که ارتباط بین مقدار برگشتی و ورودی را مشخص می کند.
- تابع مقدار x را می گیرد و بر اساس ضابطه موجود، y را محاسبه کرده و برمی گرداند.

```
float func (float num)
{
    float result;
    if (num < 0)
        result=-num;
    else
        result=num;
    return result;
}
```

```
float f (float x)
{
    float y;
    y=x*x;
    return y;
}
```

(نام و نوع ورودی) نام تابع نوع برگشتی

```
{
    پیاده سازی ضابطه تابع با یک سری دستورات
    برگرداندن مقدار برگشتی
}
```

```
#include <stdio.h>
float f (float x)
{
    float y;
    y=x*x;
    return y;
}
void main()
{
    float z;
    z=f(3.5);
    printf("%f",z);
}
```

احضار تابع →

محل قرارگیری تابع

```
#include .....
#include ....
..... →
void main()
{
    .....
    .....
}
```

```
float f (float x)
{
    float y;
    y=x*x;
    return y;
}
```



CodeVisionAVR

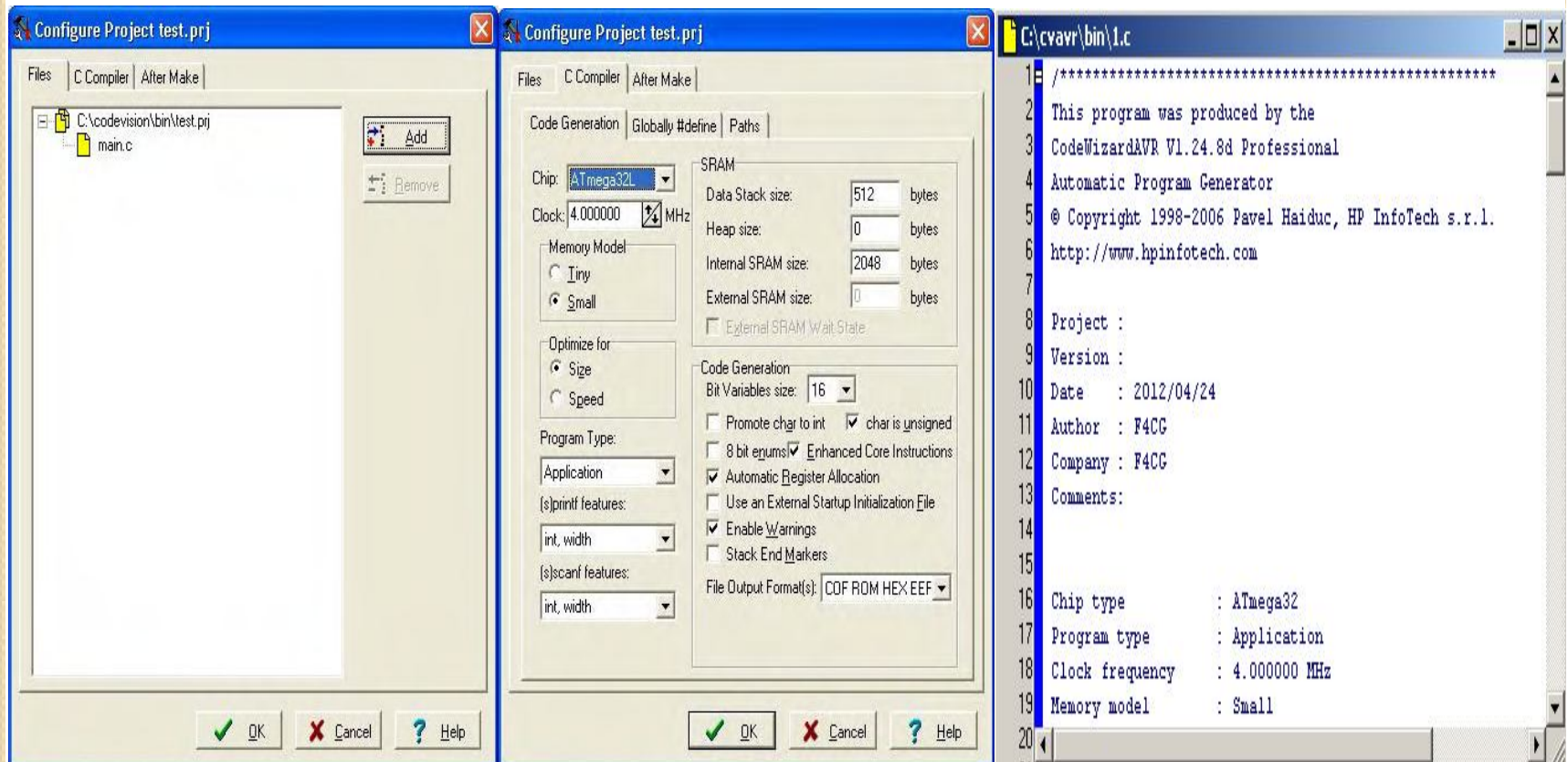
HP InfoTech

C Compiler, Integrated Development Environment,
Automatic Program Generator and In-System Programmer
for the Atmel AVR Family of Microcontrollers

Version 1.24.1d Evaluation
© Copyright 1998-2004 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.ro>
e-mail: office@hpinfotech.ro

Freeware, for evaluation and non-commercial use only

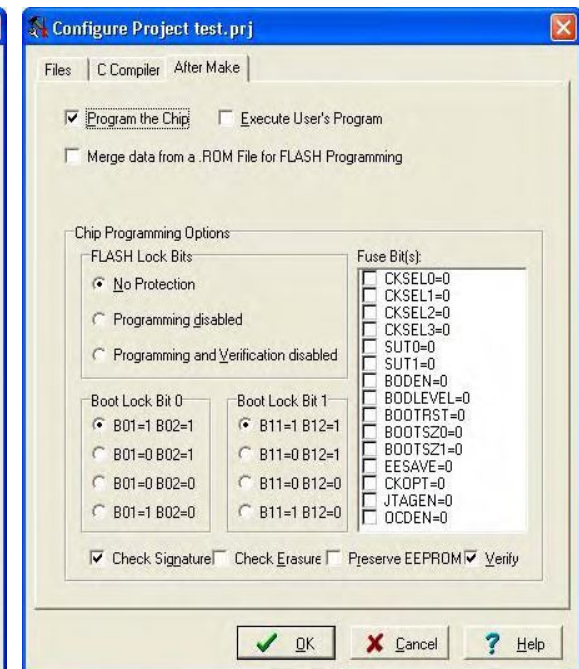
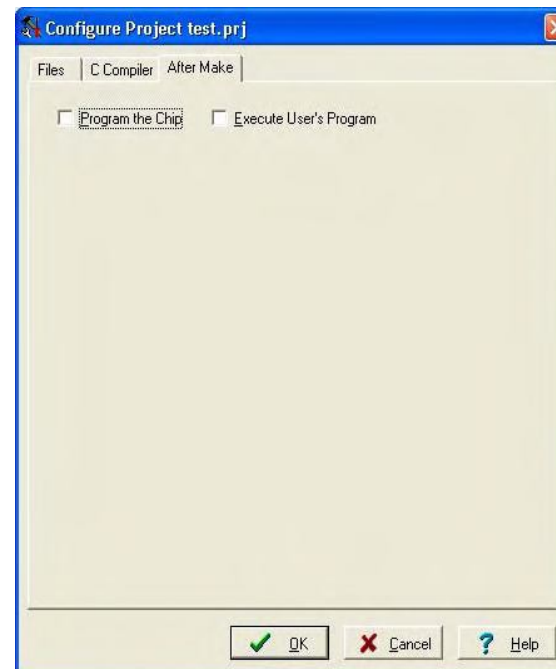
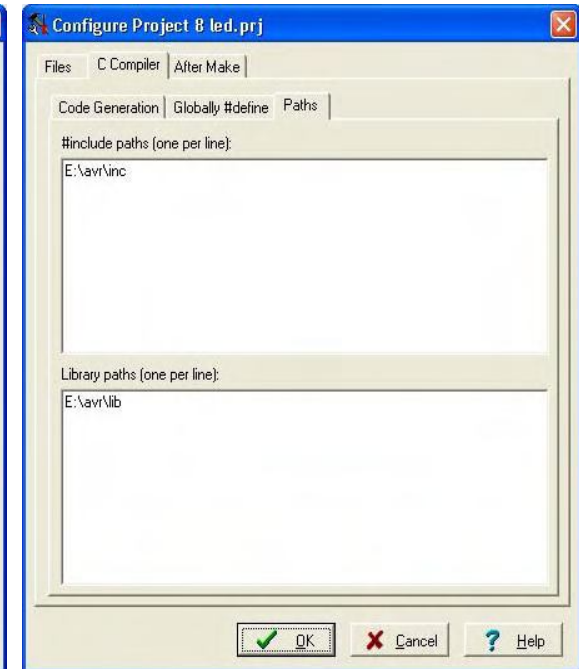
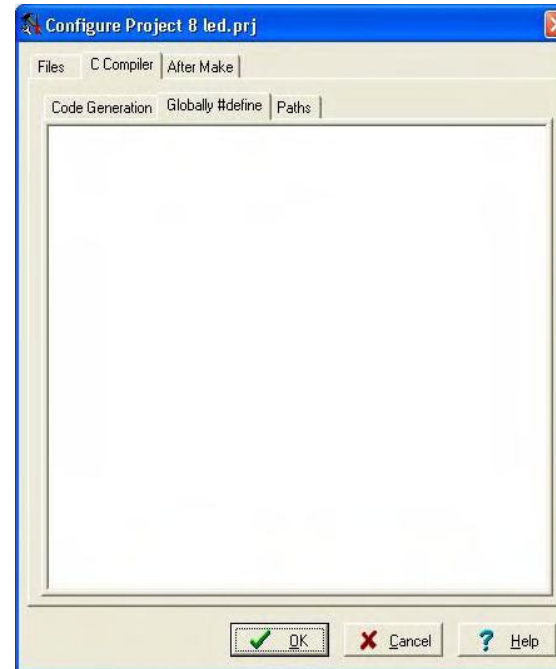
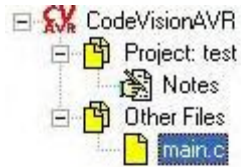
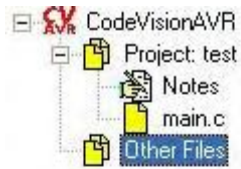
آشنایی با محیط Codevision



میکرو پرو세서

توسعه سیستم
سازمان راجی

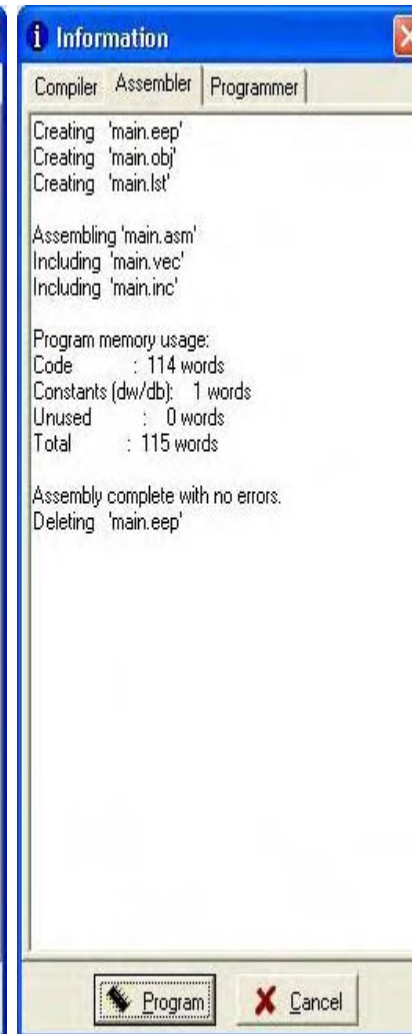
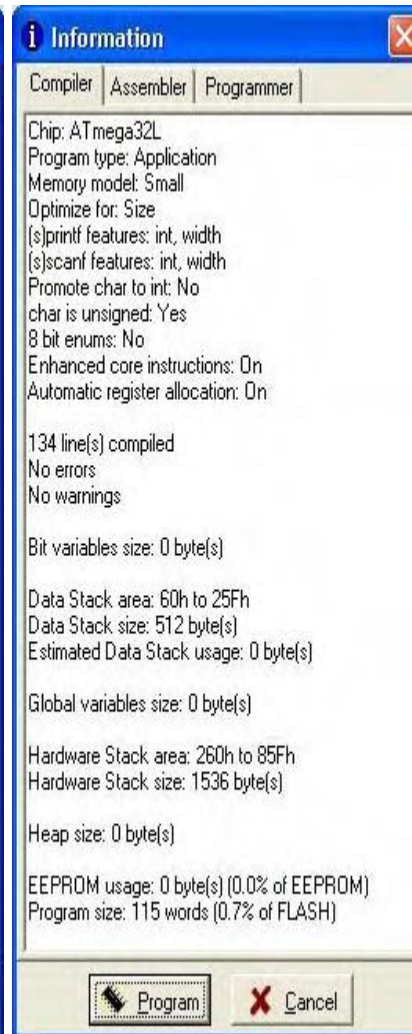
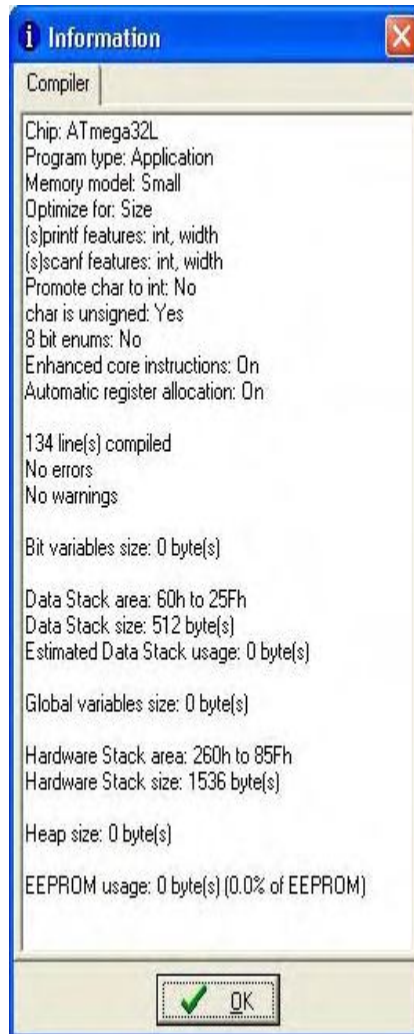
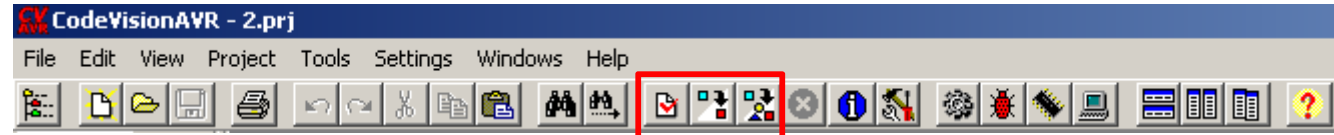
آشنایی با محیط Codevision



میکرو پروسیسور

تیم و تنظیم
سلطان راجی

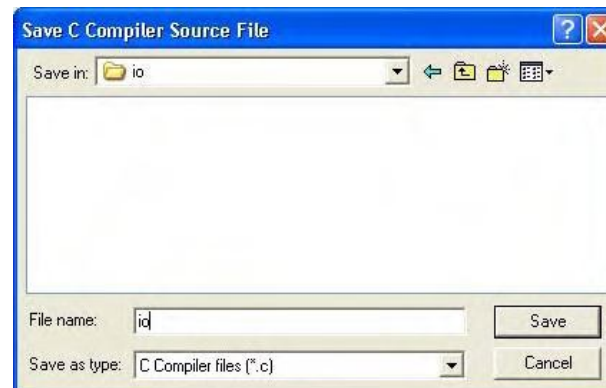
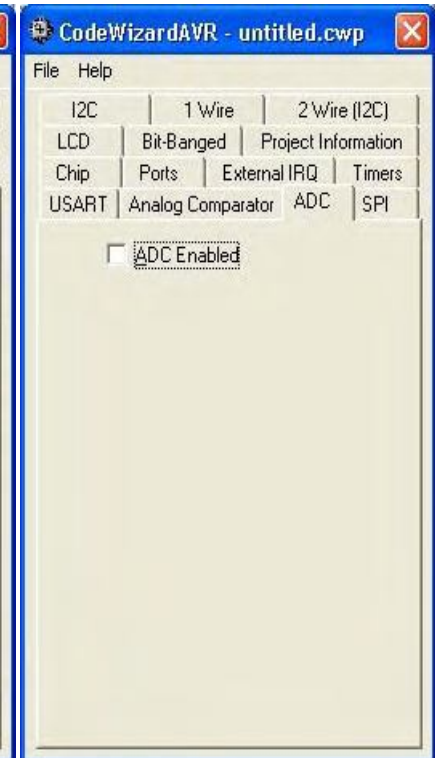
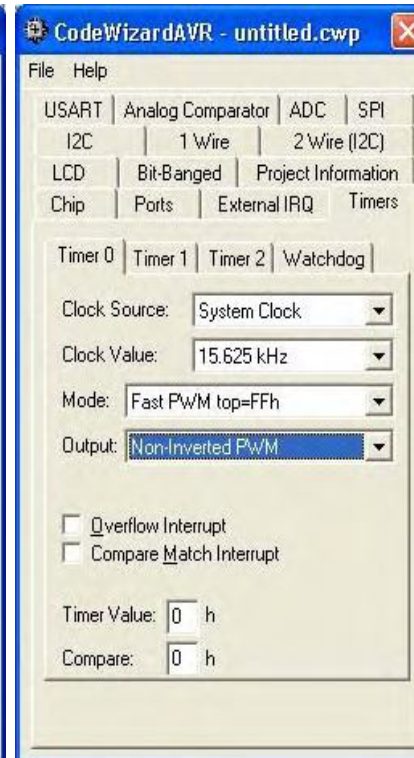
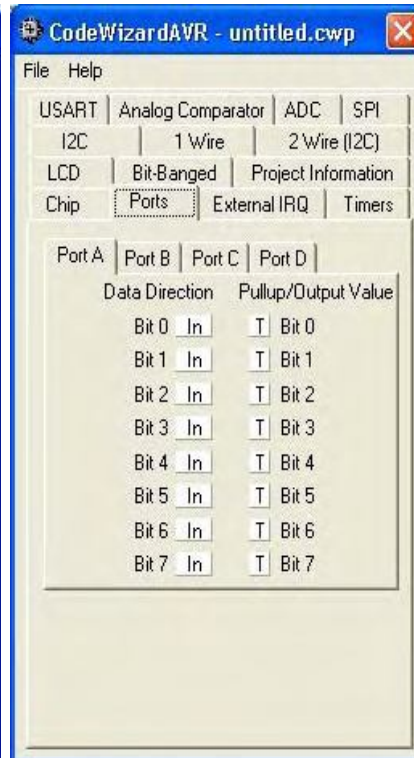
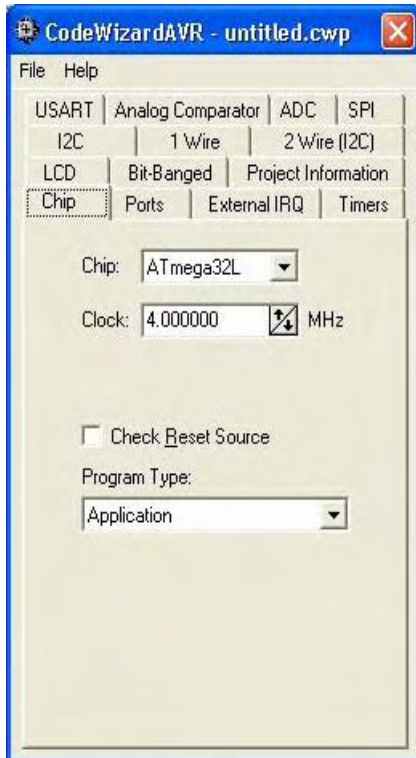
آشنایی با محیط Codevision



میکرو پرو세서

توسعه سیستم
سلطان راجی

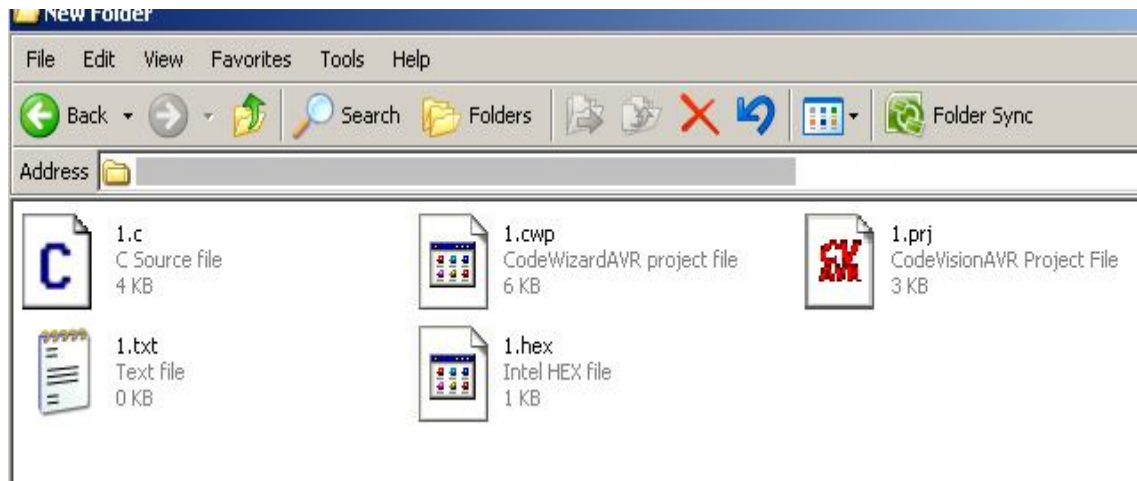
آشنایی با محیط Codevision



میکرو پرو세서

توسعه سیستم
 سلمان راجی

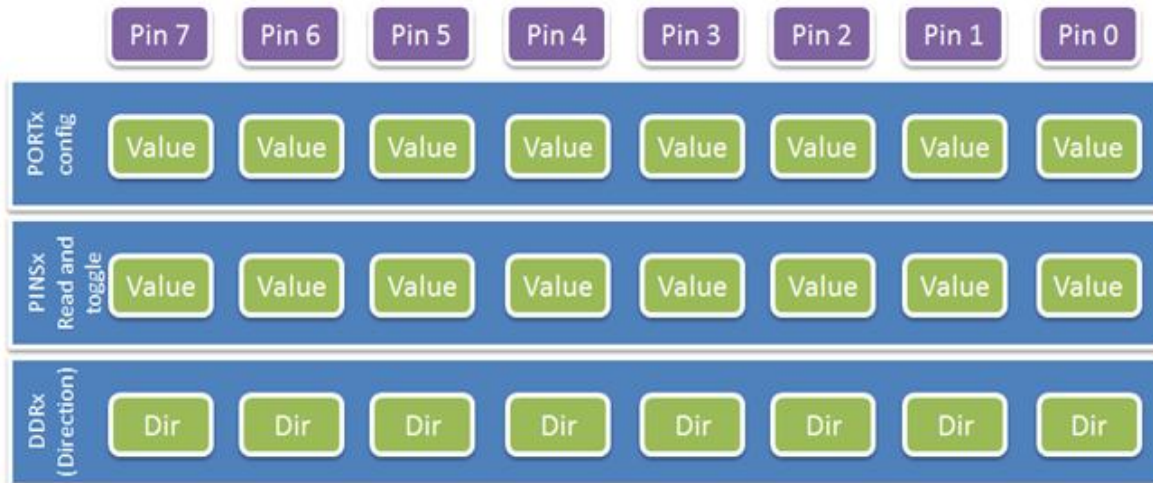
نحوہ برنامه ریزی میکروکنٹرلر



میکرو پروسیسور

تیسو تنظیم
سلان راجی

ساختر پورتی میکروکنٹرلر



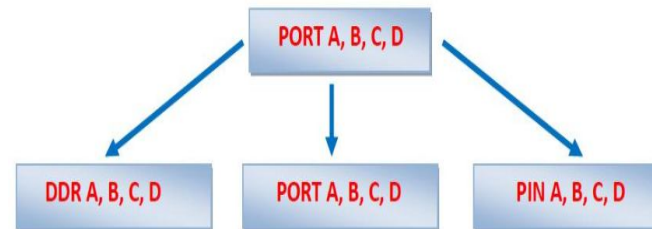
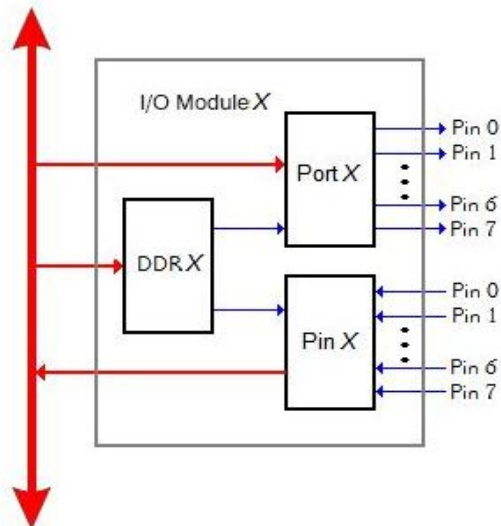
DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

```
int main(void)
{
  DDRA=0b11111111; //set all pins as output
  PORTA=0b10101010; //write ones and zeros to pins
  return 0;
}
```

```
int main(void)
{
  DDRA=0b00000000; //set all pins as output
  x=PINA; //read data on pins
  return 0;
}
```

```
int main(void)
{
  DDRA=0b00000000; //set all pins as output
  PORTA=0b11111111; //set pullup resistors
  x=PINA; //read data on pins
  return 0;
}
```

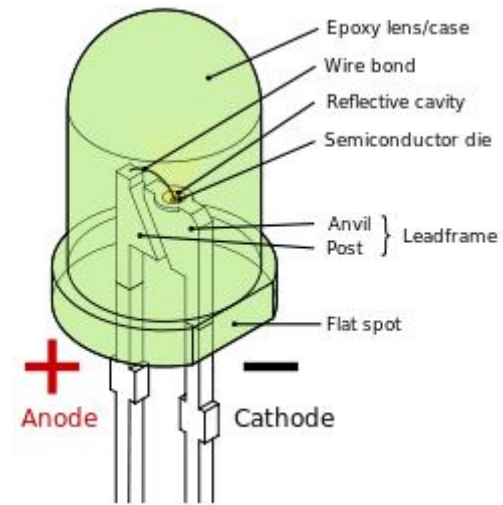
Data Bus



میکرو پروسیسور

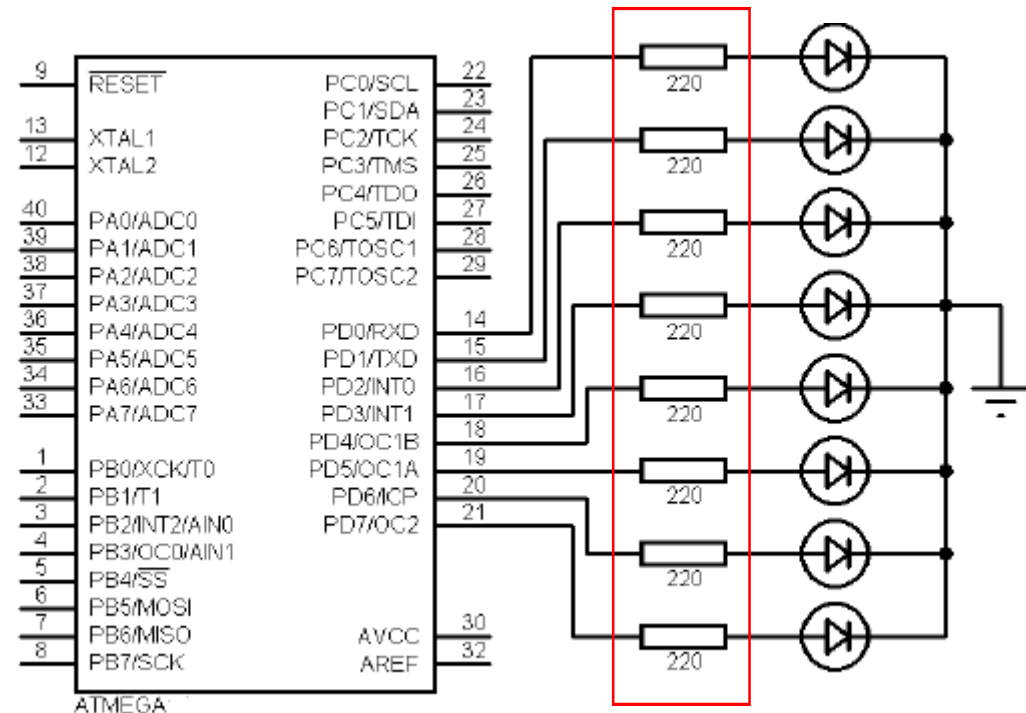
تیسو تنظیم
سلطان راجہ

طراحی یک فلاشر ساده



```
#include<mega32.h> } → تعریف کتابخانه ها
#include<delay.h>
#define xtal 1000000 → تعریف فرکانس کاری
int i; → تعریف متغیر (سراسری)
void main(void)
{
  DDRA=0xff; → خروجی کردن پورت
  while(1) → حلقه تکرار بی نهایت
  {
```

```
for(i=1;i<=128;i=i*2)
{
  PORTD=i;
  delay_ms(100);
}
for(i=64;i>1;i=i/2)
{
  PORTD=i;
  delay_ms(100);
}
```

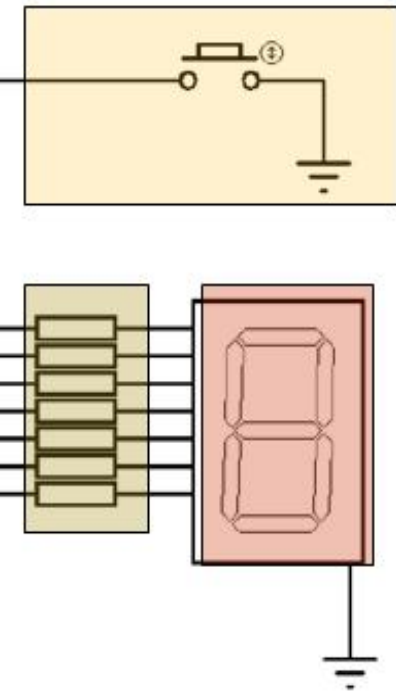
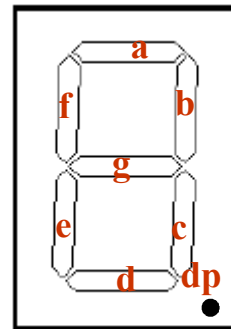
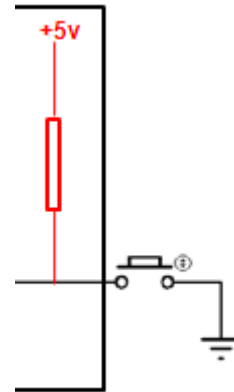


میکرو پرو세서

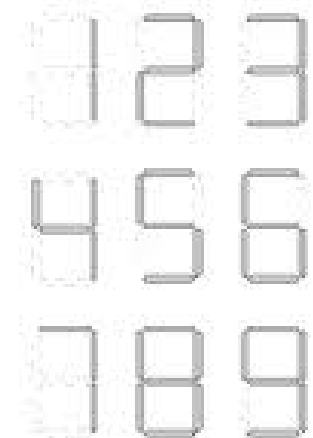
توسعه سیستم
سلطان راجی

طراحی شمارنده تک رقمی

```
#include <mega32.h>
#include < delay.h>
#define xtal 4000000
char digits[16]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,
0x7d,0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};
char p_state,key,i;
void main(void)
{
    DDRD=0xff;
    PORTD=digits[0];
    DDRC=0x00;
    PORTC=0xff;
    while(1)
    {
        key=PINC&0b00000001;
        delay_ms(10);
        if(key==0)
        {
            if(key!=p_state)
            {
                if(i==15)
                {
                    i=0;
                    PORTD=digits[i];
                }
                else
                {
                    i++;
                    PORTD=digits[i];
                    p_state=0;
                }
            }
            else
            {
                p_state=1;
            }
        }
    }
}
```



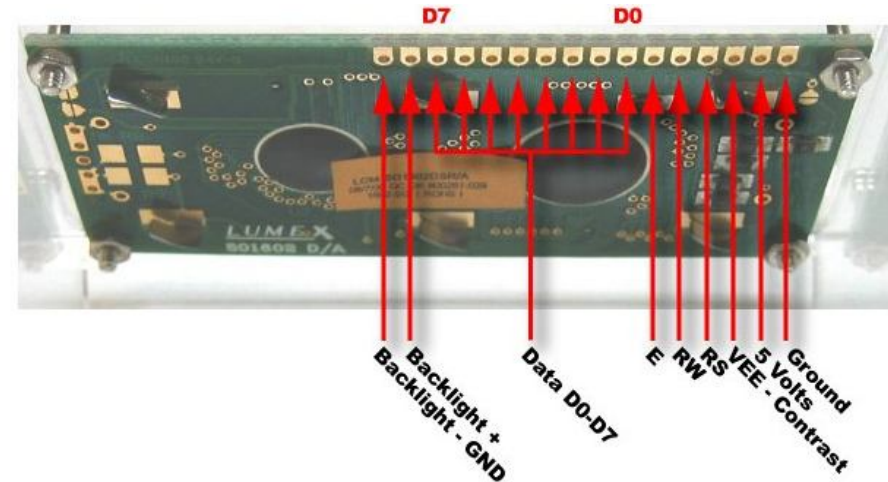
کاراکتر	Dp	g	f	e	d	c	b	a	HEX
0	0	0	1	1	1	1	1	1	0x3F
1	0	0	0	0	0	1	1	0	0x06
2	0	1	0	1	1	0	1	1	0x5B
3	0	1	0	0	1	1	1	1	0x4F
4	0	1	1	0	0	1	1	0	0x66
5	0	1	1	0	1	1	0	1	0x6D
6	0	1	1	1	1	1	0	1	0x7D
7	0	0	0	0	0	1	1	1	0x07
8	0	1	1	1	1	1	1	1	0x7F
9	0	1	1	0	1	1	1	1	0x6F



میکرو پرو세서

توسعه سیستم
سلام راجی

راه اندازی LCD های متنی



```
#include<stdio.h>
#include<mega32.h>
#include<delay.h>
#include<lcd.h>
#define xtal 4000000
#asm
.equ _lcd_port=0x1b; PORTA
#endasm
void main(void)
{
char buffer[10];
char w;
PORTB=0xff;
DDRB=0x00;
lcd_init(16);
lcd_clear();
while(1)
{
w=~PINB;
if(w!=0x00)
{
lcd_clear();
lcd_gotoxy(0,0);
printf(buffer,"number=%d",w);
lcd_puts(buffer);
delay_ms(100);
}
else
{
lcd_clear();
lcd_putsf("number=0");
delay_ms(100);
}
}
}
```

PORTB : 0x18
PORTC : 0x15
PORTD : 0x12

LCD
LM016L

+5v

RS

RW

E

D0

D1

D2

D3

D4

D5

D6

D7

VSS

VDD

VEE

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

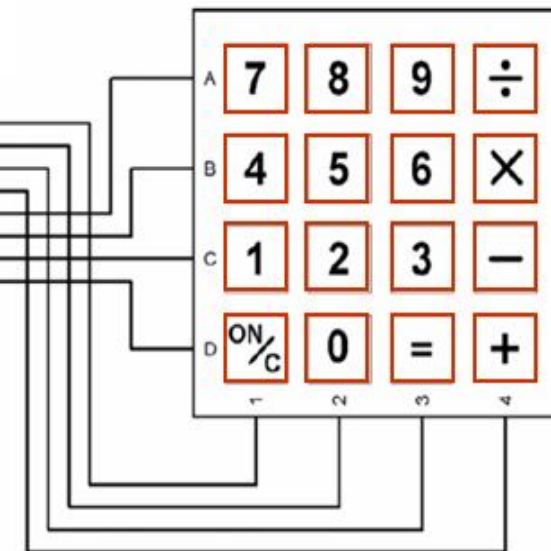
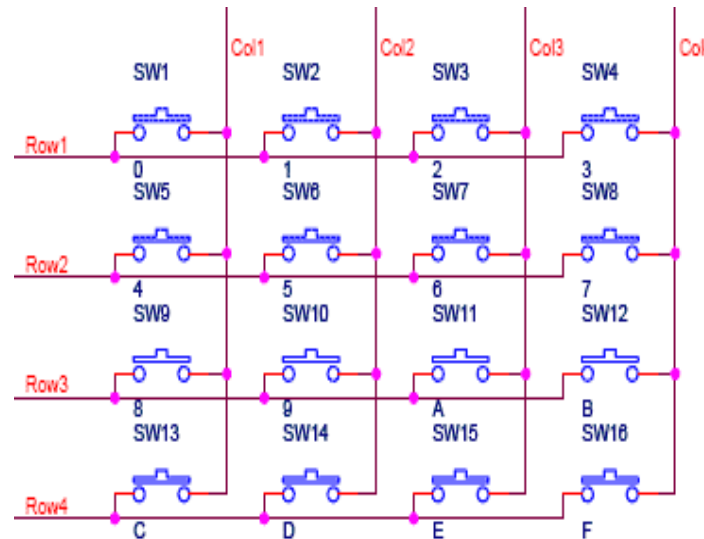
264

265

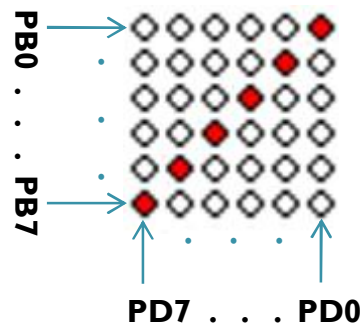
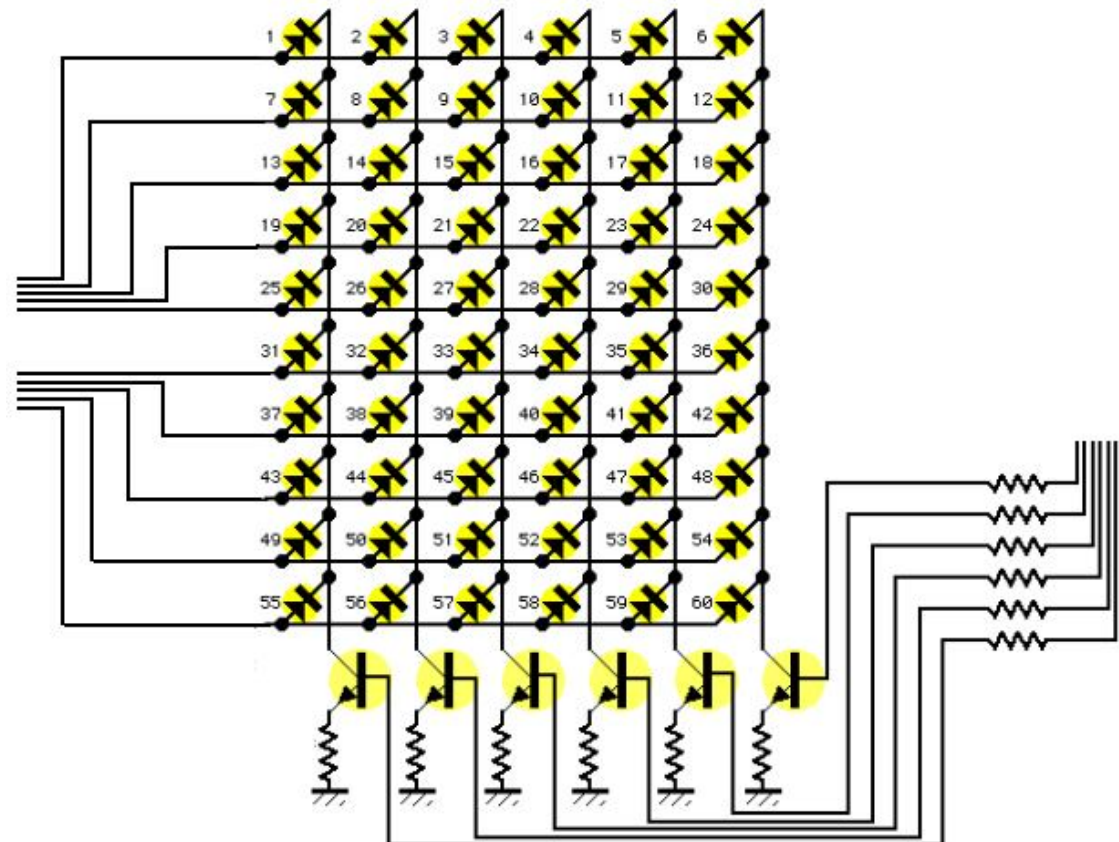
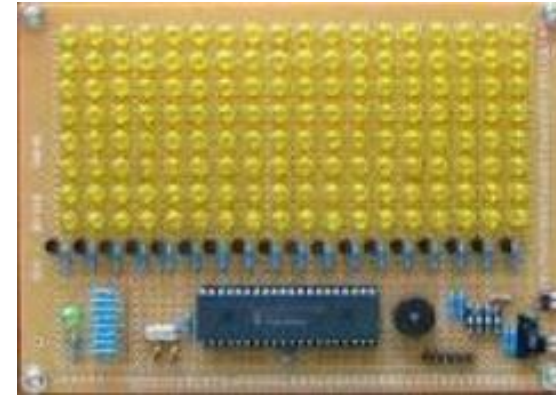
266

راه اندازی کی بورد

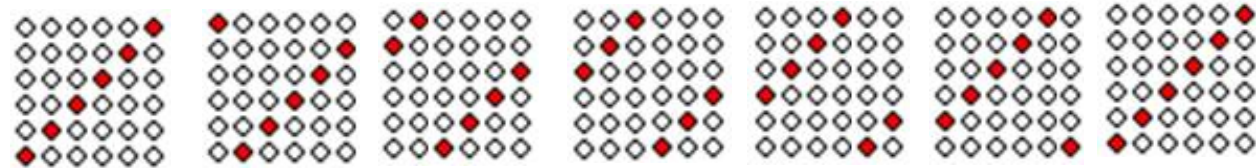
```
#include<mega32.h>
#include< delay.h>
#define xtal 4000000
char key,butnum;
char keytbl[16]={0xee,0xed,0xeb,0xe7,
0xde,0xdd,0xdb,0xd7,0xbe,0xbd,0xbb,
0xb7,0x7e,0x7d,0x7b,0x77};
void main(void)
{
  DDRB=0xff;
  PORTB=0xff;
  while(1)
  {
    DDRC=0x0f;
    PORTC=0xf0;
    delay_us(5);
    key=PINC;
    DDRC=0xf0;
    PORTC=0x0f;
    delay_us(5);
    key=key|PINC;
    delay_ms(10);
    if(key!=0xff)
    {
      for(butnum=0;butnum<16;butnum++)
      {
        if(keytbl[butnum]==key) break;
      }
      if(butnum==16) butnum=0;
      else butnum++;
    }
    else butnum=0;
    PORTB=~butnum;
  }
}
```



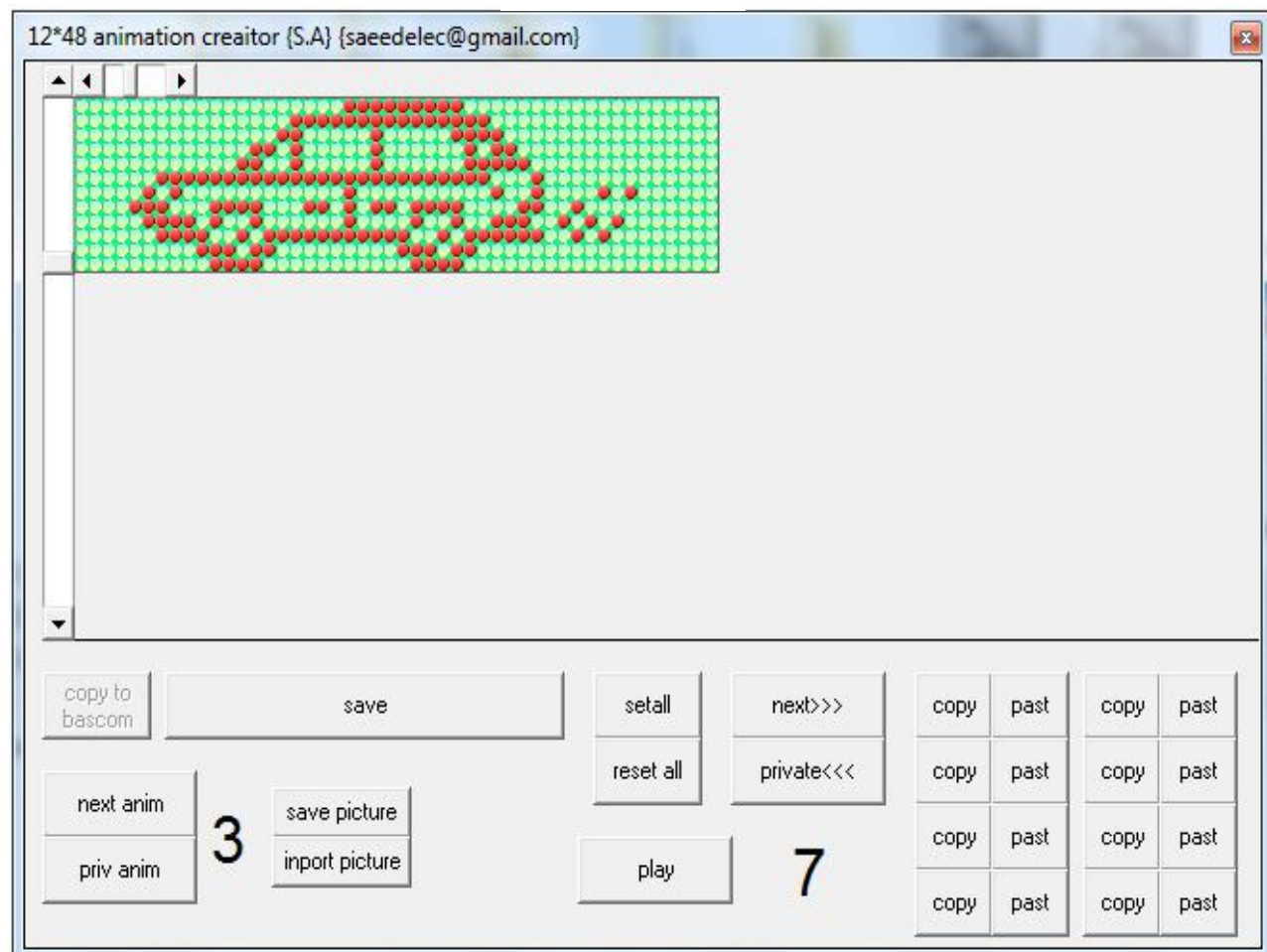
```
#include<stdio.h>
#include<delay.h>
char row[6]={0x01,0x02,0x04,0x08,0x10,0x20};
main()
{
  DDRB=0xff;
  DDRD=0xff;
  int i=1;
  while(1)
  {
    PORTD = i;
    delay_us(50);
    PORTB = row[i];
    delay_ms(15);
    i = i * 2;
    if (i == 64)
    i = 1;
  }
}
```




```
#include<stdio.h>
#include<delay.h>
char row[6]={0x01,0x02,0x04,0x08,0x10,0x20};
main()
{
  DDRB=0xff;
  DDRD=0xff;
  int i=1, k=0, l=0;
  while(1)
  {
    l--;
    if(l== -7)
    l=-1;
    k=l+7;
    if ( k == 6)
    k=0;
    while(1)
    {
      PORTD = i;
      delay_us(50);
      PORTB = row[k];
      delay_ms(15);
      k++;
      if ( k==6)
      k=0;
      i = i * 2;
      if (i == 64)
      break;
    }
  }
}
```



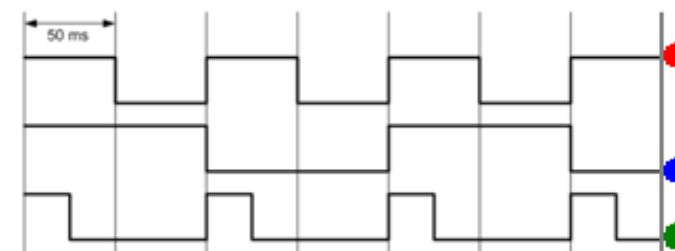
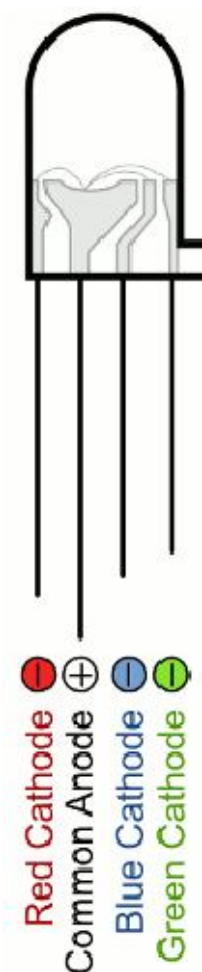
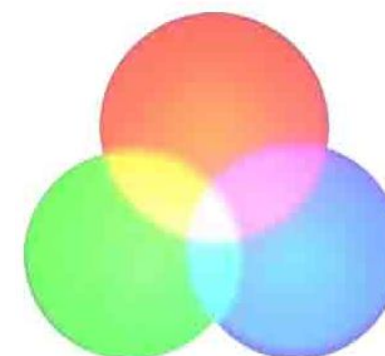
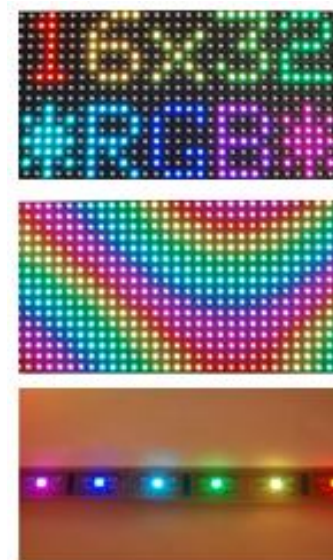
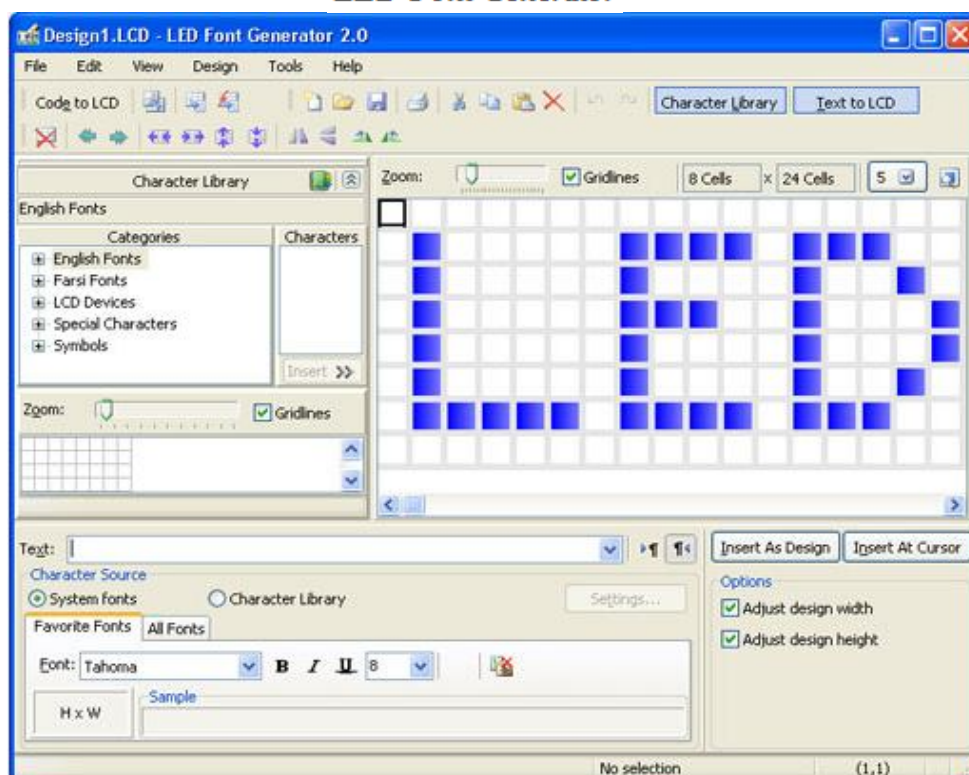
Animation creator



رنگ و فونت در تابلوروان



LED Font Generator



میکرو پرو سسور

تبدیل و تنظیم
سلمان راجی

راه اندازی موتورهای پله‌ای

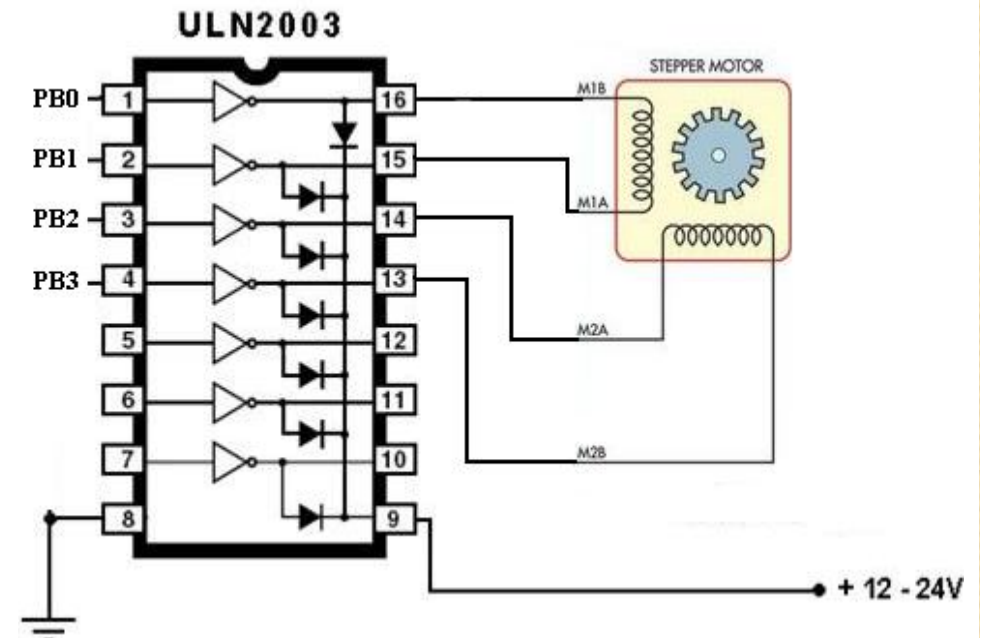
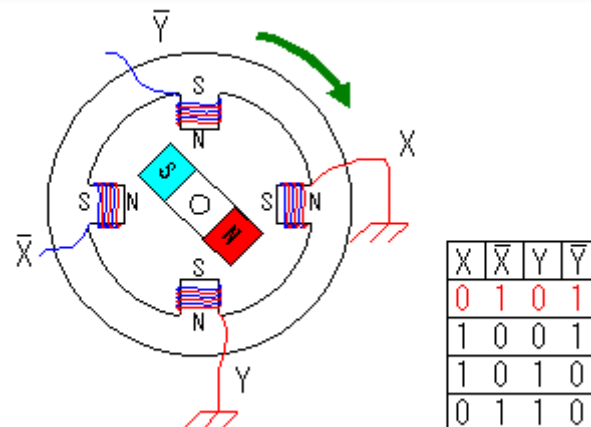
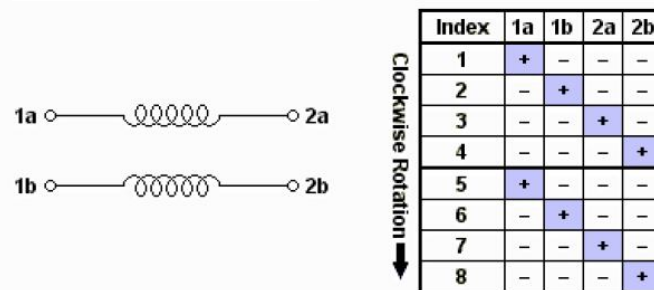
۱- ولتاژ: موتورهای پله‌ای با ولتاژ کارهای مختلفی ساخته میشوند. رایج ترین آنها ۵، ۱۲ و ۲۴ ولت است. همواره اعمال ولتاژ کمی بیشتر از ولتاژ نامی موتور برای افزایش گشتاور در کارکرد بهتر موتور توصیه می شود.

۲- جریان: مقدار جریانی است که، وقتی موتور در حال کار است در سیم پیچها جاری میشود. در ابتدای هر پله، جریان بیشتری از سیم پیچها می گذرد.

۳- گشتاور: گشتاور موتور یکی از مهمترین عوامل در انتخاب یک موتور است. گشتاور موتور معرف مقدار ماکزیمم گشتاوری است که می توان به موتور اعمال کرد بدون اینکه، حرکت موتور دچار اختلال شود.

۴- زاویه گام: زاویه گام زاویه ای است که شفت موتور با اعمال یک پالس طی می کند، زاویه گام معمولاً بر حسب درجه بیان می شود. موتورهایی با زاویه گام های ۰.۹، ۱.۵، ۷/۵، ۳/۶، ۱/۸، ۰/۷۲ درجه ساخته شده است.

```
#include<stdio.h>
#include<delay.h>
main()
{
  DDRB=0xff;
  while(1)
  {
    PORTB=0xaa;
    delay_ms(200);
    PORTB=0x10;
    delay_ms(200);
  }
}
```



میت

تیم و تنظیم
سلطان راجی

میکرو پرو سسور

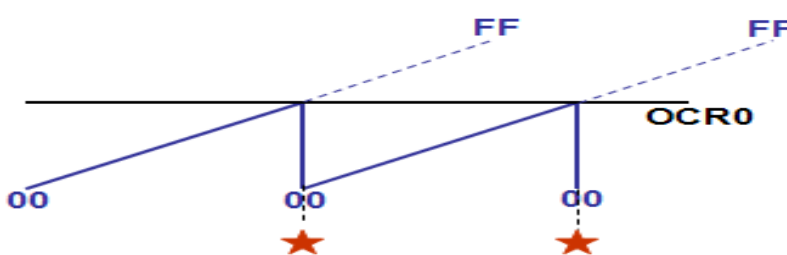
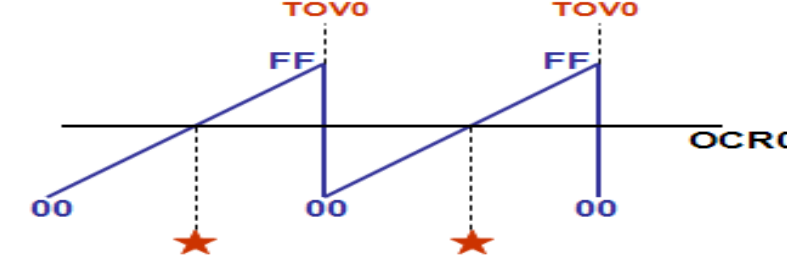
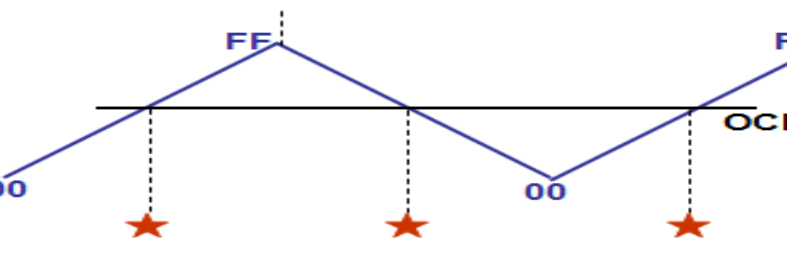
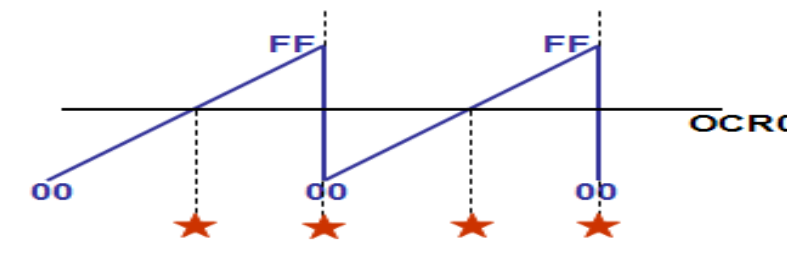
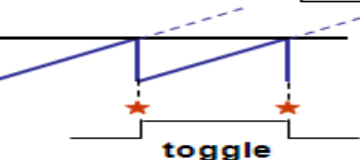
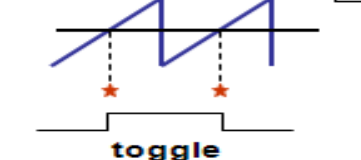
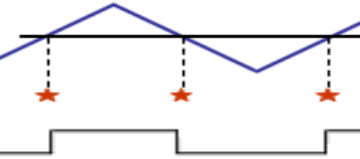
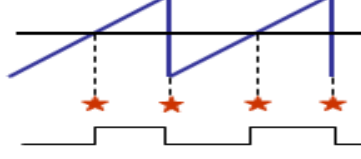


میکرو پرو سسور

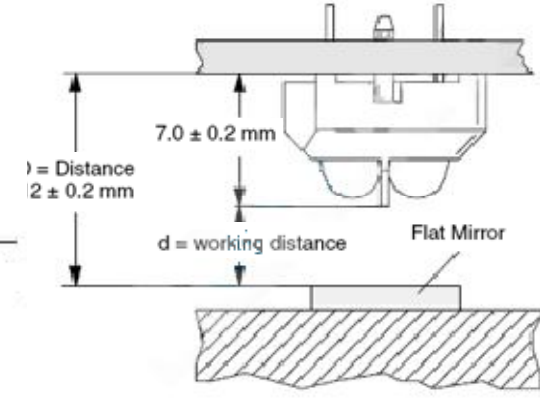
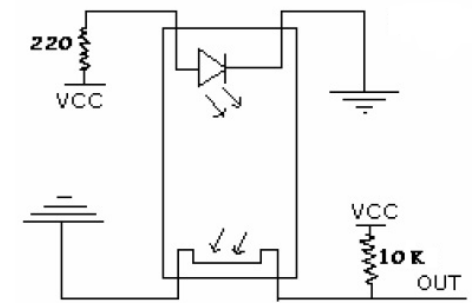
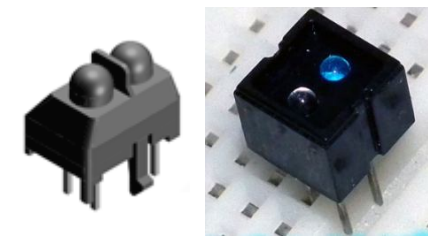
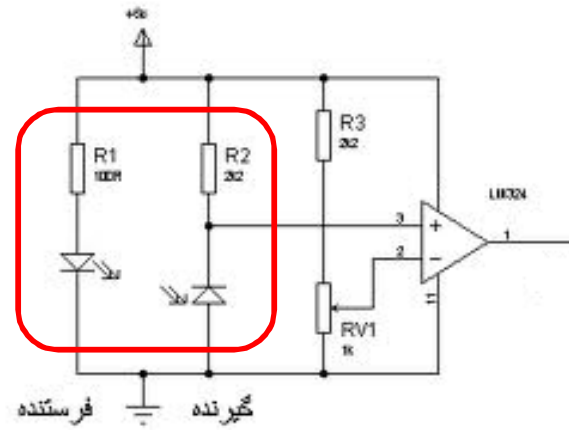
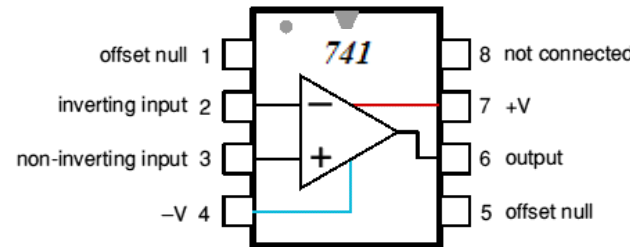
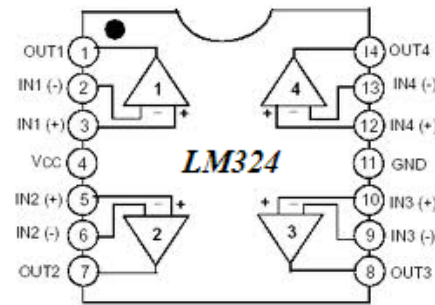
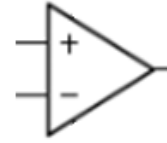
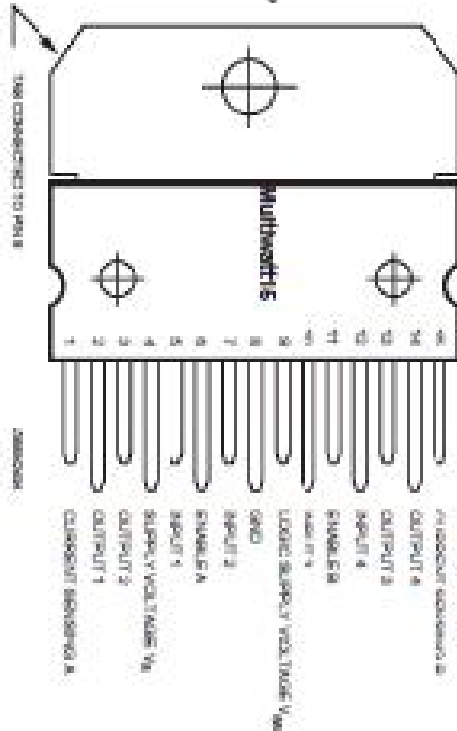
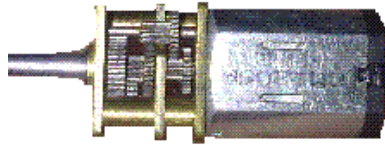
میکرو پرو سسور

میکرو پرو سسور

میکرو پرو سسور

CTC 	Normal 
PWM, Phase correct 	Fast PWM 
CTC  <pre>#include<mega32.h> Void main(void){ DDRB=0xff; TCNT0=0x00; OCR0=0x7c; TCCR0=0x1c; While(1); }</pre>	Normal  <pre>#include<mega32.h> Void main(void){ DDRB=0xff; TCNT0=0x00; OCR0=0x7c; TCCR0=0x11; While(1); }</pre>
PWM, Phase correct  <pre>#include<mega32.h> Void main(void){ DDRB=0xff; TCNT0=0x00; OCR0=0x7c; TCCR0=0x69; While(1); }</pre>	Fast PWM  <pre>#include<mega32.h> Void main(void){ DDRB=0xff; TCNT0=0x00; OCR0=0x7c; TCCR0=0x62; While(1); }</pre>

قطعات بکاررفته در ربات های مسیریاب



میکرو پرو سوسر

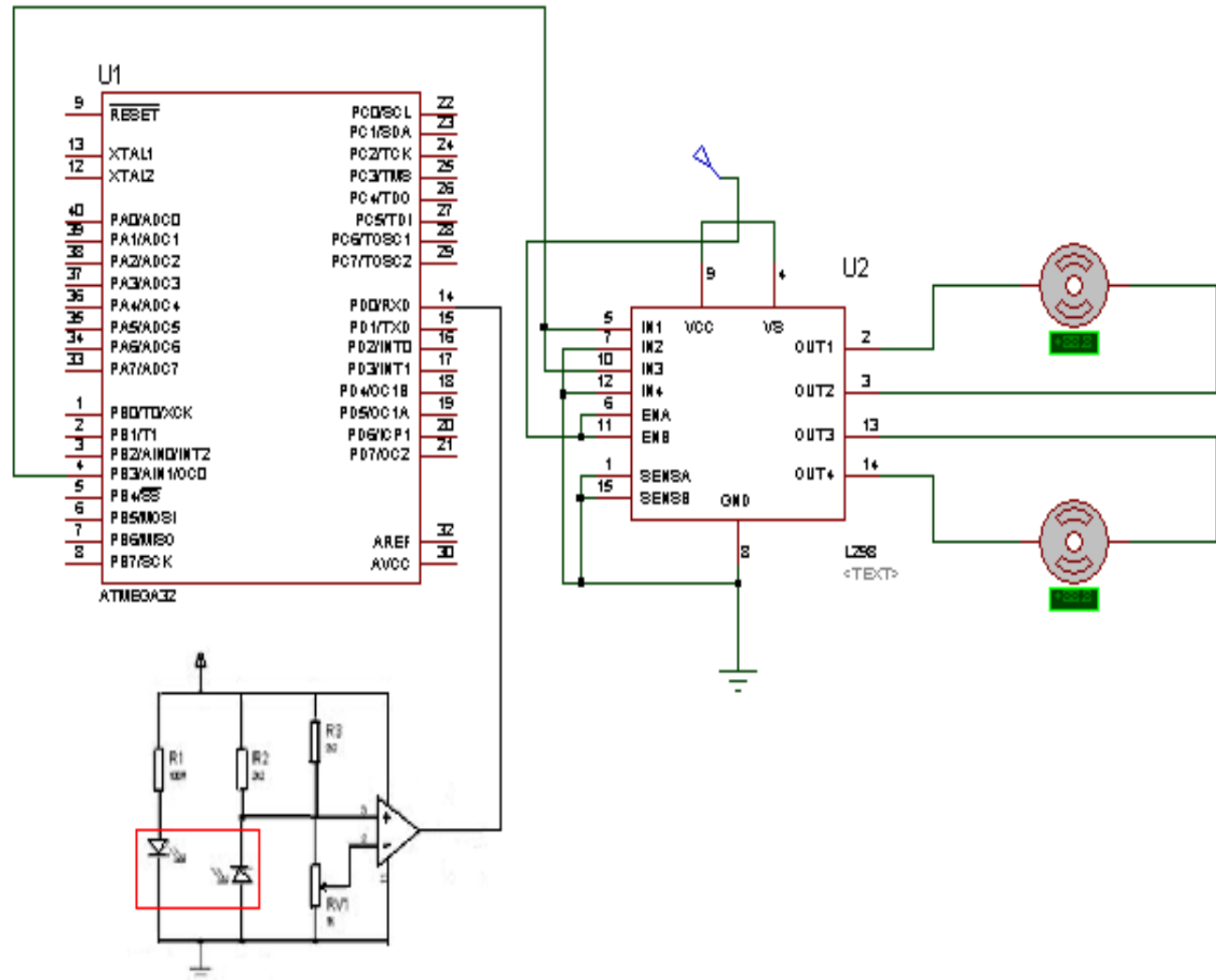
توسعه تخصصی
سلطان راجی

تنظیم سرعت اتوماتیک در ربات

Speed: High

Speed: Low

```
#include<stdio.h>
#include<delay.h>
main()
{
    DDRD=0x00;
    DDRB=0xff;
    TCCR0=0x52;
    OCR0=128;
    while(1)
    {
        x=PIND.0;
        if ( x == 0)
        {
            OCR0=5;
        }
        if ( x == 1)
        {
            OCR0=250;
        }
    }
}
```



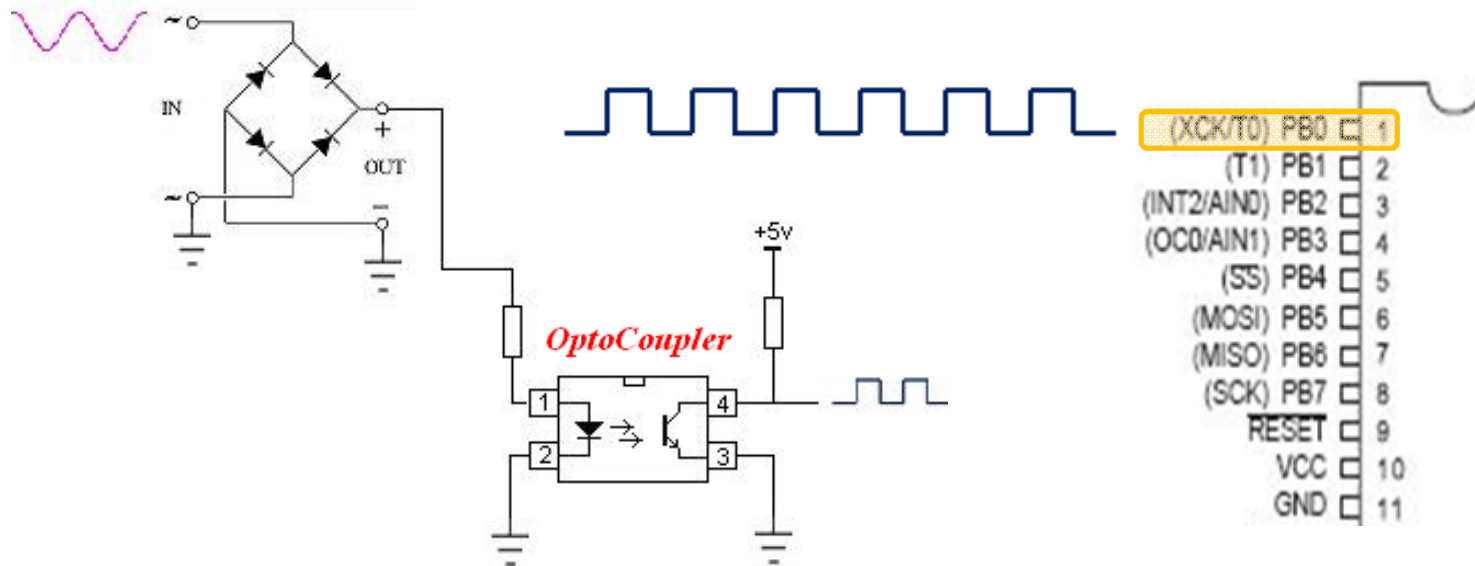
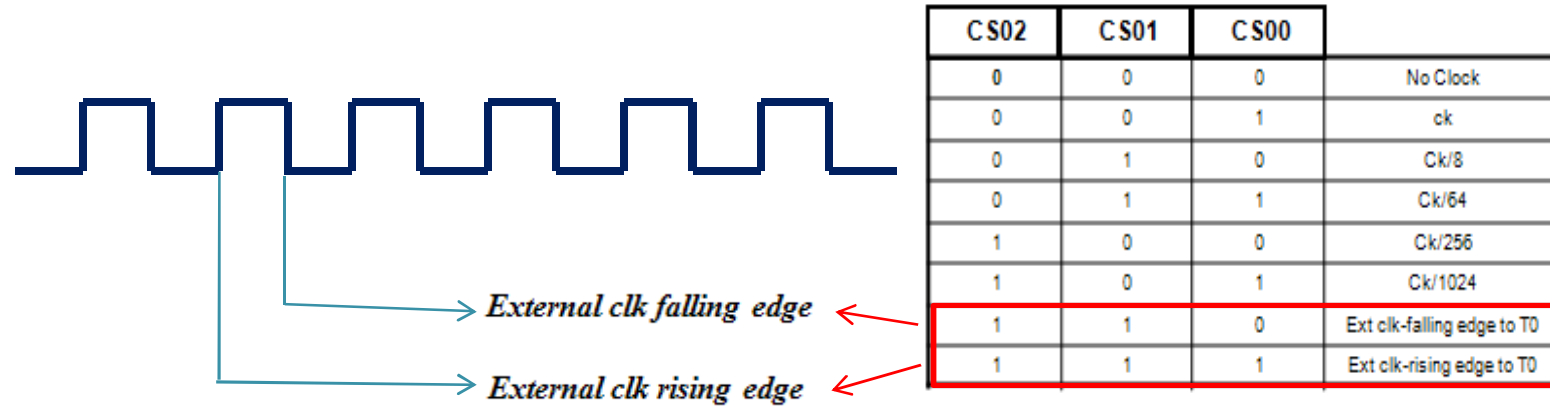
میکرو پرو سسور

توسعه سیستم
سلطان راجی

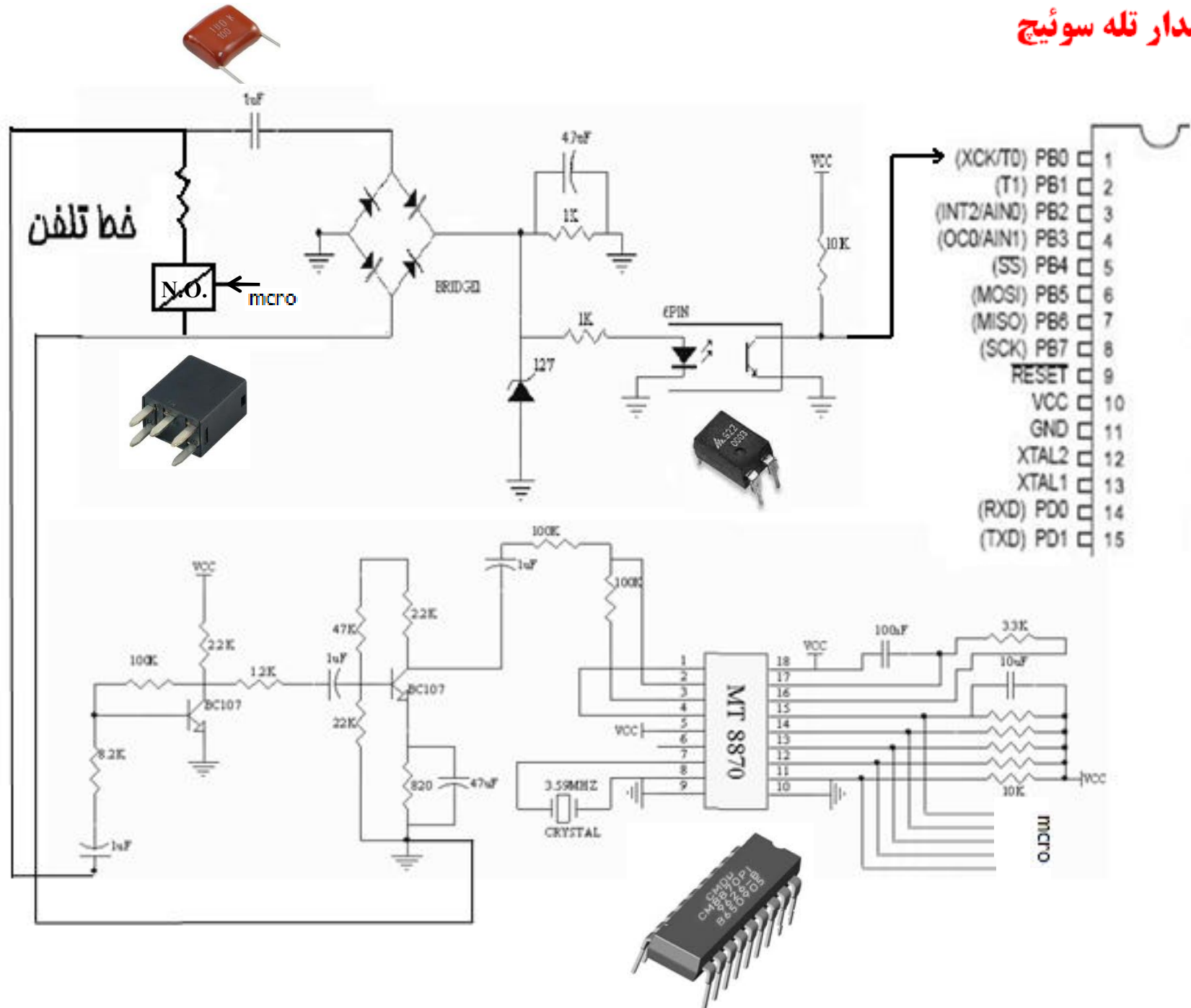
تایمر صفر بعنوان شمارنده پالس خارجی

TCNT0:	TCNT07	TCNT06	TCNT05	TCNT04	TCNT03	TCNT02	TCNT01	TCNT00
---------------	--------	--------	--------	--------	--------	--------	--------	--------

TCCR0:	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
	0	0	0	0	0	CS02	CS01	CS00



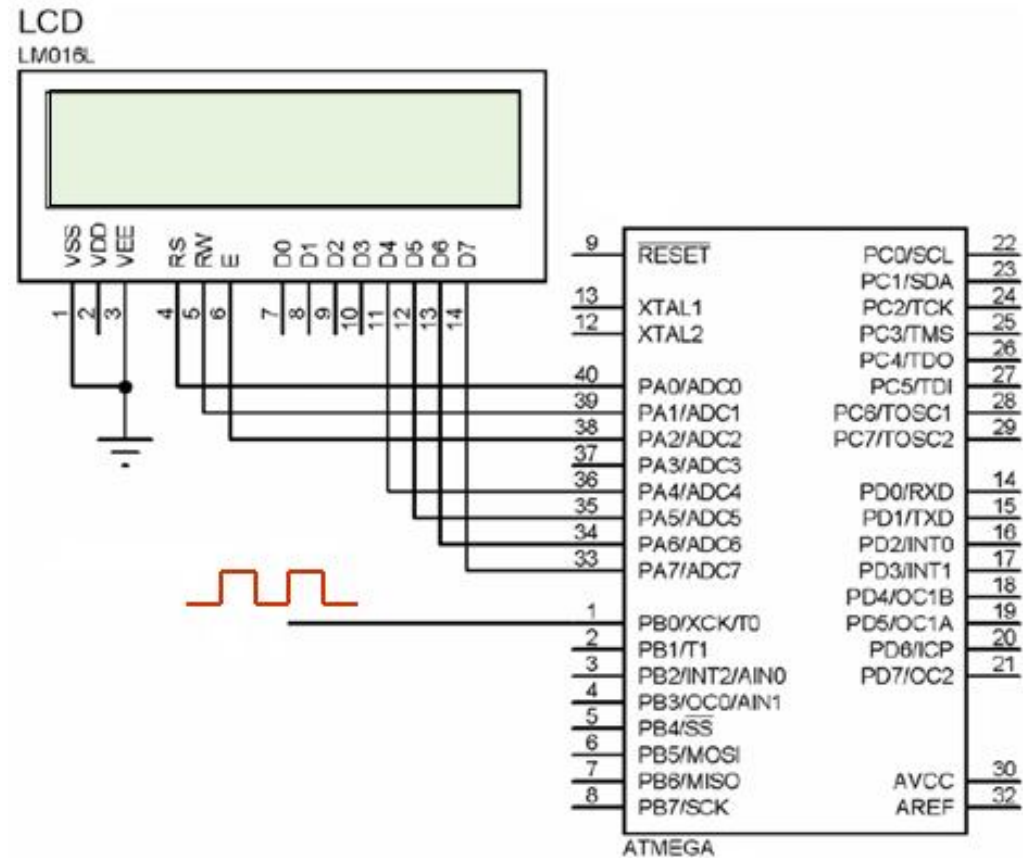
مدار تله سوئیچ



میکرو کنترلر

توسعه سیستم
سلطان راجی


```
#include<mega32.h>
#include< delay.h>
#include< stdio.h>
#include< lcd.h>
#define xtal 1000000
#asm
.equ __lcd_port=0x1b;
#endasm
char buffer[20];
int f=0;
Void main(void)
{
    TCNT0=0x00;
    OCR0=0x00;
    TCCR0=0x06;
    while(1)
    {
        delay_ms(1000);
        TCCR0=0x00;
        f=f+TCNT0;
        sprintf(buffer,"frequency=%d",f);
        lcd_clear();
        lcd_puts(buffer);
        TCCR0=0x06;
        TCNT0=0;
    }
}
```



Maximum Frequency: 255Hz

TCNT0:	TCNT07	TCNT06	TCNT05	TCNT04	TCNT03	TCNT02	TCNT01	TCNT00
	0	0	0	0	0	0	0	0
	Increase							
255								

وقفه تایمر صفر

TIMSK:	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
--------	-------	-------	--------	--------	--------	-------	-------	-------

فعال ساز وقفه تطابق مقایسه ←

فعال ساز وقفه سرریز ←

شماره	نام در CodeVision	آدرس	معنی وقفه	توضیح
۱		\$000	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
۲	EXT_INT0	\$002	INT0	External Interrupt Request 0
۳	EXT_INT1	\$004	INT1	External Interrupt Request 1
۴	TIM2_COMP	\$006	TIMER2 COMP	Timer/Counter2 Compare Match
۵	TIM2_OVF	\$008	TIMER2 OVF	Timer/Counter2 Overflow
۶	TIM1_CAPT	\$00A	TIMER1 CAPT	Timer/Counter1 Capture Event
۷	TIM1_COMPA	\$00C	TIMER1 COMPA	Timer/Counter1 Compare Match A
۸	TIM1_COMPB	\$00E	TIMER1 COMPB	Timer/Counter1 Compare Match B
۹	TIM1_OVF	\$010	TIMER1 OVF	Timer/Counter1 Overflow
۱۰	TIM0_OVF	\$012	TIMER0 OVF	Timer/Counter0 Overflow
۱۱	SPI_STC	\$014	SPI STC	Serial Transfer Complete
۱۲	USART_RXC	\$016	USART, RXC	USART, Rx Complete
۱۳	USART_DRE	\$018	USART, UDRE	USART Data Register Empty
۱۴	USART_TXC	\$01A	USART, TXC	USART, Tx Complete
۱۵	ADC_INT	\$01C	ADC	ADC Conversion Complete
۱۶	EE_RDY	\$01E	EE_RDY	EEPROM Ready
۱۷	ANA_COMP	\$020	ANA_COMP	Analog Comparator
۱۸	TWI	\$022	TWI	Two-wire Serial Interface
۱۹	EXT_INT2	\$024	INT2	External Interrupt Request 2
۲۰	TIM0_COMP	\$026	TIMER0 COMP	Timer/Counter0 Compare Match
۲۱	SPM_READY	\$028	SPM_RDY	Store Program Memory Ready

غیر فعال نمودن وقفه سراسری

#asm

cli (#asm("cli"))

#endasm

فعال سازی وقفه سراسری

#asm

sei (#asm("sei"))

#endasm

SREG: I T H S V N Z C

Interrupt [TIM0_OVF] void timer0_ovf_isr(void)

```
{
.
.
}
```

Interrupt [TIM0_COMP] void timer0_comp_isr(void)

```
{
.
.
}
```

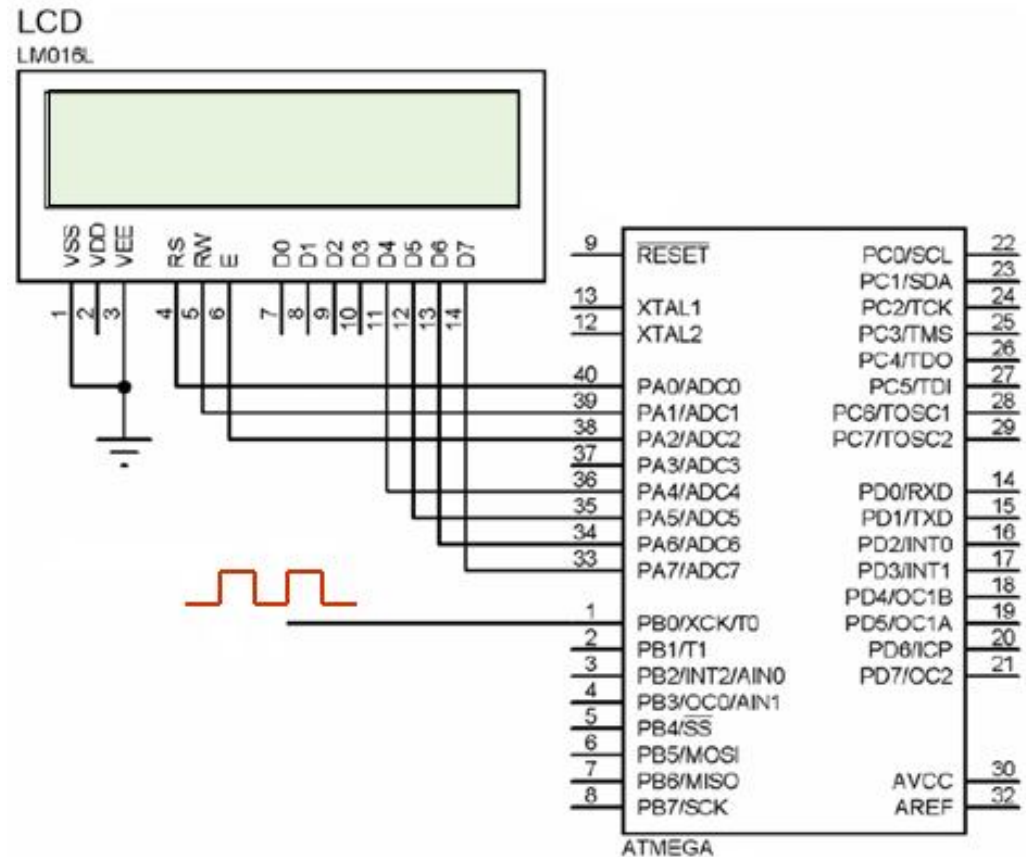
محل تعریف زیر برنامه وقفه

قبل از تعریف تابع اصلی (main) و بعد از تعریف متغیرهای سراسری

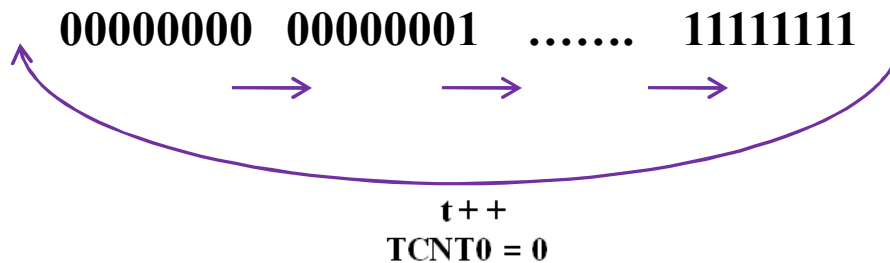
میکرو پرو세서

توسعه سیستم
سلطان راجی

```
#include<mega32.h>
#include<delay.h>
#include<stdio.h>
#include<lcd.h>
#define xtal 8000000
#asm
.equ __lcd_port=0x1b;
#endasm
int t;
long int f;
char buffer[20];
interrupt [TIM0_OVF] void f_m(void)
{
    t++;
}
Void main(void) {
    TCNT0=0x00;
    OCR0=0x00;
    TCCR0=0x00;
    TIMSK=0x01;
    lcd_init(16);
    while(1)
    {
        TCCR0=0x06;
        #asm("sei")
        delay_ms(1000);
        #asm("cli");
        f=t*256+TCNT0;
        sprintf(buffer,"frequency=%d",f);
        lcd_clear();
        lcd_puts(buffer);
        TCCR0=0x00;
        t=0;
        TCNT0=0;
    }
}
```



TCNT0



TCCRIA:	COMIAI	COMIA0	COMIBI	COMIB0	FOCIA	FOCIB	WGM1I	WGM10
---------	--------	--------	--------	--------	-------	-------	-------	-------

TCCRIB:	ICNCI	ICESI	-	WGM13	WGM12	CS12	CS1I	CS10
---------	-------	-------	---	-------	-------	------	------	------

TCNTI:	TCNT IF	TCNT IE	TCNT ID	TCNT IC	TCNT IB	TCNT IA	TCNT I9	TCNT I8	TCNT I7	TCNT I6	TCNT I5	TCNT I4	TCNT I3	TCNT I2	TCNT I1	TCNT I0
--------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

OCRIA:	OCRA IF	OCRA IE	OCRA ID	OCRA IC	OCRA IB	OCRA IA	OCRA I9	OCRA I8	OCRA I7	OCRA I6	OCRA I5	OCRA I4	OCRA I3	OCRA I2	OCRA I1	OCRA I0
--------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

OCRIB:	OCRB IF	OCRB IE	OCRB ID	OCRB IC	OCRB IB	OCRB IA	OCRB I9	OCRB I8	OCRB I7	OCRB I6	OCRB I5	OCRB I4	OCRB I3	OCRB I2	OCRB I1	OCRB I0
--------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

ICRI:	ICR IF	ICR IE	ICR ID	ICR IC	ICR IB	ICR IA	ICR I9	ICR I8	ICR I7	ICR I6	ICR I5	ICR I4	ICR I3	ICR I2	ICR I1	ICR I0
-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

	WGM13	WGM12	WGM11	WGM10	Mode کاری	TOP	TOV=1
*	*	*	*	*	Normal	0xFFFF	0xFFFF
۱	*	*	*	۱	PWM, Phase Correct, 8-bit	0x00FF	0
۲	*	*	۱	*	PWM, Phase Correct, 9-bit	0x01FF	0
۳	*	*	۱	۱	PWM, Phase Correct, 10-bit	0x03FF	0
۴	*	۱	*	*	CTC	OCR1A	0xFFFF
۵	*	۱	*	۱	Fast PWM, 8-bit	0x00FF	TOP
۶	*	۱	۱	*	Fast PWM, 9-bit	0x01FF	TOP
۷	*	۱	۱	۱	Fast PWM, 10-bit	0x03FF	TOP
۸	۱	*	*	*	PWM, Phase and Frequency Correct	ICR1	0
۹	۱	*	*	۱	PWM, Phase and Frequency Correct	OCR1A	0
۱۰	۱	*	۱	*	PWM, Phase Correct	ICR1	0
۱۱	۱	*	۱	۱	PWM, Phase Correct	OCR1A	0
۱۲	۱	۱	*	*	CTC	ICR1	0xFFFF
۱۳	۱	۱	*	۱	رزرو شده	-	-
۱۴	۱	۱	۱	*	Fast PWM	ICR1	TOP
۱۵	۱	۱	۱	۱	Fast PWM	OCR1A	TOP

(XCK/T0) PB0	1
(T1) PB1	2
(INT2/AIN0) PB2	3
(OC0/AIN1) PB3	4
(SS) PB4	5
(MOSI) PB5	6
(MISO) PB6	7
(SCK) PB7	8
RESET	9
VCC	10
GND	11
XTAL2	12
XTAL1	13
(RXD) PD0	14
(TXD) PD1	15
(INT0) PD2	16
(INT1) PD3	17
(OC1B) PD4	18
(OC1A) PD5	19
(ICP1) PD6	20

Normal Mode،CTC Mode

COM1A1/COM1B1	COM1A0/COM1B0	وضعیت پین OC1A یا OC1B
0	0	خیر فعال (I/O معمولی)
0	1	Toggle در وضعیت تطابق
1	0	Clear در وضعیت تطابق
1	1	Set در وضعیت تطابق

Phase & Frequency Correct Mode،Phase Correct Mode

COM1A1/COM1B1	COM1A0/COM1B0	وضعیت پین OC1A یا OC1B
0	0	خیر فعال (I/O معمولی)
0	1	اگر $WGM1[3:0] = 9, 14$ باشد: Toggle پین OC1A در وضعیت تطابق و OC1B پین I/O معمولی برای دیگر حالت های $WGM1[3:0]$: خیر فعال (I/O معمولی)
1	0	Clear در وضعیت تطابق و شمارش صعودی. Set در وضعیت تطابق و شمارش نزولی
1	1	Set در وضعیت تطابق و شمارش صعودی. Clear در وضعیت تطابق و شمارش نزولی

Fast PWM Mode

COM1A1/COM1B1	COM1A0/COM1B0	وضعیت پین OC1A یا OC1B
0	0	خیر فعال (I/O معمولی)
0	1	اگر $WGM1[3:0] = 15$ باشد: Toggle پین OC1A در وضعیت تطابق و OC1B پین I/O معمولی برای دیگر حالت های $WGM1[3:0]$: خیر فعال (I/O معمولی)
1	0	Clear در وضعیت تطابق و Set در وضعیت TOP (PWM خیر معکوس)
1	1	Set در وضعیت تطابق و Clear در وضعیت TOP (PWM معکوس)

CS02	CS01	CS00	وضعیت منبع کلاک تایمر
0	0	0	بدون کلاک (متوقف)
0	0	1	کلاک سیستم (بدون تقسیم)
0	1	0	کلاک سیستم /8
0	1	1	کلاک سیستم /16
1	0	0	کلاک سیستم /256
1	0	1	کلاک سیستم /1024
1	1	0	لبه ی پایین رونده ی پالس خارجی (T1)
1	1	1	لبه ی بالا رونده ی پالس خارجی (T1)

TIFR:

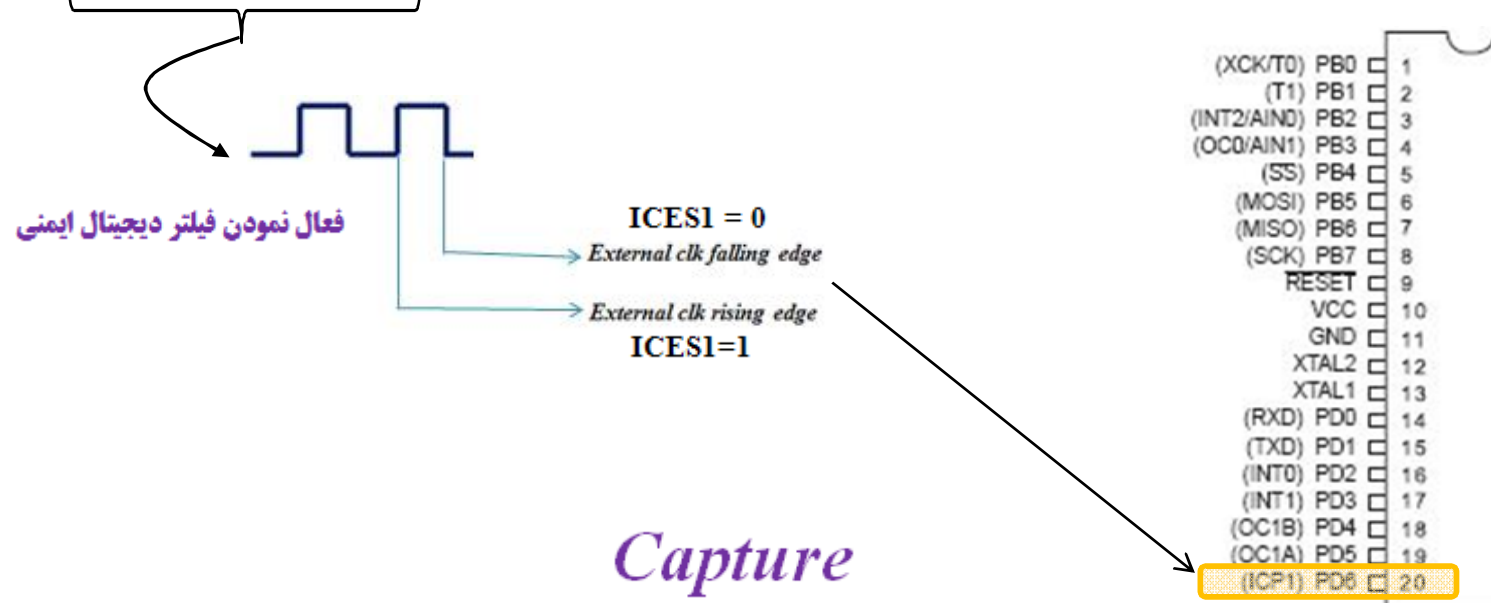
OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
------	------	------	-------	-------	------	------	------

TIMSK:

OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
-------	-------	--------	--------	--------	-------	-------	-------

TCCR1B:

ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
-------	-------	---	-------	-------	------	------	------



TCNT1:

TCNT1F	TCNT1E	TCNT1D	TCNT1C	TCNT1B	TCNT1A	TCNT19	TCNT18	TCNT17	TCNT16	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

ICR1:

ICR1F	ICR1E	ICR1D	ICR1C	ICR1B	ICR1A	ICR19	ICR18	ICR17	ICR16	ICR15	ICR14	ICR13	ICR12	ICR11	ICR10
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

تولید موج سینوسی با استفاده از تایمر یک

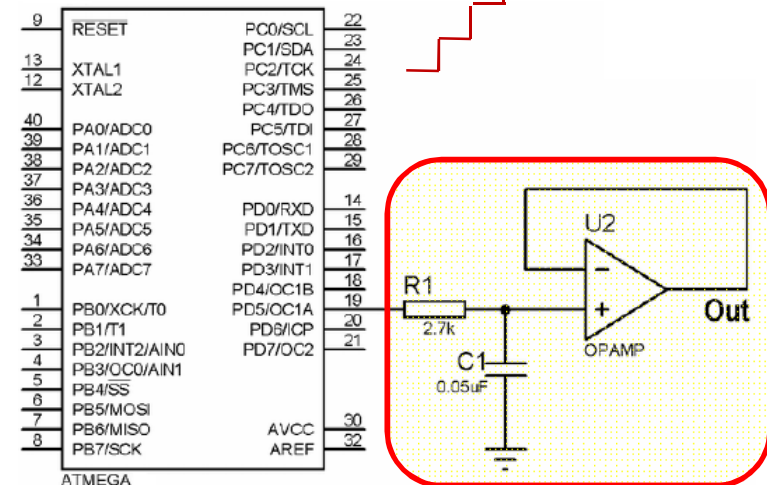
```
#include <mega32.h>

flash char sinewave[256]={
0x80,0x83,0x86,0x89,0x8C,0x8F,0x92,0x95,0x99,0x9C,0x9F,0xA2,0xA5,0xA8,0xAB,0xAE,
0xB1,0xB4,0xB6,0xB9,0xBC,0xBF,0xC1,0xC4,0xC7,0xC9,0xCC,0xCE,0xD1,0xD3,0xD6,0xD8,
0xDA,0xDC,0xDF,0xE1,0xE3,0xE5,0xE6,0xE8,0xEA,0xEC,0xED,0xEF,0xF1,0xF2,0xF3,0xF5,
0xF6,0xF7,0xF8,0xF9,0xFA,0xFB,0xFC,0xFC,0xFD,0xFE,0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,
0xFF,0xFF,0xFF,0xFF,0xFF,0xFE,0xFE,0xFE,0xFE,0xFD,0xFC,0xFC,0xFB,0xFA,0xF9,0xF8,0xF7,
0xF5,0xF4,0xF3,0xF2,0xF0,0xEF,0xED,0xEB,0xEA,0xE8,0xE6,0xE4,0xE2,0xE0,0xDE,0xDC,
0xD9,0xD7,0xD5,0xD2,0xD0,0xCE,0xCB,0xC8,0xC6,0xC3,0xC1,0xBE,0xBB,0xB8,0xB5,0xB3,
0xB0,0xAD,0xAA,0xA7,0xA4,0xA1,0x9E,0x9B,0x97,0x94,0x91,0x8E,0x8B,0x88,0x85,0x82,
0x7E,0x7B,0x78,0x75,0x72,0x6F,0x6C,0x69,0x65,0x62,0x5F,0x5C,0x59,0x56,0x53,0x50,
0x4D,0x4B,0x48,0x45,0x42,0x3F,0x3D,0x3A,0x37,0x35,0x32,0x30,0x2D,0x2B,0x29,0x26,
0x24,0x22,0x20,0x1E,0x1C,0x1A,0x18,0x16,0x14,0x13,0x11,0x0F,0x0E,0x0D,0x0B,0x0A,
0x09,0x08,0x06,0x05,0x05,0x04,0x03,0x02,0x02,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x02,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,
0x0A,0x0B,0x0D,0x0E,0x0F,0x11,0x13,0x14,0x16,0x18,0x1A,0x1C,0x1E,0x20,0x22,0x24,
0x26,0x29,0x2B,0x2D,0x30,0x32,0x35,0x37,0x3A,0x3D,0x3F,0x42,0x45,0x48,0x4B,0x4D,
0x50,0x53,0x56,0x59,0x5C,0x5F,0x62,0x65,0x69,0x6C,0x6F,0x72,0x75,0x78,0x7B,0x7E
};

char i=0;

interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{
    OCR1A=sinewave[i];
    i++;
    if(i==255)
        i=0;
}

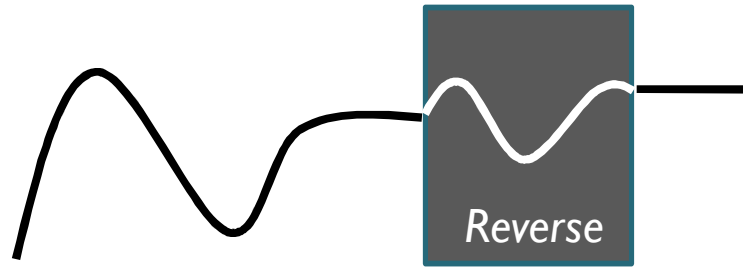
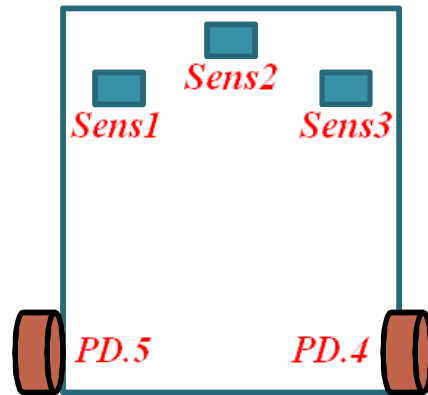
void main(void) {
    DDRD=0xFF;
    TCCR1A=0x81;
    TCCR1B=0x09;
    TIMSK=0x10;
    #asm("sei");
    while (1);
}
```



میکرو پروسیسور

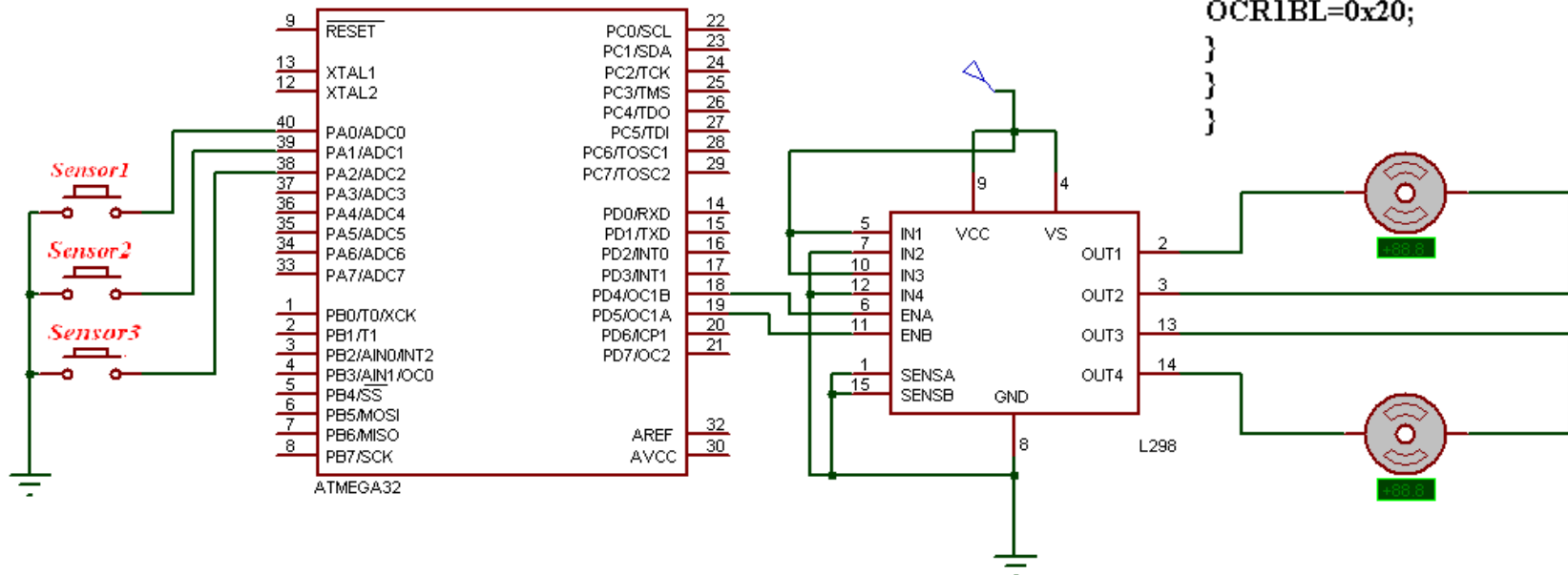
توسعه سیستم
سلام راجی

طراحی کامل ربات تعقیب خط



```
#include<mega32.h>
#include<delay.h>
void main(void)
{
    DDRD=0xff;
    DDRA=0x00;
    TCCR1A=0xa3;
    TCCR1B=0x0d;
    OCR1AH=0x02;
    OCR1AL=0x00;
    OCR1BH=0x02;
    OCR1BL=0x00;
    while(1)
    {
        x=PINA;
        y=PINA.0+PINA.1+PINA.2;
        if ( y>1)
            x = ~x;
    }
}
```

```
if ( x==1)
{
    OCR1AH=0x03;
    OCR1AL=0x20;
    OCR1BH=0x00;
    OCR1BL=0xfa;
}
if ( x==2)
{
    OCR1AH=0x02;
    OCR1AL=0x00;
    OCR1BH=0x02;
    OCR1BL=0x00;
}
if ( x==4)
{
    OCR1AH=0x00;
    OCR1AL=0xfa;
    OCR1BH=0x03;
    OCR1BL=0x20;
}
}
```



میکرو پروسیسور

توسعه سیستم
سلام راجی

فعالسازی وقفه‌های خارجی

فعالسازی وقفه سراسری

غیر فعال نمودن وقفه سراسری

#asm

cli (#asm("cli"))

#endasm

#asm

sei (#asm("sei"))

#endasm

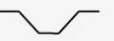
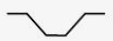
SREG: I T H S V N Z C

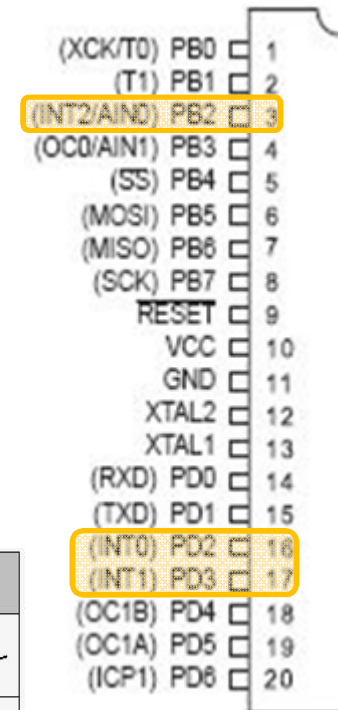
GICR: INT1 INT0 INT2 - - - - -

GICR: INTF1 INTF0 INTF2 - - - - -

MCUCR: - ISC2 - - - - -

MCUCSR: - - - - - ISC11 ISC10 ISC01 ISC00

ISC01	ISC00	نوع تریگر شدن وقفه ی صفر	ISC11	ISC10	نوع تریگر شدن وقفه ی یک
0	0	 سطح منطقی صفر در پین INT0	0	0	 سطح منطقی صفر در پین INT1
0	1	 تغییر در سطح منطقی پین INT0	0	1	 تغییر در سطح منطقی پین INT1
1	0	 لبه ی پایین رونده در پین INT0	1	0	 لبه ی پایین رونده در پین INT1
1	1	 لبه ی بالا رونده در پین INT0	1	1	 لبه ی بالا رونده در پین INT1
ISC2		نوع تریگر شدن وقفه ی دو			
0		لبه ی پائین رونده			
1		لبه ی بالا رونده			



(void) نام روتین سرویس وقفه void [شماره ی بردار وقفه] interrupt

{

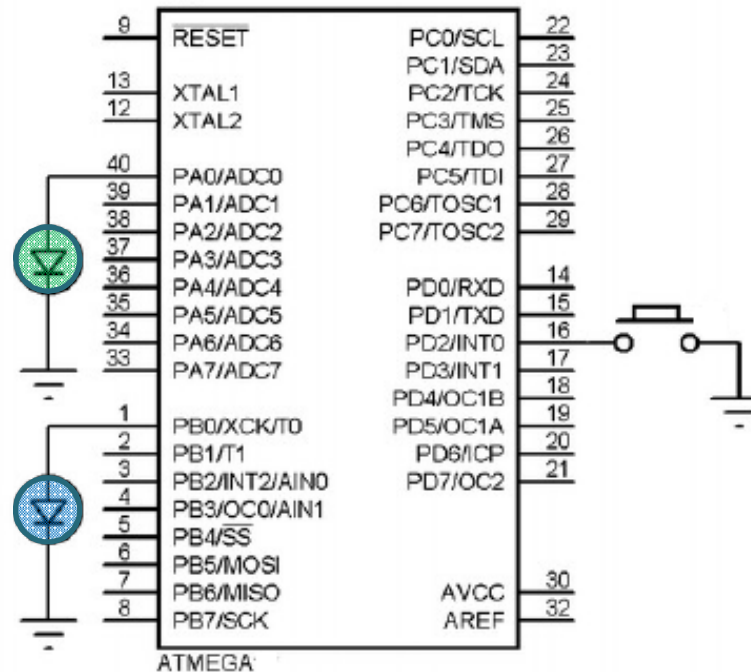
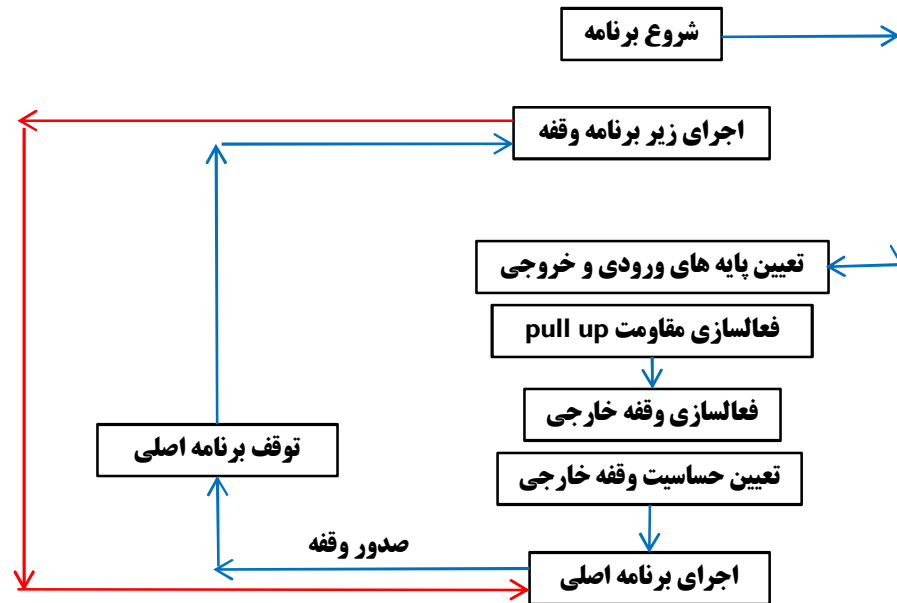
برنامه ی سرویس وقفه

}

میکرو پرو세서

توسعه سیستم
سلطان راجی

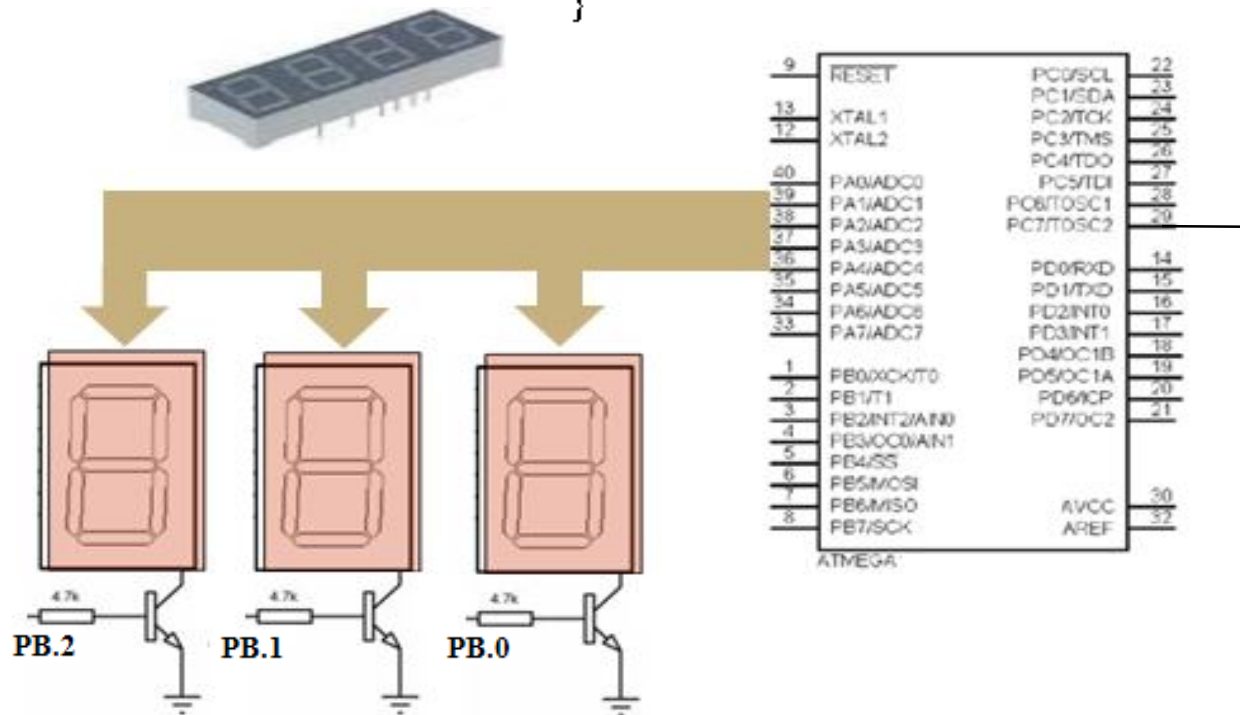
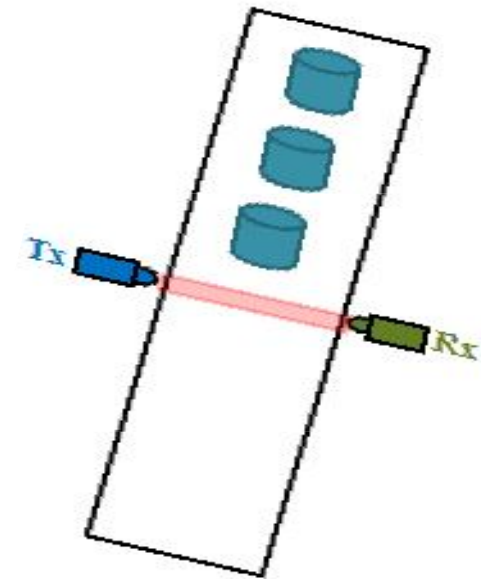

```
#include<mega32.h>
#include<delay.h>
interrupt [2] void Led_on (void)
{
PORTA=0x01;
delay_ms(1000);
PORTA=0x00;
}
void main(void)
{
DDRB=0xff;
PORTB=0x00;
DDRA=0xff;
PORTA=0x00;
DDRD=0x00;
PORTD=0xff;
GICR=0b01000000;
MCUCR=0b00000010;
#asm("sei")
while(1)
{
PORTB=0x01;
delay_ms(500);
PORTB=0x00;
delay_ms(500);
};
}
```



```
#include<mega32.h>
#include<delay.h>
int t = a = b = c = 0;
char digit[10]={0x3f,0x06,0x5b,
0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
interrupt [2] void count(void)
{
t++;
}
void main(void)
{
DDRA=0xff;
DDRB=0xff;
DDRD=0x00;
GICR=0b01000000;
MCUCR=0b00000011;
#asm("sei")
```

```
while(1)
{
if ( t >= 1000)
t = 0;
a = t / 100;
b = (t - (a * 100)) / 10;
c = (t - (a * 100) - (b * 10));
PORTB=0x01;
PORTA=digit[a];
delay_ms(5);
PORTB=0x02;
PORTA=digit[b];
delay_ms(5);
PORTB=0x04;
PORTA=digit[c];
delay_ms(5);
};
}
```

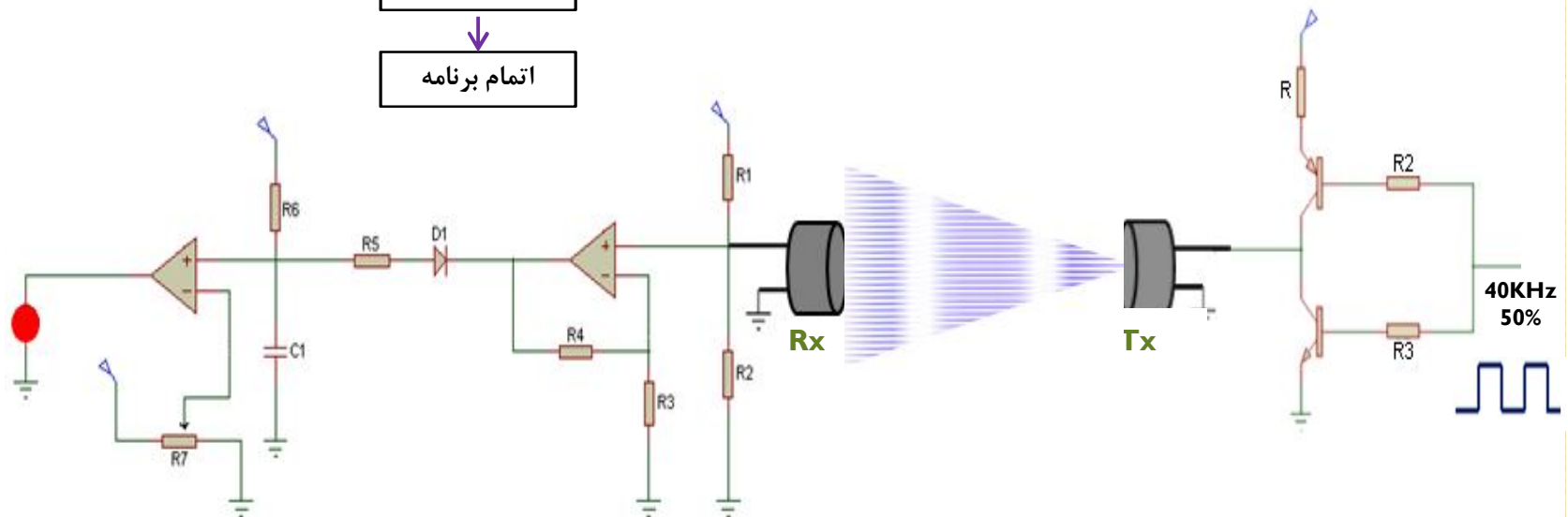
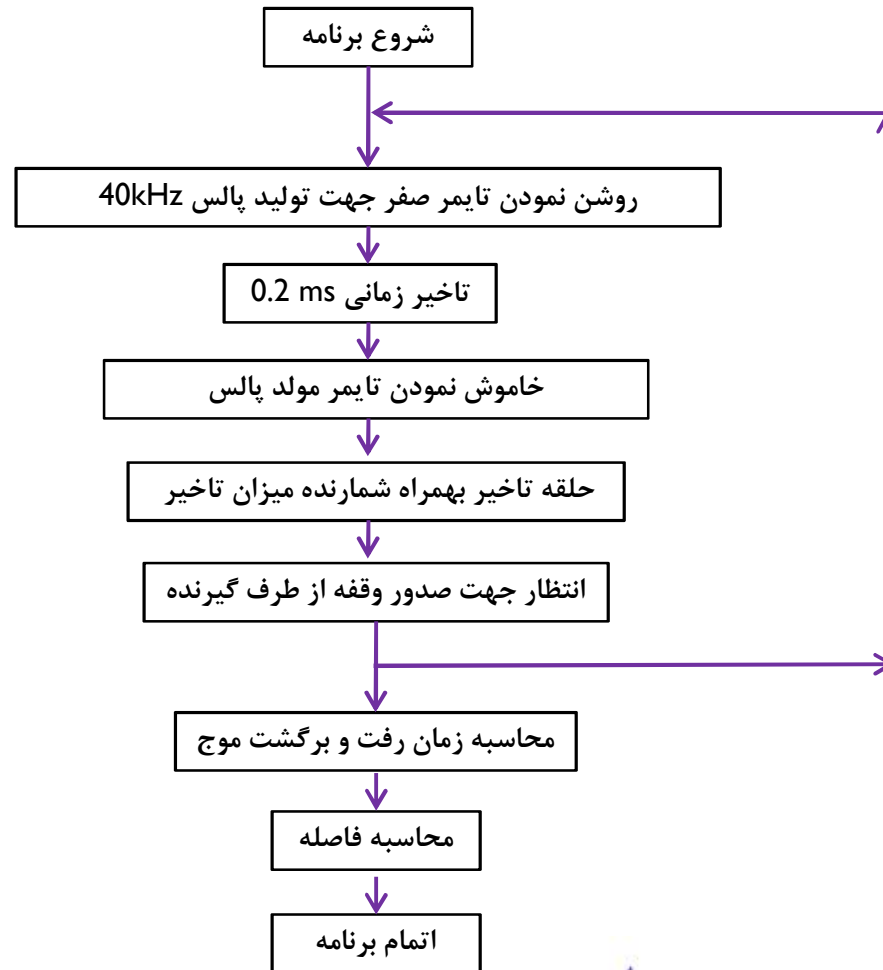
شمارنده نوری خط تولید



راه اندازی سنسورهای اولتراسونیک



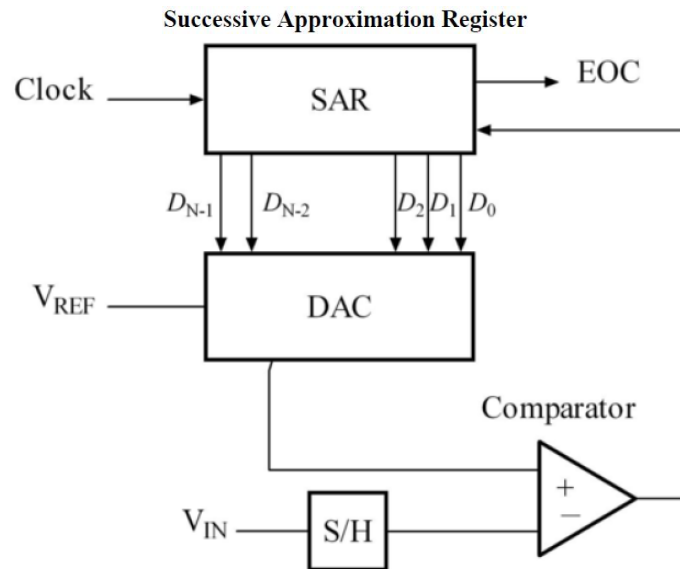
Temperature (C)	Speed of sound (m/sec)
-10	325.5
0	331.5
10	337.5
20	343.5
30	349.5
40	355.5
50	361.5



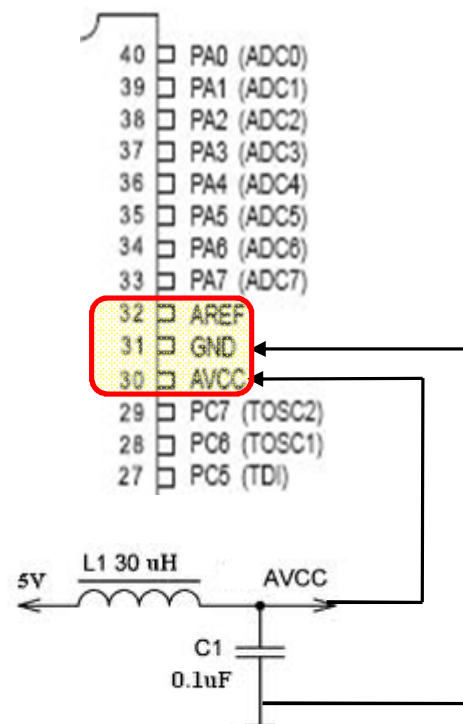
میکرو پرو سسور

توسعه سیستم
سلطان راجی

تبدیل سیگنال آنالوگ به دیجیتال



ابتدا رجیستر SAR با عدد باینری 10000000 بارگذاری می شود و این عدد توسط DAC با مقدار آنالوگ ورودی مقایسه می شود. در صورتی که عدد مقایسه شده بزرگتر باشد و خروجی مقایسه کننده باعث می شود، SAR بیت MSB را پاک کرده و بیت قبل از آن را یک کند و در نتیجه عدد 01000000 در DAC بارگذاری می شود. در صورتی که عدد مقایسه شده کوچکتر باشد خروجی مقایسه کننده باعث می شود بیت کوچکتر نیز یک شده و در نتیجه عدد 11000000 در ورودی DAC بارگذاری شود. این عمل تا پیدا شدن مقدار آنالوگ ادامه داشته و در این زمان بیت End of Conversion به نشانه ی پایان تبدیل یک می شود.



- 10-bit Resolution
- ± 2 LSB Absolute Accuracy
- 65 - 260 μ s Conversion Time
- Up to 15 kSPS at Maximum Resolution
- 8 Multiplexed Single Ended Input Channels
- 7 Differential Input Channels
- 2 Differential Input Channels with Optional Gain of 10x and 200x
- 0 - VCC ADC Input Voltage Range
- Selectable 2.56V ADC Reference Voltage
- Free Running or Single Conversion Mode
- ADC Start Conversion by Auto Triggering on Interrupt Sources
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler

ADMUX:

REFS1

REFS0

ADLAR

MUX4

MUX3

MUX2

MUX1

MUX0

ADLR=0

ADCL	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
ADCH	-	-	-	-	-	-	ADC9	ADC8

ADLR=1

ADCL	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2
ADCH	ADC1	ADC0	-	-	-	-	-	-

REFS1	REFS0	ولتاژ مرجع
۰	۰	ولتاژ پایه ی AREF
۰	۱	ولتاژ پایه ی AVCC
۱	۰	رزرو شده
۱	۱	ولتاژ داخلی ۲.۵۶ ولت

- 40 PA0 (ADC0)
- 39 PA1 (ADC1)
- 38 PA2 (ADC2)
- 37 PA3 (ADC3)
- 36 PA4 (ADC4)
- 35 PA5 (ADC5)
- 34 PA6 (ADC6)
- 33 PA7 (ADC7)
- 32 AREF
- 31 GND
- 30 AVCC

MUX4..0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
00000	ADC0	N/A		
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			
01000	N/A	ADC0	ADC0	10x
01001		ADC1	ADC0	10x
01010 ⁽¹⁾		ADC0	ADC0	200x
01011 ⁽¹⁾		ADC1	ADC0	200x
01100		ADC2	ADC2	10x
01101		ADC3	ADC2	10x
01110 ⁽¹⁾		ADC2	ADC2	200x
01111 ⁽¹⁾		ADC3	ADC2	200x
10000		ADC0	ADC1	1x
10001		ADC1	ADC1	1x
10010		ADC2	ADC1	1x
10011		ADC3	ADC1	1x
10100		ADC4	ADC1	1x
10101		ADC5	ADC1	1x
10110		ADC6	ADC1	1x
10111		ADC7	ADC1	1x
11000		ADC0	ADC2	1x
11001		ADC1	ADC2	1x
11010		ADC2	ADC2	1x
11011		ADC3	ADC2	1x
11100		ADC4	ADC2	1x
MUX4..0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
11101		ADC5	ADC2	1x
11110	1.22 V (V _{BO})	N/A		
11111	0 V (GND)			

میکرو پروسیسور

توسعه سیستم
سلطان راجی

ADCSRA:

ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
------	------	-------	------	------	-------	-------	-------

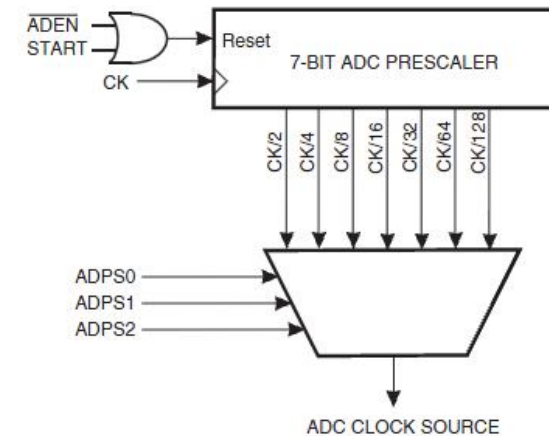
1

SFIOR:

ADTS2	ADTS1	ADTS0	-	-	-	-	-
-------	-------	-------	---	---	---	---	---

ADPS2	ADPS1	ADPS0	ضریب تقسیم
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

ADTS2	ADTS1	ADTS0	منبع راه اندازی
0	0	0	Free Running
0	0	1	مقایسه کننده ی آنالوگ
0	1	0	وقته ی خارجی صفر
0	1	1	تطبیق مقایسه ی تایمر صفر
1	0	0	سرریز تایمر صفر
1	0	1	تطبیق مقایسه ی B تایمر یک
1	1	0	سرریز تایمر یک
1	1	1	Capture تایمر یک



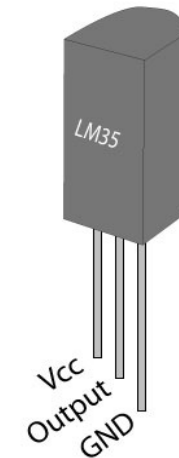
By default, the successive approximation circuitry requires an input clock frequency between 50kHz and 200kHz to get maximum resolution. If a lower resolution than 10 bits is needed, the input clock frequency to the ADC can be higher than 200kHz to get a higher sample rate.

اندازه گیری و نمایش دما

```
#include<mega32.h>
#include<delay.h>
#include<stdio.h>
#include<lcd.h>
#define xtal 8000000
#asm
.equ _lcd_port=0x12;
#endasm
char lcd_buff[10];
int adc_in;
float temp;
unsigned int read_adc(void)
{
    ADMUX=0xc0;
    ADCSRA |=0x40;
    while((ADCSRA & 0x10)= = 0);
    ADCSRA|=10;
    return ADCW;
}
```

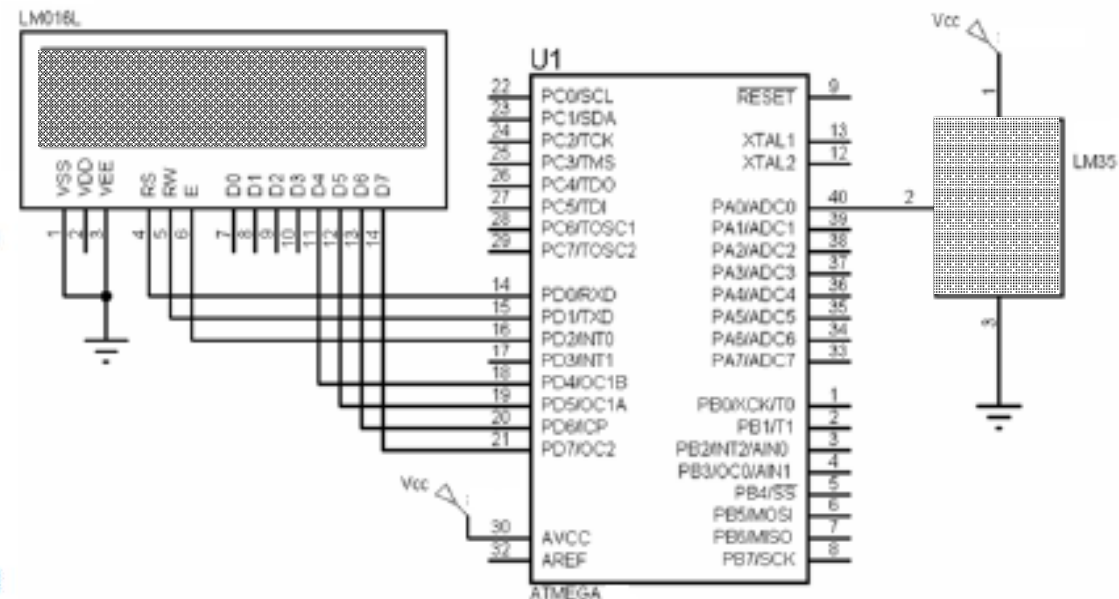
```
void main(void)
{
    PORTA=0x00;
    DDRA=0x00;
    ADMUX=0xc0;
    ADCSRA=0x86;
    lcd_init(16);
    while(1)
    {
        adc_in=read_adc()
        temp=adc_in/4;
        sprintf(lcd_buff,"Temp=%f",temp);
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_puts(lcd_buff);
        delay_ms(1000);
    }
}
```

temp=adc_in/4;



Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full -55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±¼°C typical
- Low impedance output, 0.1 Ω for 1 mA load

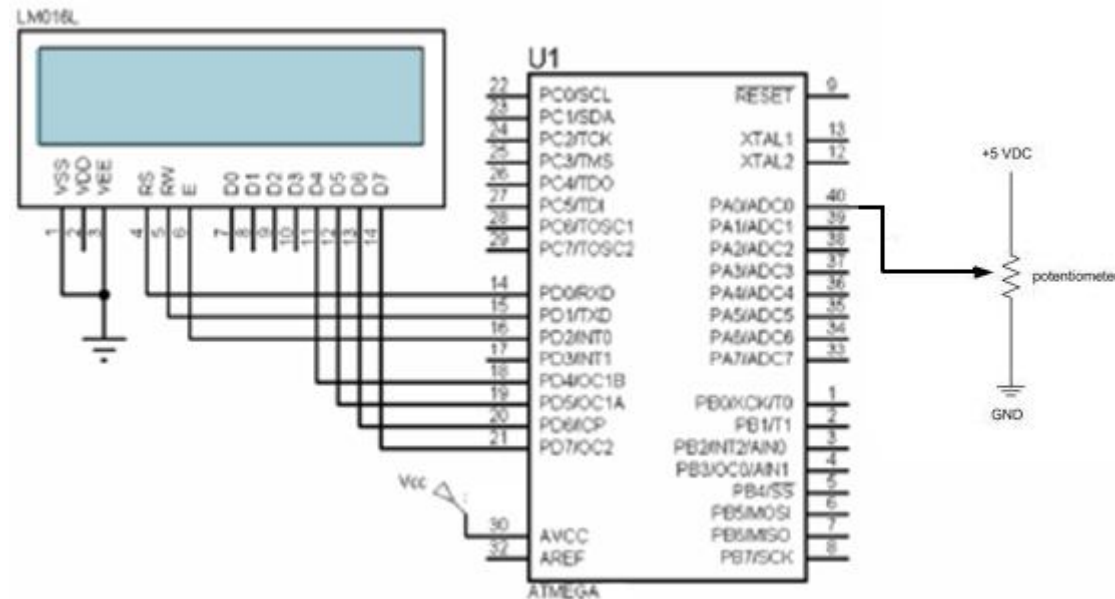
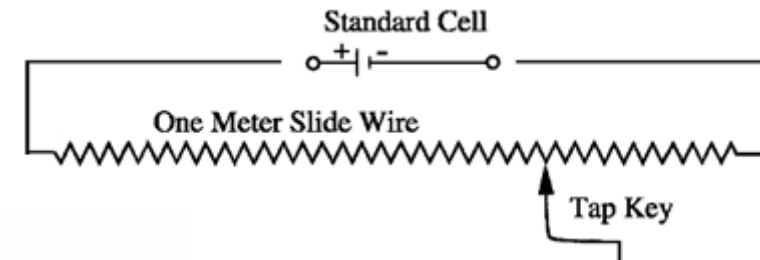


اندازه گیری الکترونیکی و نمایش فاصله



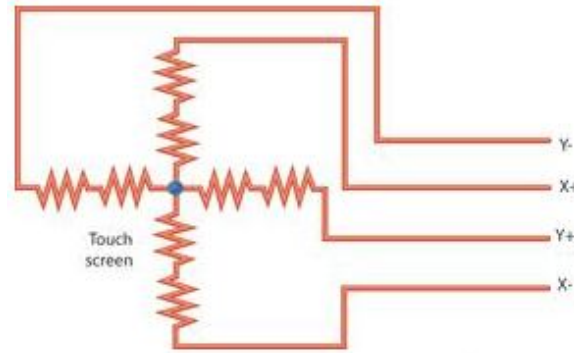
```
#include<mega32.h>
#include<delay.h>
#include<stdio.h>
#include<lcd.h>
#define xtal 8000000
#asm
.equ _lcd_port=0x12;
#endasm
char lcd_buff[10];
int adc_in;
float temp;
unsigned int read_adc(void)
{
    ADMUX=0x40;
    ADCSRA |=0x40;
    while((ADCSRA & 0x10)= = 0);
    ADCSRA|=10;
    return ADCW;
}
```

```
void main(void)
{
    PORTA=0x00;
    DDRA=0x00;
    ADMUX=0x40;
    ADCSRA=0x86;
    lcd_init(16);
    while(1)
    {
        adc_in=read_adc()
        temp=adc_in/10; // L=10cm, Vab=+5v, Sens. =5/1024=5mv, Measur. Sens. =1mm
        sprintf(lcd_buff,"Temp=%f",temp);
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_puts(lcd_buff);
        delay_ms(1000);
    }
}
```



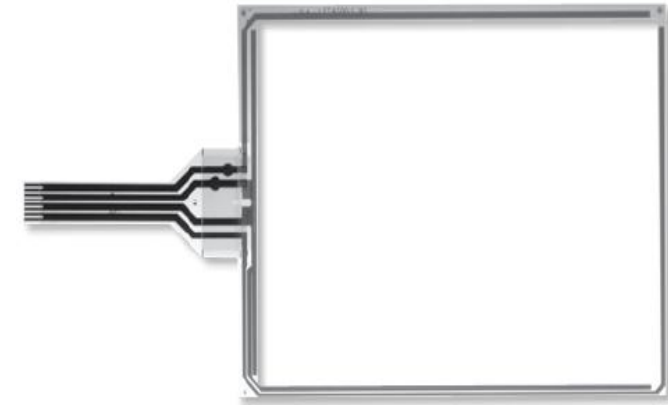
میکرو پرو سسور

تیسو تنظیم
سلطان راجہ



4-Wire Analog Resistive Touch Screens

Optical	
Light Transmission	Analog: 80% standard Digital: 78% standard
Film Options	Anti-glare, anti-Newton ring standard
Electrical	
Power Level	1 mA @ 5V DC (resistive load)
Insulation Impedance	10MΩ minimum @ 25V DC
Linearity	3% maximum (analog)
Chattering Time	10 milliseconds maximum
Mechanical	
Touch Activation Force	1.4N maximum
Available Sizes	5.7" ~ 15" standard
Durability	
Surface Hardness	2H (JIS K5600)
Expected Operational Life	1,000,000 operations minimum
Environmental	
Operating Temperature Range	-10°C ~ +60°C (+14°F ~ +140°F)
Storage Temperature Range	-20°C ~ +70°C (-4°F ~ +158°F)
Relative Humidity	+60°C (+140°F), humidity 90%, 240 hours

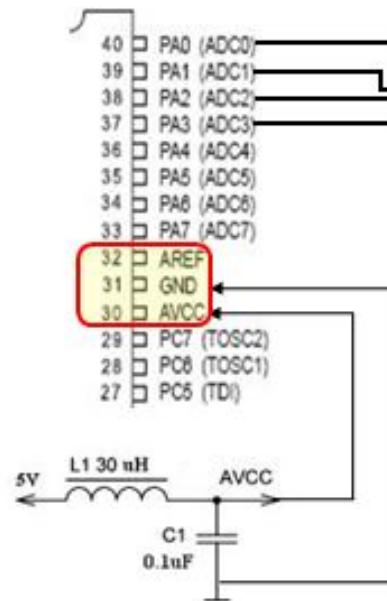


Touch Screen

Vcc
X
Gnd
Hiz

Touch Screen

Hiz
Vcc
Y
Gnd



```
#include<mega32.h>
#include<delay.h>
#define xtal 8000000
int adc_in;
unsigned int read_adc(void)
{
    ADCSRA |= 0x40;
    while((ADCSRA & 0x10) == 0);
    ADCSRA |= 10;
    return ADCW;
}
```

```
void main(void)
{
    while(1)
    {
        DDRA=0b00000101;
        PORTA=0b00000001;
        ADMUX=0x41;
        ADCSRA=0x86;
        x=read_adc();
        delay_ms(10);
        DDRA=0b00001010;
        PORTA=0b00000010;
        ADMUX=0x42;
        ADCSRA=0x86;
        y=read_adc();
        if (0b0000000100<x<0b1110011011 && 0b0000000101<y<0b0000110110)
        {
            Start Program
        }
        if (0b0000000100<x<0b1110011011 && 0b0110110001<y<0b1101100010)
        {
            Stop Program
        }
    }
}
```

Calibration

START

(0000000100,0000000101)

STOP

(1110011011,0000110110)

(0000000100,0110110001)

(1110011011,1101100010)

ارسال اطلاعات بصورت سریال

UDR(Trans.):	UDR7		UDR6		UDR5		UDR4		UDR3		UDR2		UDR1		UDR0	
UDR(Rec.):	UDR7		UDR6		UDR5		UDR4		UDR3		UDR2		UDR1		UDR0	
UCSRA:	RXC		TXC		UDRE		FE		DOR		PE		U2X		MPCM	
UCSRB:	RXCIE		TXCIE		UDRIE		RXEN		TXEN		UCSZ2		RXB8		TXB8	
UCSRA:	URSEL		UMSEL		UPM1		UPM0		USBS		UCSZ1		UCSZ0		UCPOL	
UBRR:	U15	U14	U13	U12	U11	U10	U9	U8	U7	U6	U5	U4	U3	U2	U1	U0

اندازه ی کاراکتر	UCSZ0	UCSZ1	UCSZ2
۵ بیت	۰	۰	۰
۶ بیت	۱	۰	۰
۷ بیت	۰	۱	۰
۸ بیت	۱	۱	۰
رژرو شده	۰	۰	۱
رژرو شده	۱	۰	۱
رژرو شده	۰	۱	۱
حالت ۹ بیتی	۱	۱	۱



UCPOL	تغییر داده ی ارسالی روی پین TxD	نمونه برداری از داده ی دریافتی روی پین RxD
۰	لبه ی بالا رونده ی پالس XCK	لبه ی پایین رونده ی پالس XCK
۱	لبه ی پایین رونده ی پالس XCK	لبه ی بالا رونده ی پالس XCK

وضعیت بیت توازن	UPM0	UPM1	محاسبه UBRR از روی Baud Rate	محاسبه Baud Rate از روی UBRR	Mode کاری
غیر فعال	۰	۰	$UBRR = \frac{f_{osc}}{16 \times Baud} - 1$	$Baud = \frac{f_{osc}}{16(UBRR + 1)}$	آسنکرون (U2X=0)
رژرو شده	۱	۰	$UBRR = \frac{f_{osc}}{8 \times Baud} - 1$	$Baud = \frac{f_{osc}}{8(UBRR + 1)}$	آسنکرون با سرعت دوبرابر (U2X=1)
توازن زوج	۰	۱	$UBRR = \frac{f_{osc}}{16 \times Baud} - 1$	$Baud = \frac{f_{osc}}{16(UBRR + 1)}$	عملکرد سنکرون
توازن فرد	۱	۱	$UBRR = \frac{f_{osc}}{16 \times Baud} - 1$	$Baud = \frac{f_{osc}}{16(UBRR + 1)}$	

(XCK/T0) PB0	1
(T1) PB1	2
(INT2/AIN0) PB2	3
(OC0/AIN1) PB3	4
(SS) PB4	5
(MOSI) PB5	6
(MISO) PB6	7
(SCK) PB7	8
RESET	9
VCC	10
GND	11
XTAL2	12
XTAL1	13
(RXD) PD0	14
(TXD) PD1	15
(INT0) PD2	16
(INT1) PD3	17
(OC1B) PD4	18
(OC1A) PD5	19
(ICP1) PD6	20

میکرو پرو세서

توسعه سیستم
سلام راجی

Usart Functions

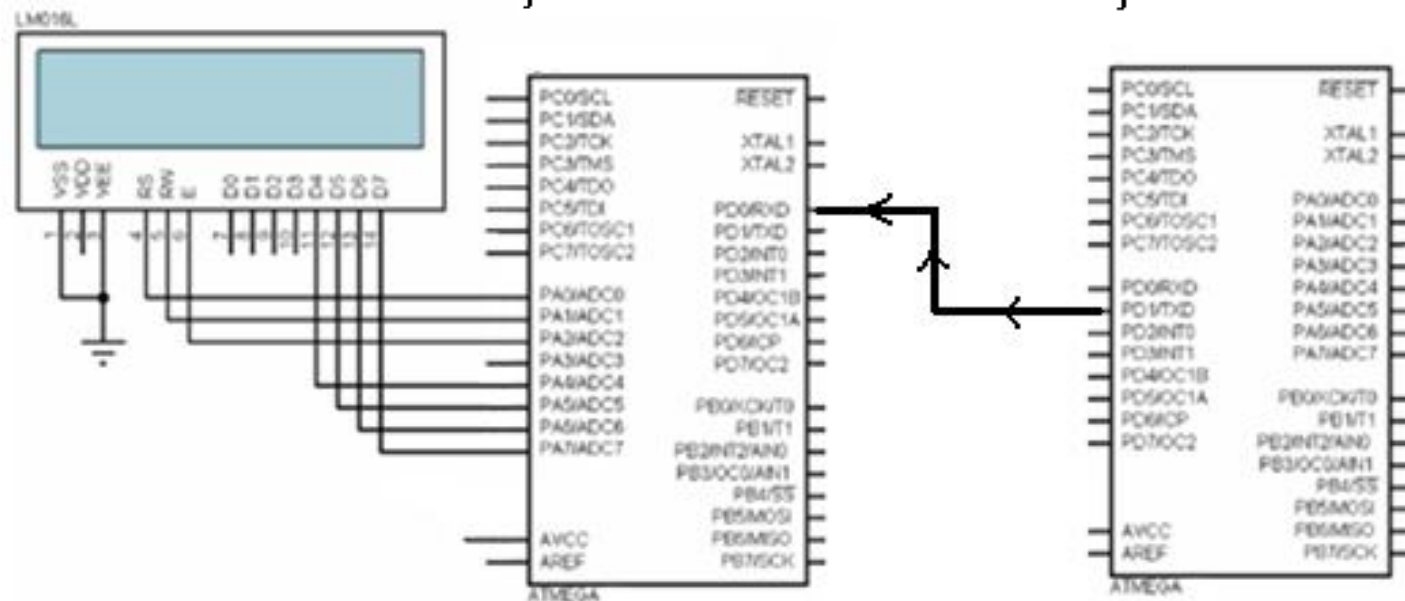
putchar(--)	puts(--)	printf(" ",--)
-- =getchar()	putsf(--)	scanf(" ",--)
	gets(-- , size)	

```
#include<mega32.h>
#include<delay.h>
#include<stdio.h>
#include<lcd.h>
#define xtal 1000000
#asm
.equ _lcd_port=0x1b
#endasm
char buff[5];
char a;
DDRA=0xff;
```

```
void main(void)
{
UCSRA=0x00;
UCSRB=0x10;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x33;
a=getchar();
sprintf(buff,"%c",a);
lcd_puts(buff);
while(1);
}
```

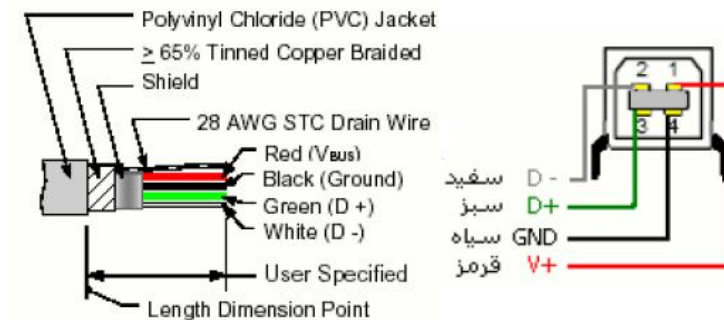
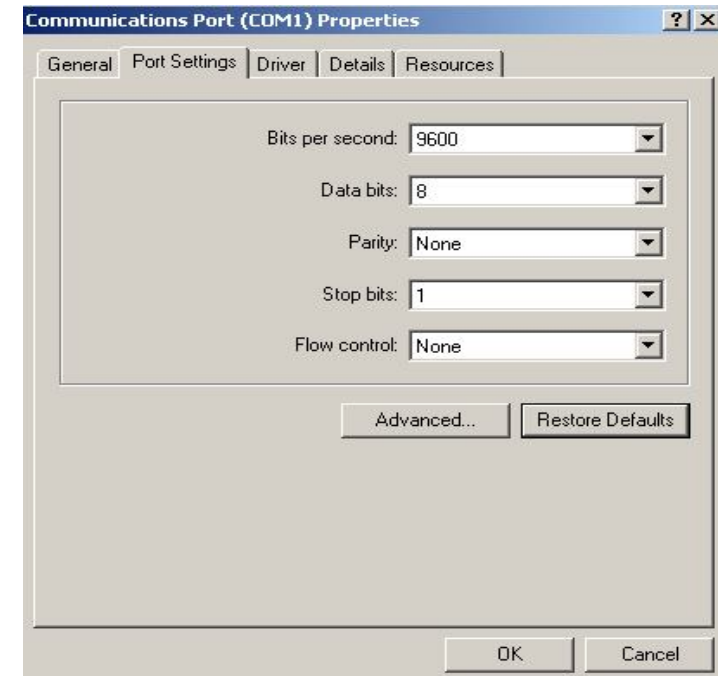
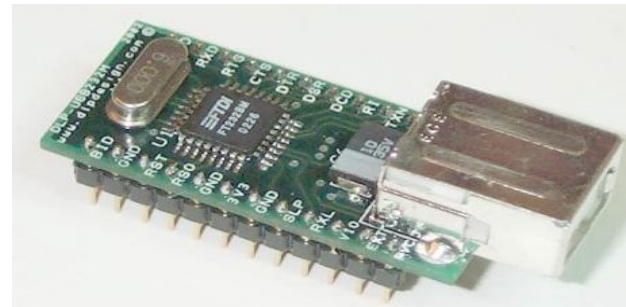
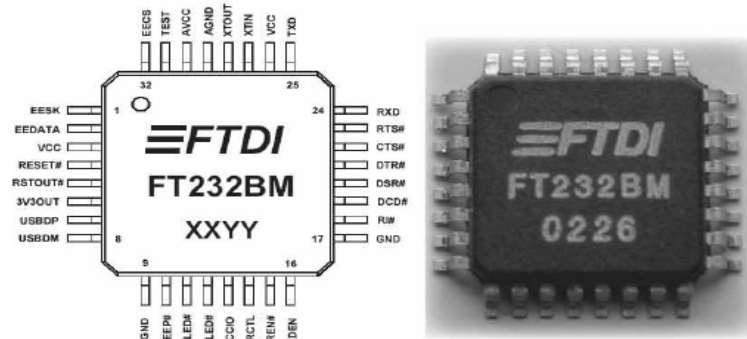
```
#include<mega32.h>
#include<delay.h>
#include<stdio.h>
#define xtal 1000000
```

```
void main(void)
{
UCSRA=0x00;
UCSRB=0x08;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x33;
delay_ms(1000);
putchar(1);
while(1);
}
```



ارتباط سریال میکروکنترلر و کامپیوتر

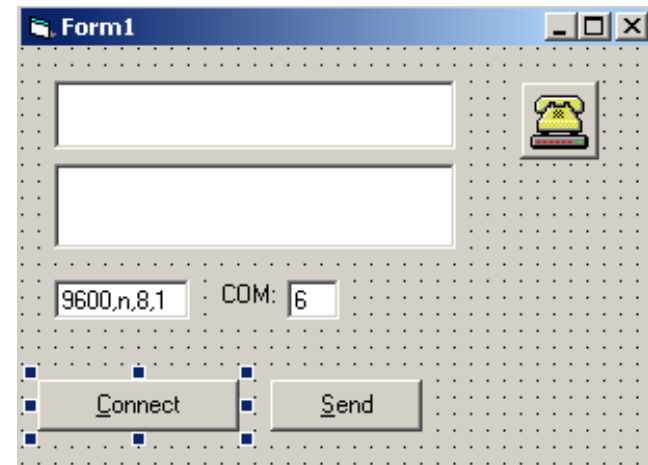
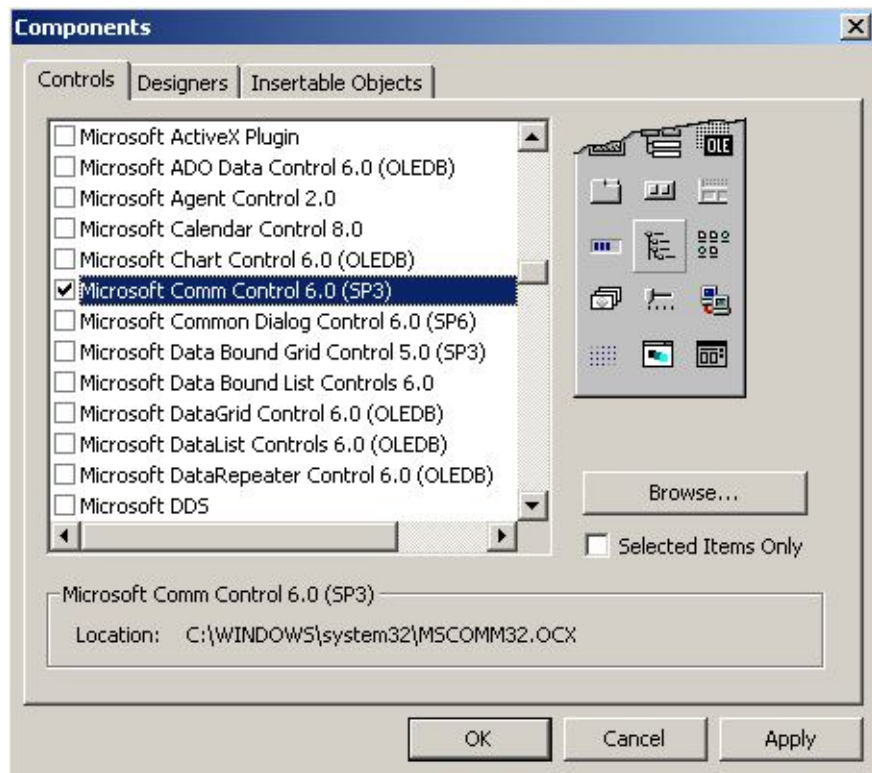
<http://www.ftdichip.com/Drivers/VCP.htm>



میکرو پرو سسور

توسعه سیستم
سلطان راجبی

ارتباط سریال میکروکنترلر و کامپیوتر



```
Private Sub Comm_OnComm()  
    Text2.Text = Text2.Text + Comm.Input
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
    If Command1.Caption = "&Connect" Then  
        Comm.CommPort = comnum.Text  
        Comm.Settings = comset.Text  
        Comm.PortOpen = True  
        Command1.Caption = "&Disconnect"
```

```
    Else
```

```
        Comm.PortOpen = False  
        Command1.Caption = "&Connect"
```

```
    End If
```

```
End Sub
```

```
Private Sub Command2_Click()
```

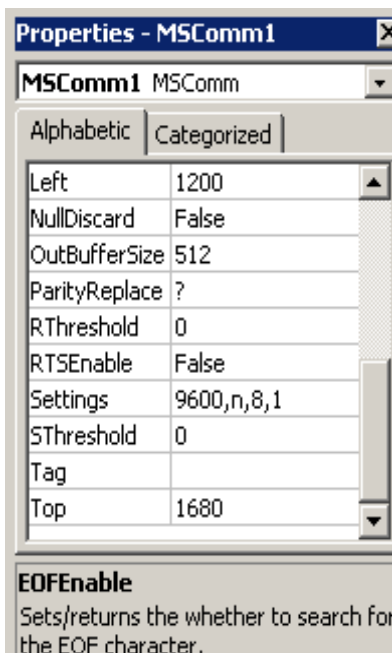
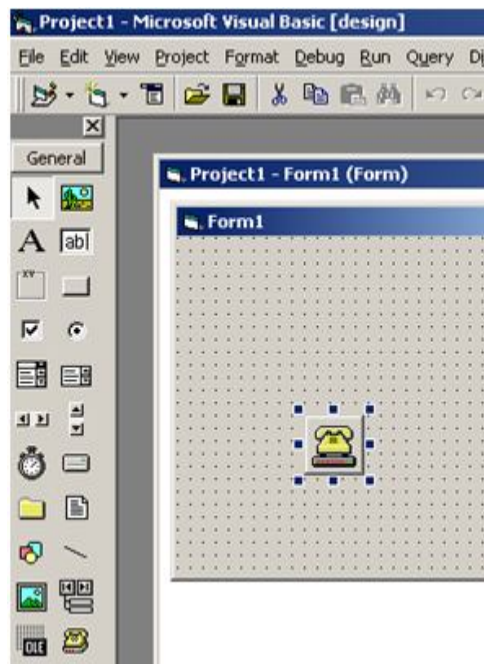
```
    If Comm.PortOpen = True Then  
        Comm.Output = Text1.Text
```

```
    Else
```

```
        MsgBox "frst open the port!"
```

```
    End If
```

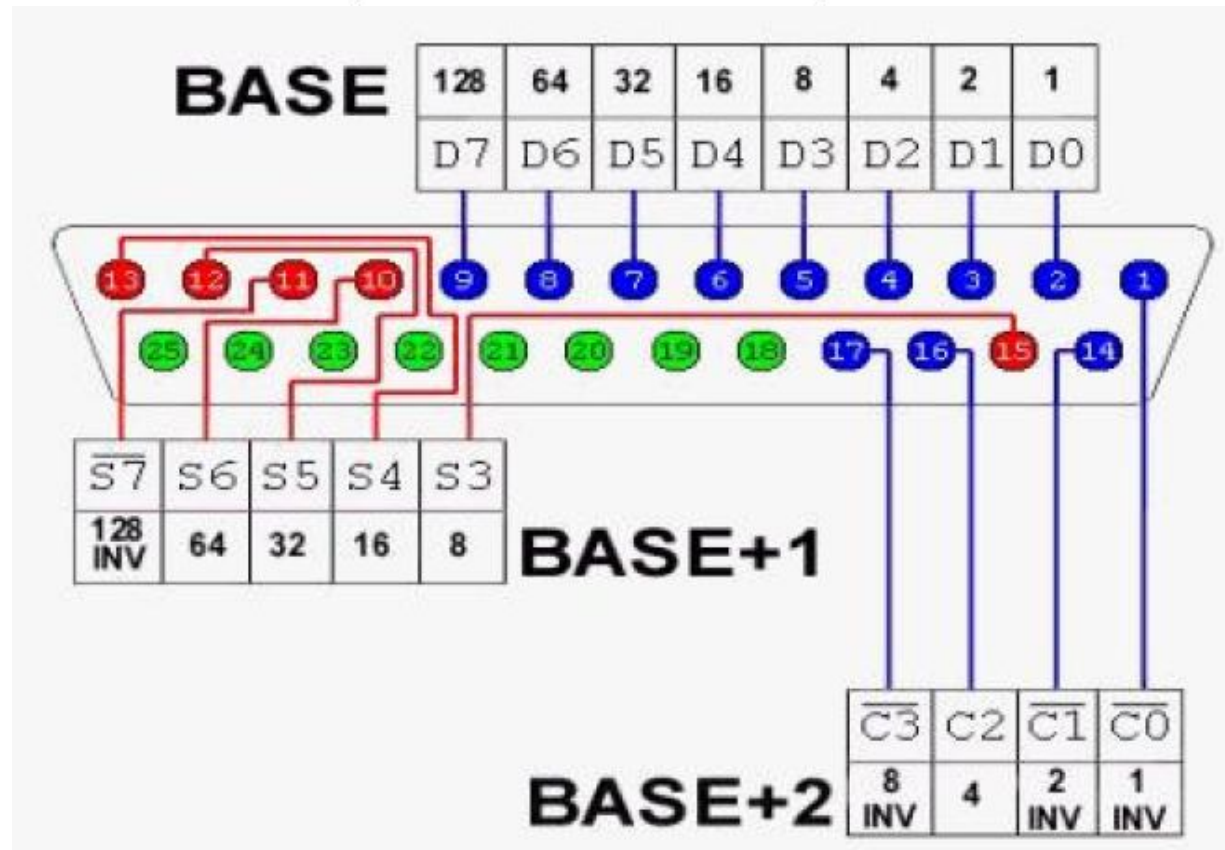
```
End Sub
```



میکرو پروسیسور

تیسو تنظیم
سلطان راجہ

BASE=0x378 , BASE+1=0x379 , BASE+2=0x37a



Outp(0x378,0xaa);
Outport(0x37a,175);

A=inp(0x37a);
B=inport(0x379);



PARMON.EXE

Parallel Port Monitor				
Port	Dec	Hex	Binary	Beep
378	255	FF	11111111	No
379	127	7F	01111111	No
37A	204	CC	11001100	No

اتصال ماژولهای فرستنده و گیرنده به میکروکنترلر



Transmitter

Vcc Gnd Data

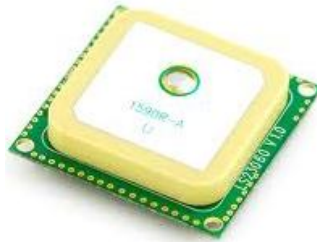
Receiver

Vcc Gnd Data Enable

```
$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6A
```

Where:

RMC	Recommended Minimum sentence C
123519	Fix taken at 12:35:19 UTC
A	Status A=active or V=Void.
4807.038,N	Latitude 48 deg 07.038' N
01131.000,E	Longitude 11 deg 31.000' E
022.4	Speed over the ground in knots
084.4	Track angle in degrees True
230394	Date - 23rd of March 1994



1-GND

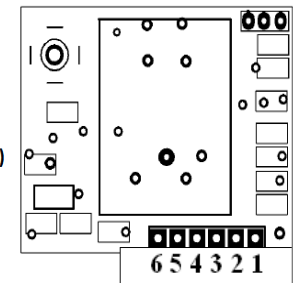
2-VCC

3-RXD (INPUT)

4-TXD (OUTPUT)

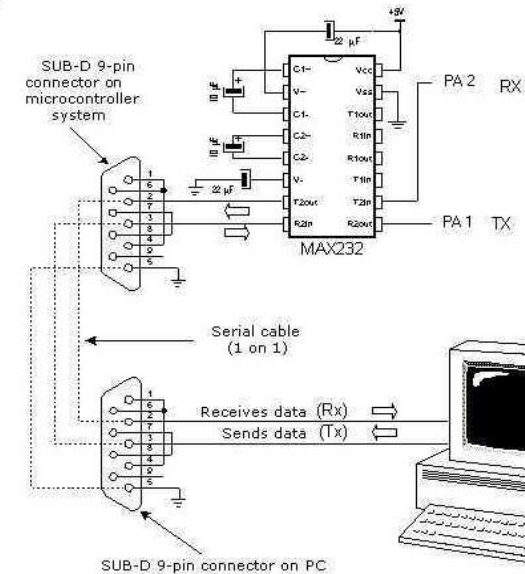
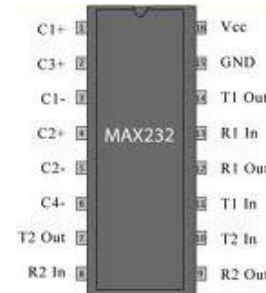
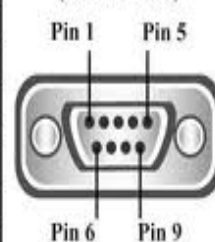
5-RESET

6-GND



Pin 1	N/C
Pin 2	RXD
Pin 3	TXD
Pin 4	DTR
Pin 5	GND
Pin 6	DSR
Pin 7	RTS
Pin 8	CTS
Pin 9	Power Input

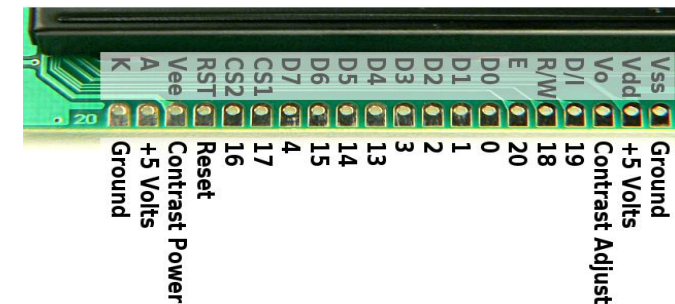
RS232 Pinout
(9 Pin Male)



میکرو پرو سسور

توسعه سیستم
سلام راجی

تہیہ و تنظیم
سلمان راہی



```
for (i=0;i<127;i++){
do{
i=0;
glcd_setpixel(i,p);
delay_us(100);
i++;
if(i==128)
i=0;
}
}
}
```

Pin No.	Symbol	Level	Description
1	VDD	5.0V	Supply voltage for logic
2	Vss	0V	Ground
3	V _a	(Variable)	Operating voltage for LCD
4	DB0	H/L	Data bit 0
5	DB1	H/L	Data bit 1
6	DB2	H/L	Data bit 2
7	DB3	H/L	Data bit 3
8	DB4	H/L	Data bit 4
9	DB5	H/L	Data bit 5
10	DB6	H/L	Data bit 6
11	DB7	H/L	Data bit 7
12	CS1	L	Select Column 1~ Column 64
13	CS2	L	Select Column 65~ Column 128
14	RST	L	Reset signal
15	R/W	H/L	H: Read (MPU- Module) , L: Write (MPU - Module)
16	D/I	H/L	H: Data , L : Instruction
17	E	H	Enable signal
18	Vee	—	Negative Voltage output
19	A	—	Power Supply for LED backlight (+)
20	K	—	Power Supply for LED backlight (-)





```
#include <mega32.h>
#include <delay.h>
#define ADC_VREF_TYPE 0xC0
#include <stdlib.h>
#include <stdio.h>
#define PORTRAIT
#define xtal 8000000
#define LCD_RST_DDR DDRB
#define LCD_RST_PORT PORTB
#define LCD_RST_PIN 0
```

```
#define LCD_RS_DDR DDRB
#define LCD_RS_PORT PORTB
#define LCD_RS_PIN 1
```

```
#define LCD_CS_DDR DDRB
#define LCD_CS_PORT PORTB
#define LCD_CS_PIN 2
```

```
#define LCD_RD_DDR DDRB
#define LCD_RD_PORT PORTB
#define LCD_RD_PIN 3
```

```
#define LCD_WR_DDR DDRB
#define LCD_WR_PORT PORTB
#define LCD_WR_PIN 4
```

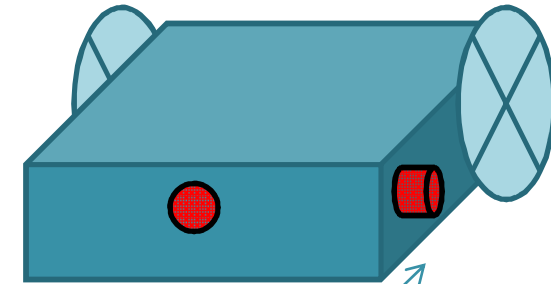
```
#define LCD_DATAPORT_MSB_DDR DDRD
#define LCD_DATAPORT_MSB_PORT PORTD
#define LCD_DATAPORT_MSB_PIN PIND
```

```
#define LCD_DATAPORT_LSB_DDR DDRC
#define LCD_DATAPORT_LSB_PORT PORTC
#define LCD_DATAPORT_LSB_PIN PINC
```

```
#define LCD_CONTROLPORT_DDR DDRB
#define LCD_CONTROLPORT_PORT PORTB
#define LCD_CONTROLPORT_PIN PINB
#include <tfilcd_functions.h>
```

```
void main(void)
{
    lcd_init();
    lcd_background_color(YELLOW);
    border();
    lcd_gotoxy(8,3);
    lcd_putsf("*** H E L L O ***",BLACK,0,PURPLE);
    lcd_gotoxy(8,5);
    lcd_putsf(" W E L C O M ",BLACK,0,PURPLE);
    lcd_gotoxy(3,7);
    lcd_putsf(" This is my project ",BLACK,0,GREEN);
    lcd_gotoxy(3,14);
    lcd_putsf(" $$ In the Name of GOD $$",BLACK,0,GREEN);
    delay_ms(5000);
    lcd_clear_screen();
    lcd_background_color(YELLOW);
    border();
    lcd_gotoxy(1,2);
    lcd_putsf(" Select your favorite color ",BLACK,0,PURPLE);
    lcd_gotoxy(9,3);
    lcd_putsf("for drawing ",BLACK,0,PURPLE);
    delay_ms(1000);
    lcd_draw_circle(50,110,30,1,BLUE);
    lcd_draw_circle(160,110,30,1,BLACK);
    lcd_draw_circle(50,220,30,1,GREEN);
    lcd_draw_circle(160,220,30,1,RED);
}
```


رباتهای لابرینت



*Motion
Priority*

Right

Left

Back

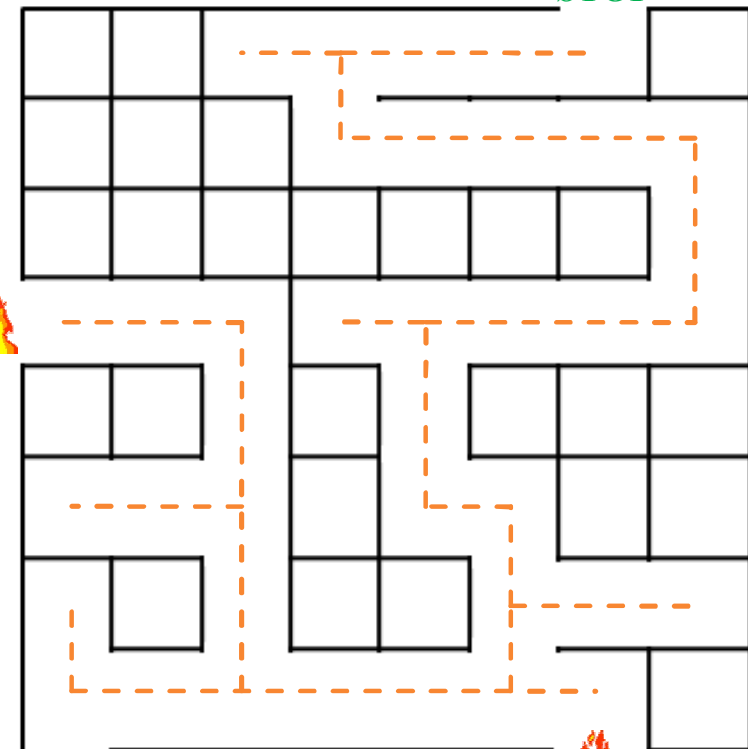
W

N

E

S

STOP



START

```
void right (void)
{
j=1;
PORTB=0;
for (i=1 ; i<=n ; i++)
{
PORTA=j;
delay_ms(50);
j = j*2;
if (j == 16)
j = 1;
}
}
```

```
void left (void)
{
j=1;
PORTA=0;
for (i=1 ; i<=n ; i++)
{
PORTB=j;
delay_ms(50);
j = j*2;
if (j == 16)
j = 1;
}
}
```

```
void direct (void)
{
j=1;
for (i=1 ; i<=n ; i++)
{
PORTB=j;
PORTA=j;
delay_ms(50);
j = j*2;
if (j == 16)
j = 1;
}
}
```

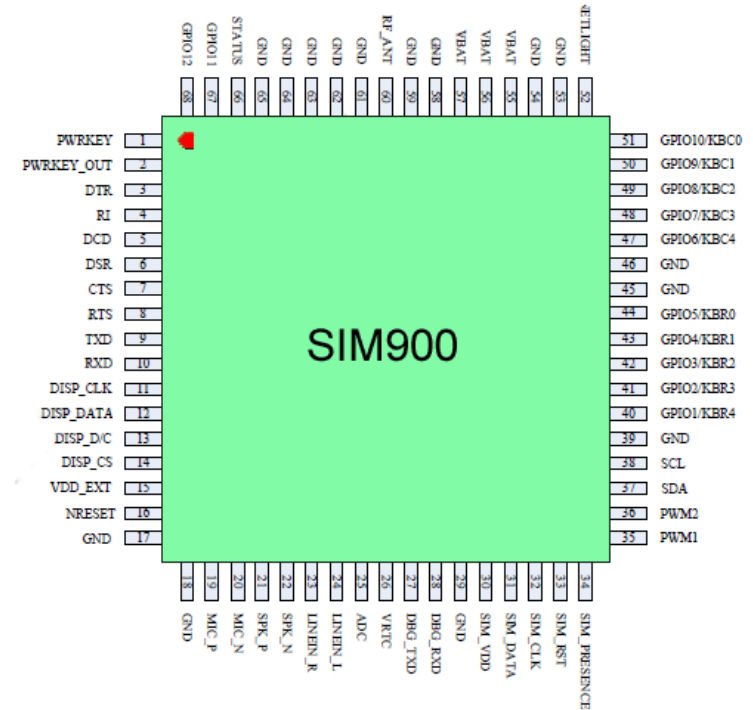
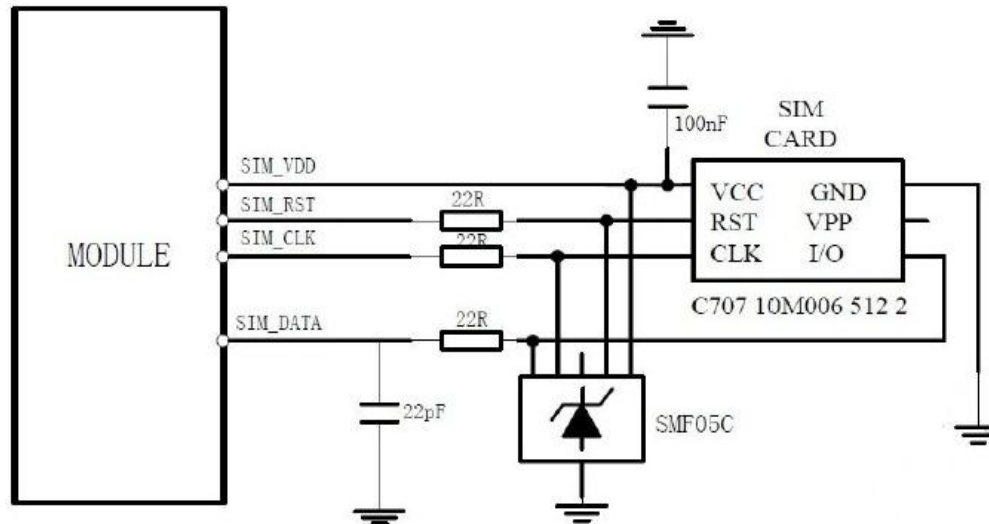
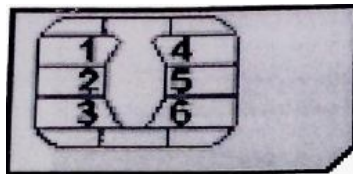
میکرو پروسیسور

توسعه سیستم
سلطان راجی

میکرو پرو سسور



5- Vpp: برای برنامه ریزی سیم کارت (دلخواه)
6- Data: پایه ی دیتای سیم کارت



State	SIM900 function
Off	SIM900 is not running
64ms On/ 800ms Off	SIM900 does not find the network
64ms On/ 3000ms Off	SIM900 find the network
64ms On/ 300ms Off	GPRS communication





میکرو پرو سسور

تهیه و تنظیم:
لمان راجی