

"به نام یزدان پاک"



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

گزارش آزمایش اول اسمبلی  
اعضای گروه:

محمد چوپان ۹۸۳۱۱۲۵

محمد سپهر توکلی کرمانی ۹۸۳۱۱۱۱

تاریخ آزمایش : ۱۴۰۰/۱۰/۰۹

## آزمایش ۱:

### هدف آزمایش:

- آشنایی با شیوه پیکربندی رجیستر ها و مدیریت سطح پایین با زبان اسمبلی

### قطعات آزمایش:

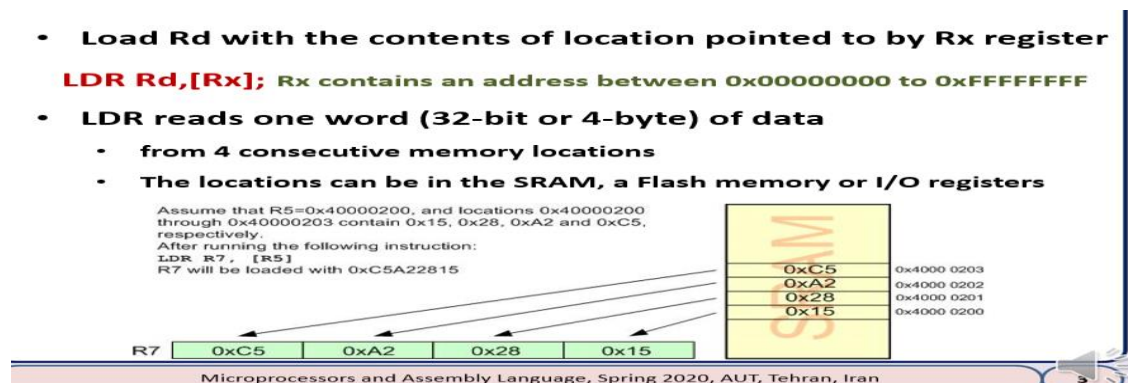
- میکروکنترلر ATSAM3N4A
- دیود نورانی LED
- کلید
- مقاومت  $220\Omega$
- مقاومت  $10K\Omega$

در این آزمایش باید توسط اسمبلی برنامه ای بنویسیم که بتواند چراغ هایی که ب پردازنده ما متصل است را روشن کند و برای راه انداختن هر کدام باید وقفه مرتبط با آن را اجرا کنیم.

### آنچه باید در پیش گزارش نوشته شود:

- توضیحات مختصری درباره ی دستورات MOV, LDR و STR دهید.
- ایده ای برای پیاده سازی تابع تاخیر در زبان اسمبلی ارائه دهید.
- به پرسش ها در بخش مقدمه ای در رابطه با keil که با رنگ قرمز مشخص شده است پاسخ دهید.

**دستور ldr:** این دستور به معنای load است که مقدار موجود در آن ادرس را در رجیستر مورد نظر ما ذخیره میکند همانند عکس زیر:

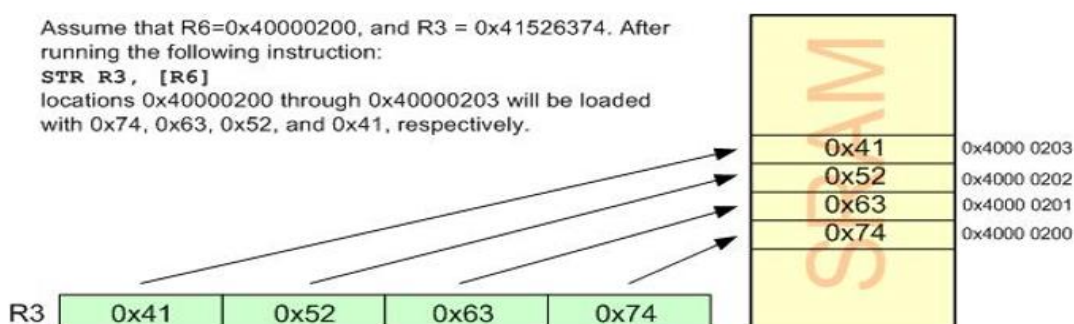


**دستور mov :** مقادیر موجود در یک رجیستر را به رجیستر دیگر انتقال می دهد که حال عملوند دوم میتواند یک عدد نیز باشد که با علامت # میتوان یک عدد را مشخص کرد و یا \_ مبنای آن را

**دستور str :** این دستور دستور store که عکس دستور ldr است و مقادیر موجود در یک رجیستر را درون حافظه می گذارد.

- **Store register Rd into locations pointed to by Rx**

### STR Rd,[Rx]



حال با دستوراتی مانند ldrb & strb تنها یک بایت را بار گذاری کرد.

و نحوه قرار گرفتن در حافظه نیز طوری است که بیت کم ارزش تر در حافظه کمتر قرار میگیرد.

● ایده ای برای پیاده سازی تابع تاخیر در زبان اسمبلی ارائه دهید.

برای پیاده سازی این تابع از یک حلقه می توانیم استفاده کنیم که با استفاده از آن برای مثال ۱۰۰ با یک حلقه را پیمایش کند .

```
loop_delay
    ADD r4,r4,#1

    cmp r4,r5
    bne loop_delay

    bx lr
```

- به پرسش ها در بخش مقدمه ای در رابطه با keil که با رنگ قرمز مشخص شده است پاسخ دهید.

بخش یک : بخش های stop build , batch build , rebuild , build , translate ابزار هایی که به هنگام ساختن کد مورد استفاده قرار می گیرند هستند پس از تغییر کد از دستور rebuild استفاده می کنیم. **با مطالعه در رابطه با بقیه دستور های گفته شده توضیحات مختصری ارائه دهید.**

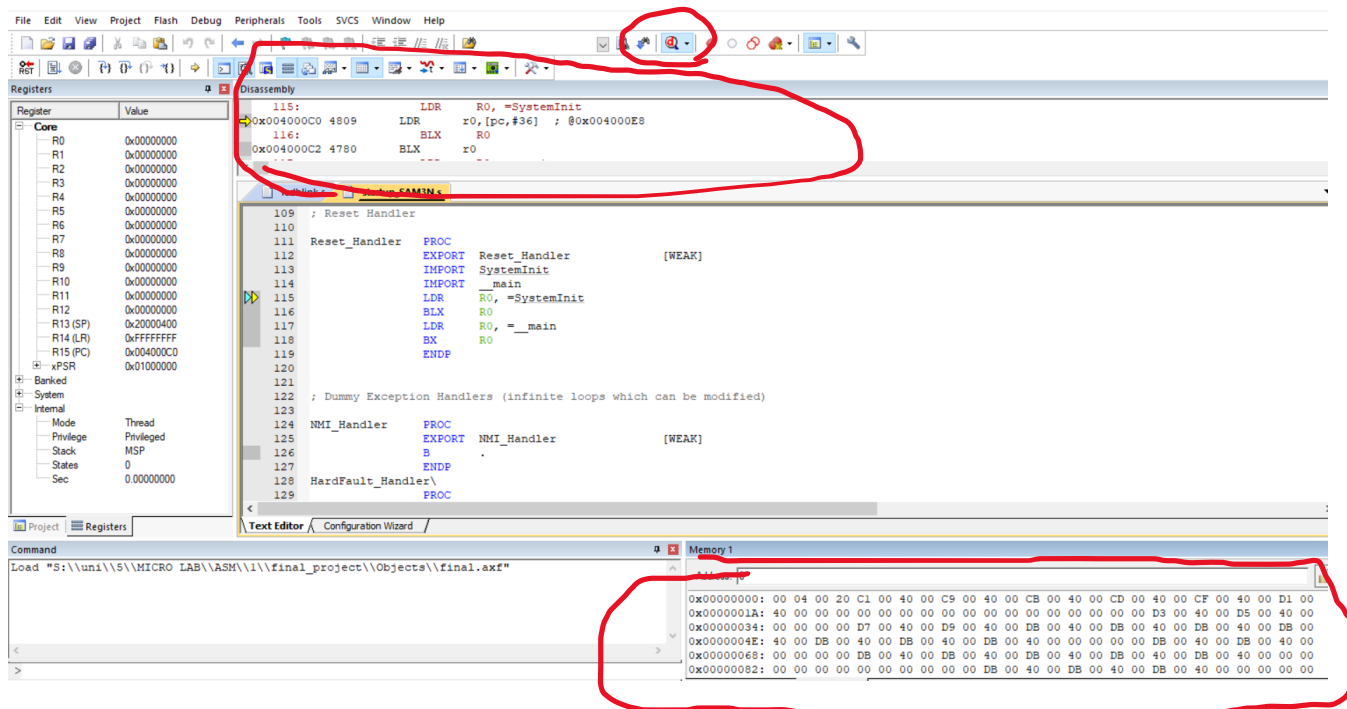
**دستور stop build :** این دستور تنها در هنگام build فعال است و build را متوقف میکند.

**دستور batch build :** این دستور compile & assemble میکند فایل هایی را که عوض شدند و object file آن ها را می سازد.

**Rebuild :** تمامی فایل های یک پروژه را از ابتدا build میکند.

**Build :** یک فایل را build کرده و object مورد نظر آن را می سازد.

**Translate :** برای مثال وقتی از متغیر ها و یا از زبان c یا زبان های دیگر در میان کد اسمبلی استفاده می کنم این ها باید ترجمه شوند و به deassembler که بتواند دستور ها را اجرا کند.



همانطور که میبینید علاوه بر این ها می توان ساختار deassembler و یا حافظه را نیز دید و debug کرد.

[Previous](#)

### 27.7.1 PIO Enable Register

Name: PIO\_PER

Addresses: 0x400E0E00 (PIOA), 0x400E1000 (PIOB), 0x400E1200 (PIOC)

Access: Write-only

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register can only be written if the WPEN bit is cleared in "PIO Write Protect Mode Register".

- P0-P31: PIO Enable

0 = No effect.

1 = Enables the PIO to control the corresponding pin (disables peripheral control of the pin).

### 27.7.10 PIO Set Output Data Register

Name: PIO\_SODR

Addresses: 0x400E0E30 (PIOA), 0x400E1030 (PIOB), 0x400E1230 (PIOC)

Access: Write-only

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

- P0-P31: Set Output Data

0 = No effect.

1 = Sets the data to be driven on the I/O line.

### 27.7.13 PIO Pin Data Status Register

Name: PIO\_PDSR

Addresses: 0x400E0E3C (PIOA), 0x400E103C (PIOB), 0x400E123C (PIOC)

Access: Read-only

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

#### • P0-P31: Output Data Status

0 = The I/O line is at level 0.

1 = The I/O line is at level 1.

### 27.7.17 PIO Interrupt Status Register

Name: PIO\_ISR

Addresses: 0x400E0E4C (PIOA), 0x400E104C (PIOB), 0x400E124C (PIOC)

Access: Read-only

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

#### • P0-P31: Input Change Interrupt Status

0 = No Input Change has been detected on the I/O line since PIO\_ISR was last read or since reset.

1 = At least one Input Change has been detected on the I/O line since PIO\_ISR was last read or since reset.

کد را مطابق این داده ها تنظیم میکنیم.

```
AREA MY_DATA, DATA
PIO_PER equ 0x400E0E00
PIO_SODR equ 0x400E0E30
PIO_CODR equ 0x400E0E34
PIO_OER equ 0x400E0E10
PIO_PDSR equ 0x400E0E3C
PIO_ISR equ 0x400E0E4C
```

باقی کد را پیاده سازی میکنیم.

```
2  __main
3      bl enable_pi01
4      bl enable_pi02
5      bl enable_pi03
6
7  begin
8      ldr r1, =PIO_ISR
9      ldr r2, [r1]
0      cmp r2, #2_100000
1      beq loop
2      b begin
3
4
5  finish
6      bl led_off1
7      bl led_off2
8      bl led_off3
9      b begin
0
```



```

bl led_on1

bl led_off2

bl delay

ldr r1, =PIO_ISR
ldr r2, [r1]
cmp r2, #2_1000000
beq finish

bl led_off1

bl led_off3

bl delay

ldr r1, =PIO_ISR
ldr r2, [r1]
cmp r2, #2_1000000
beq finish

bl led_on2

bl led_on1

bl delay

ldr r1, =PIO_ISR
ldr r2, [r1]

```

```

loop
    ldr r1, =PIO_ISR
    ldr r2, [r1]
    cmp r2, #2_1000000
    beq finish

    bl led_on1
    bl led_on2
    bl led_on3

    bl delay

    ldr r1, =PIO_ISR
    ldr r2, [r1]
    cmp r2, #2_1000000
    beq finish

    bl led_off1

    bl delay

    ldr r1, =PIO_ISR
    ldr r2, [r1]
    cmp r2, #2_1000000
    beq finish

    bl led_on1

```

```

led_off3
    mov r4, #2_100

    ldr r5,=PIO_CODR
    str r4, [r5]

    bx lr

delay
    mov r4, #0
    ldr r5, =0x00C0000

loop_delay
    ADD r4,r4,#1

    cmp r4,r5
    bne loop_delay

    bx lr

end

led_on2
    mov r4, #2_10

    ldr r5,=PIO_SODR
    str r4, [r5]

    bx lr

led_on3
    mov r4, #2_100

    ldr r5,=PIO_SODR
    str r4, [r5]

    bx lr

led_off1
    mov r4, #2_1

    ldr r5,=PIO_CODR
    str r4, [r5]

    bx lr

led_off2
    mov r4, #2_10

    ldr r5,=PIO_CODR
    str r4, [r5]

    bx lr

enable_pio2
    mov r4, #2_10

    ldr r5,=PIO_PER
    str r4, [r5]

    ldr r5,=PIO_OER
    str r4, [r5]

    bx lr

enable_pio3
    mov r4, #2_100

    ldr r5,=PIO_PER
    str r4, [r5]

    ldr r5,=PIO_OER
    str r4, [r5]

    bx lr

led_on1
    mov r4, #2_1

    ldr r5,=PIO_SODR
    str r4, [r5]

    bx lr

cmp r2, #2_1000000
beq finish

bl led_off1

bl delay

ldr r1, =PIO_ISR
ldr r2, [r1]
cmp r2, #2_1000000
beq finish

bl led_off2

bl loop

table_pio1

    mov r4, #2_1

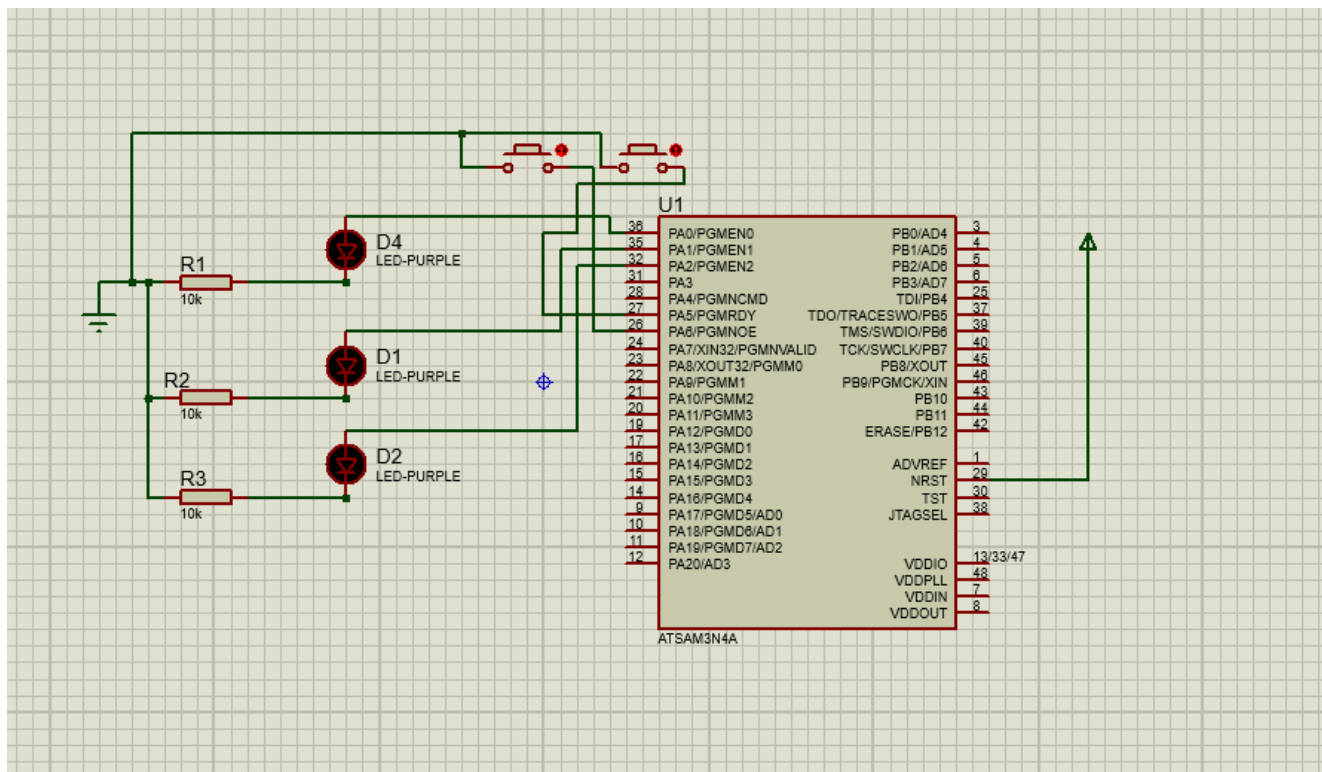
    ldr r5,=PIO_PER
    str r4, [r5]

    ldr r5,=PIO_OER
    str r4, [r5]

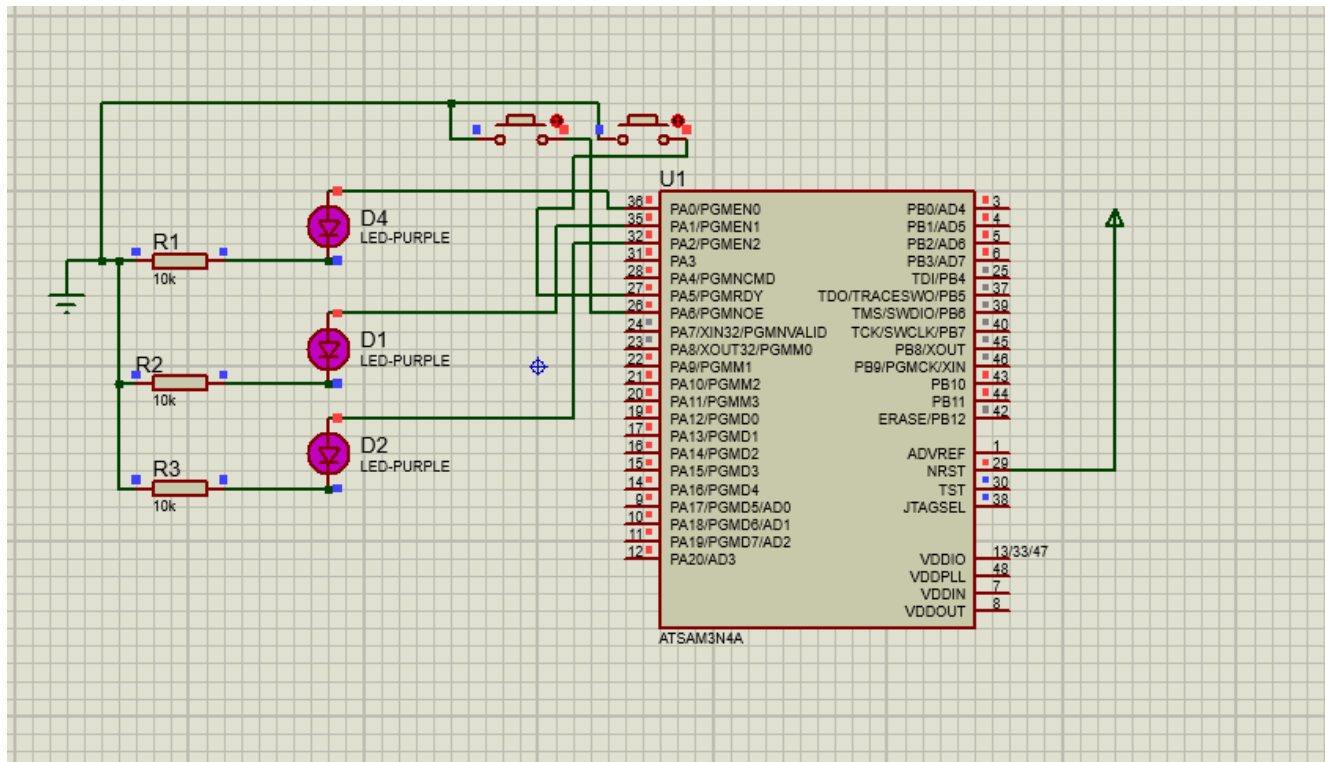
    bx lr

```

فایل hex را خروجی میگیریم . سپس مدار را پیاده سازی میکنیم.



وقتی کلید ۱ را فشار می دهیم:



و کلید ۲:

