

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

آزمایشگاه میکروکنترلر ۸۰۵۱

(دانشکده فنی تهران جنوب)

آزمایش های شماره ۱ و ۳

گردآورنده :

مهدی کمان گری

www.kamangari.blogfa.com

www.ir-micro.com



ویژگی های میکروکنترلر MCS-51:

میکروکنترلر خانواده ی ۸۰۵۱ دارای چندین مدل می باشد که نوع ۴۰ پایه ی آن مورد نظر است .
شماره میکروکنترلر های پر کاربرد این خانواده عبارتند از :
AT89C4051 ، AT89C2051 ، AT89C1051 ، AT89C55 ، AT89C52 ، AT89C51

مشخصات AT89C51:

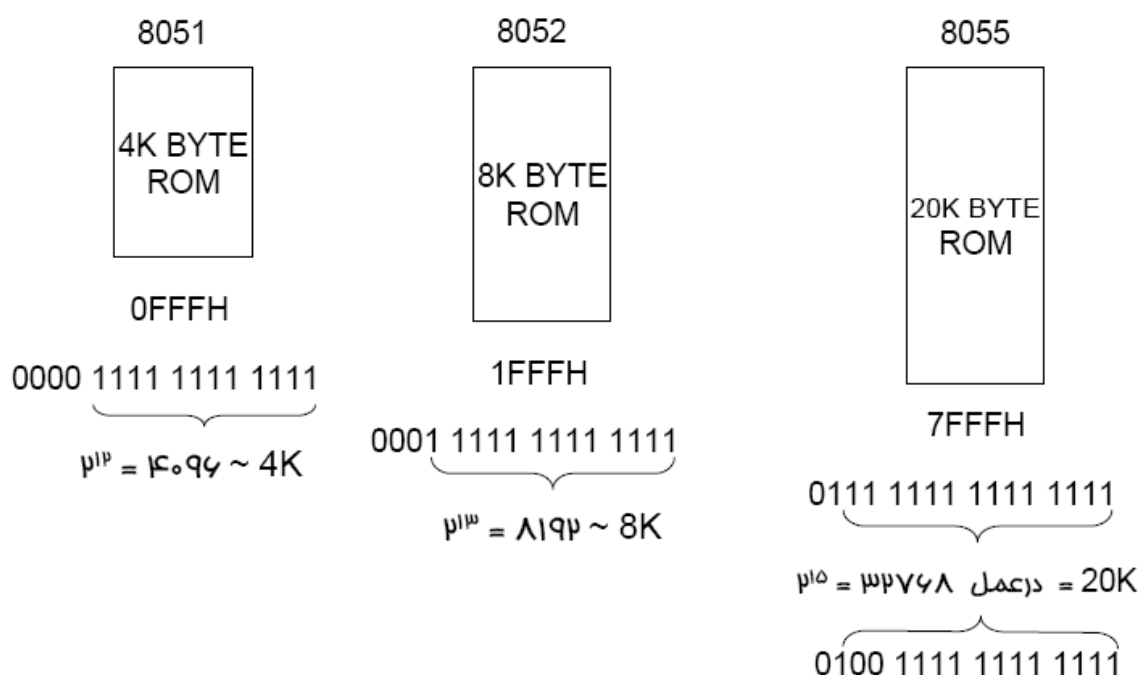
- * دارای 4K بایت حافظه ی Flash داخلی
- * تحمل ۱۰۰۰ بار نوشتن و پاک شدن
- * ماکزیمم فرکانس 24MHZ
- * 128*8 بیت حافظه ی RAM داخلی
- * ۳۲ خط ورودی و خروجی
- * دو تایمر یا کانتر ۱۶ بیتی
- * شش منبع وقفه ، کانال سریال قابل برنامه ریزی ، مدهای تغذیه ی توان پایین و کم مصرف

آدرس دهی فضای ROM or Flash:

آدرس فضای ROM or Flash میکروکنترلر AT89C51 از خانه ی 0000 تا 7FFF می باشد . به عبارت دیگر داریم :

$$0000_7FFF = 32768 \text{ bit} = 4096 \text{ byte} = 4\text{Kbyte}$$

اما فضای مورد استفاده ی ما از مجموعه ی آموزشی MDA-EMS51 ، از آدرس ۲۰۰۰ می باشد .
شکل زیر ، بیانگر مقدار حافظه ی ROM چند میکروکنترلر مهم و پر کاربرد می باشد .



آدرس دهی فضای RAM :

آدرس فضای RAM میکروکنترلر AT89C51 از خانه ی 00 تا 7F می باشد که شکل زیر آدرس دهی آن را نمایش می دهد :

تخصیص فضای حافظه در RAM برای میکرو 8051 :

7FH		برای کاربرد های دیگر
30H		
2FH		قابلیت آدرس دهی بیتی
20H		
1FH		REGISTER BANK 3
18H		
17H		
10H		REGISTER BANK 2
0FH		REGISTER BANK 1 (STACK)
08H		
07H		
00H		REGISTER BANK 0 { R0 .. R7 }

آشنایی و شرح وظایف قطعات مجموعه MDA-EMS51 :

در مجموعه ی آموزشی MDA-EMS51 (شکل زیر) ، قطعات و آی سی های زیادی وجود دارد که با مشاهده ی نقشه های مجموعه که در ضمیمه ی گزارش کار موجود است ، در مورد هر یک از قطعات آن ، توضیح مختصری می دهیم :



U11 : NEC 8255A → PPI with 24 I/O

U2 : 74LS373P → Flip flops & Latches ; octal D_type transparent latches with 3-state outputs

U3 : 27C512 → Memory IC (EPROM); CMOS 512Kbit (64KB*8) UV EPROM

U7 : 74LS138 → Decoder ; 3 line to 8 line DEC / DMAX

U13 : 74LS174P → Flip flops & Latches ; Hex D_type positive edge Triggered flip flops with clear

U9 : 74LS74 → Flip flops & Latches ; Dual positive edge Triggered D_type flip flops with clear & Preset

U5 : MAX232 → MULTI Channel RS-232 Drivers / Receivers industry standard

U18 : 74244 → Buffers & Drivers ; octal buffers /line drivers with tri-state outputs

TR8 ... TR11 → A1266 ; Transistor Amplifier or switching

U14 : 74LS04 → Gates / Inverters ; Hex inverters

کلید های کنترلی :

در این مجموعه کلیدهایی برای کنترل دستگاه و اجرای دستورات برنامه وجود دارد که به شرح زیر می باشد :

AD : مخفف Adress بوده و با فشردن این کلید ، ۱۶ بیت آدرس (چهار رقم HEX) از کاربر گرفته شده و محتوای داده ی آن نمایش داده می شود .

DA : مخفف Data بوده و با فشردن این کلید ، می توان در آدرس نمایش داده شده ، داده ی مورد نظر را نوشت . بدیهی است که خانه ی حافظه ی مورد نظر باید از نوع RAM باشد .

+ و - : برای افزایش و کاهش آدرس خانه ای از حافظه که در حال نمایش است ، بکار می رود .

GO : با فشردن این کلید ، میکروکنترلر برای اجرای برنامه به آدرس شروع 2000Hex می رود .

STP : برای اجرای خط به خط برنامه بکار می رود .

INT : با فشردن این کلید ، یک وقفه ی خارجی برای میکروکنترلر ایجاد می شود .

T0 : با فشردن این کلید ، یک پالس برای پایه ی T0 میکروکنترلر ایجاد می شود .

برنامه های ساده ، برای آشنایی با مجموعه :

PROGRAM 1 : AD 2000 → 9E
 0 0000 → 00
 + 0001 → 00
 + 0002 → 00
 - 0001 → 00

این برنامه نحوه ی آدرس دهی را نشان می دهد . به این ترتیب که محتوای آدرس ۲۰۰۰ ، مقدار 9E بوده و محتوای آدرس ۰ ، 00 می باشد . با کلیدهای + و - نیز خانه ی آدرس ها به ترتیب افزایش و کاهش می یابد .

PROGRAM 2 : AD 2000 → 9E
 0030 0030 → 00
 DA 0030 → 00
 AB 0030 → AB
 + 0031 → 00
 CD 0031 → CD

این برنامه نحوه ی آدرس دهی و مقداردهی (دادن دیتا) را نشان می دهد . به این ترتیب که محتوای آدرس ۰۰۳۰ ، که دارای مقدار اولیه ی ۰۰ بود ، به مقدار نهایی AB تغییر یافت و مقدار حافظه به آدرس ۰۰۳۱ از ۰ به CD ، مقدار دهی شد .

PROGRAM 3 : AD 2000 → 9E
 2000 2000 → 9E
 DA 2000 → 9E
 35 2000 → 35

در این برنامه دیتای حافظه به آدرس ۲۰۰۰ ، از 9E به 35 تغییر پیدا کرد .

PROGRAM 4 : 2000 74F7 MOV A,#0F7H
 2002 F590 MOV P1,A
 2004 028028 JMP 8028

در این برنامه LED شماره ی ۵ (چهارمی از بالا) روشن می شود . زیرا خروجی ، عدد ۸ یعنی ۱۰۰۰۱۰۰۰ باینری را نمایش می دهد ؛ که این عدد ، معکوس و وارون عدد F7 یعنی ۱۱۱۱۰۱۱۱ می باشد . برای اجرا کردن آن باید پس از وارد کردن کدها ، کلیدهای AD، RESET و GO را بترتیب بفشاریم . (در این جا نیازی به فشردن کلید STP نیست .)

```

PROGRAM 5 : 2000 308BFD      JNB IE1,$
              2003 74F7       MOV A,#0F7H
              2005 F590       MOV P1,A
              2007 028028     JMP 8028,MONITOR RETURN

```

در این برنامه نیز LED شماره ی ۵ (چهارمی از بالا) روشن می شود . تنها تفاوت آن با برنامه ی قبلی ، در نحوه ی اجرای آن می باشد که باید بجز کلیدهای گفته شده ، سه بار کلید STP را فشار دهیم . چون در خط اول این برنامه ، پرش وجود دارد و مشخص نیست که چه زمانی وقفه اعمال می شود . پس ، از کلید STP برای اجرای مرحله به مرحله ی برنامه استفاده می کنیم تا خروجی مشخص گردد .

تمرین ۱: برنامه ای بنویسید که با یک فرکانس قابل رویت ، چهار LED دو به دو روشن و خاموش شوند .

```

ORG 2000H
MAIN: MOV p1,#03h
      call DELAY
      MOV p1,#0Ch
      call DELAY
      JMP MAIN
DELAY: MOV R2,#8
LOOP 2: MOV R1,#250
LOOP 1: MOV R0,#250
LOOP 0: DJNZ R0,LOOP 0
        DJNZ R1,LOOP 1
        DJNZ R2,LOOP 2
RET

```

کد هگز تمرین اول :

این کد از طریق نرم افزار **Proteus** تولید شده و بر روی مجموعه ی آموزشی **MDA-EMS51** کاربرد دارد .
یعنی فقط این کد را وارد مجموعه کنید .

```

:1020000075900312200E75900C12200E80F27A99
:0C2010000879FA78FAD8FED9FADAF6223C
:00000001FF

```

کد هگز تمرین اول :

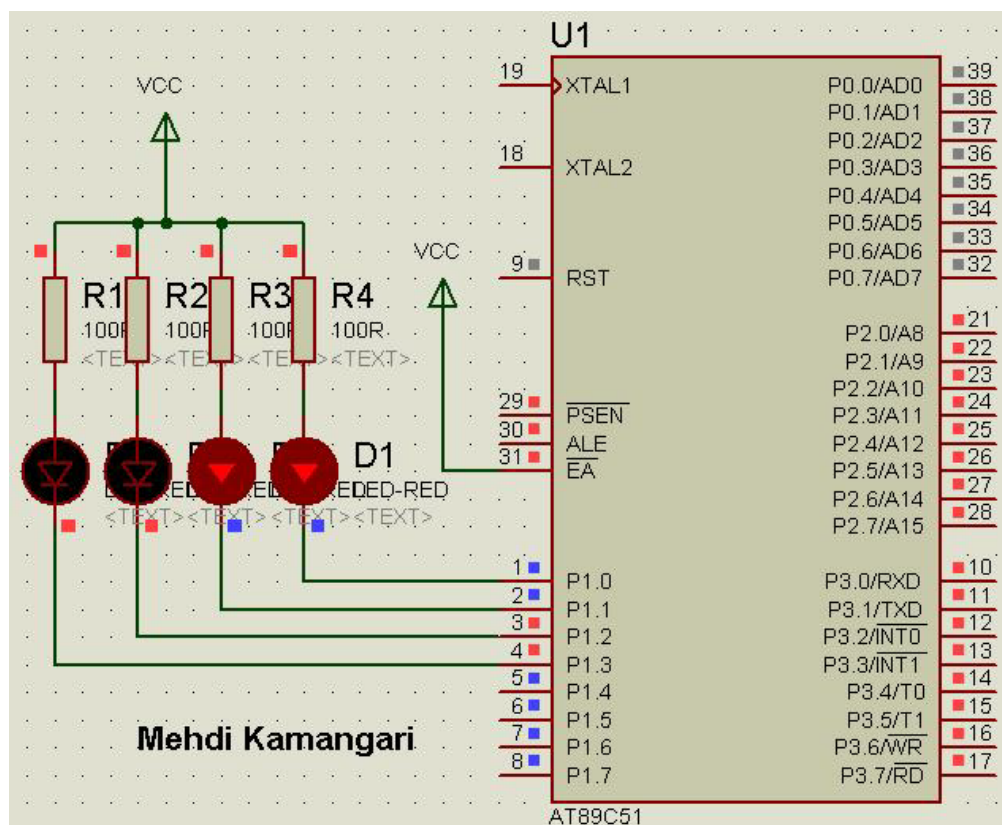
این کد از طریق نرم افزار **8051 IDE** تولید شده و برای شبیه سازی مدار ، بر روی نرم افزار **Proteus** کاربرد دارد .

```

:1020000075900312200F75900C12200F0220007A99
:0C2010000879FA78FAD8FED9FADAF6223C
:00000001FF

```

نکته: با کمال تعجب ، باید عنوان کنم که اگرچه برنامه ی SOURCE هر دو فایل هگز یکسان است ، ولی در دو نرم افزاری که فایل هگز را جداگانه تولید کردیم (یعنی یکبار در Proteus و یکبار در 8051 IDE) دو فایل هگز مختلف ایجاد شد !!! که عملکرد آنها به ترتیب در مجموعه ی آموزشی MDA-EMS51 و نرم افزار شبیه ساز Proteus می باشد و بر روی دیگری جواب نمی دهد !!!!



مدار تمرین اول

تمرین ۲: برنامه ای بنویسید که وضعیت DIPSW ها را بر روی LED ها نمایش دهد . به این مفهوم که اگر سوئیچ قطع است ، LED متناظر خاموش و در غیر اینصورت روشن شود .

```

ORG 2000H
LOOP: MOV A,P1
      SWAP A
      ORL A,#0F0H
      MOV P1,A
      JMP LOOP
  
```

روش دوم :

```

ORG 2000H
LOOP: MOV C,P1.4
      MOV P1.0,C
  
```



```

MOV C,P1.5
MOV P1.1,C
MOV C,P1.6
MOV P1.2,C
MOV C,P1.7
MOV P1.3,C
JMP LOOP

```

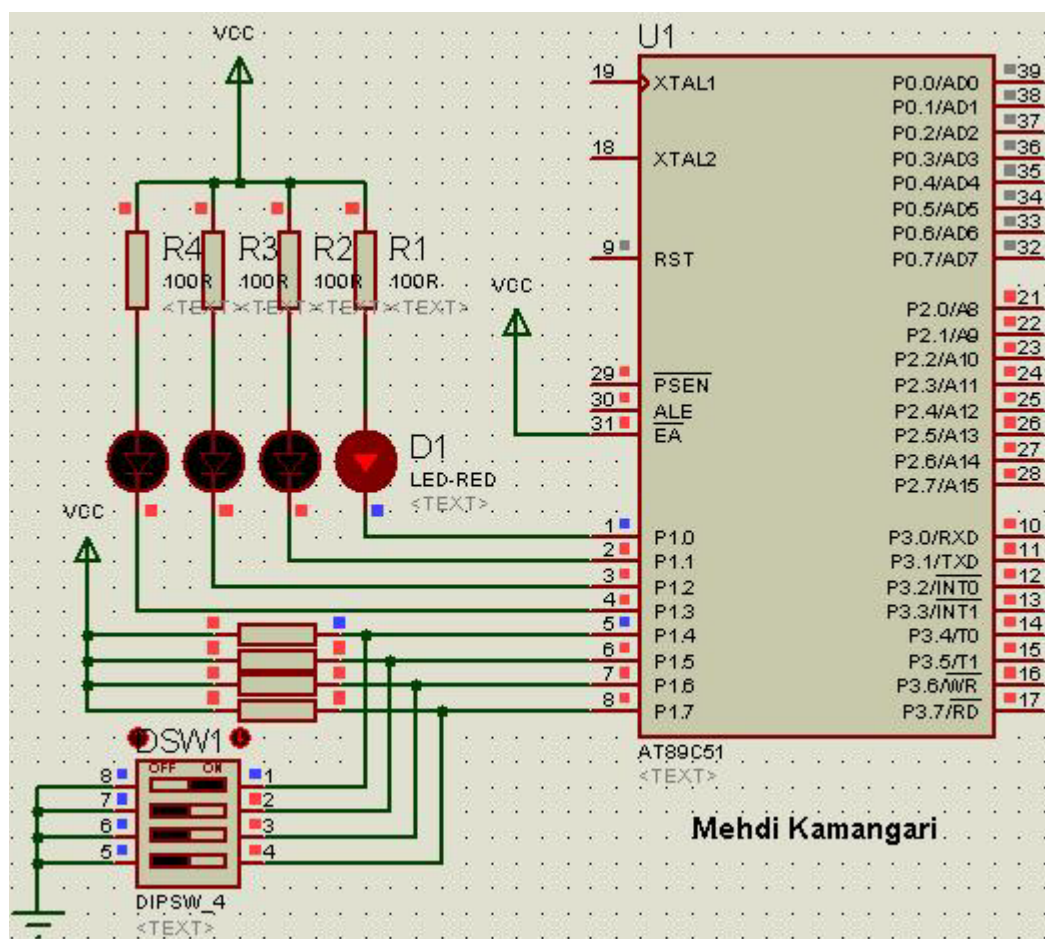
کد هگز تمرین دوم :

```

:0A200000E590C444F0F590022000C2
:000000001FF

```

کد هگز فوق در نرم افزار **8051 IDE** تولید شده و هم برای شبیه سازی مدار در Proteus و هم برای مجموعه ی آموزشی MDA-EMS51 قابل استفاده است .



کاری از : مهدی کمان گری ؛ kamangari@gmail.com

WWW.KAMANGARI.BLOGFA.COM