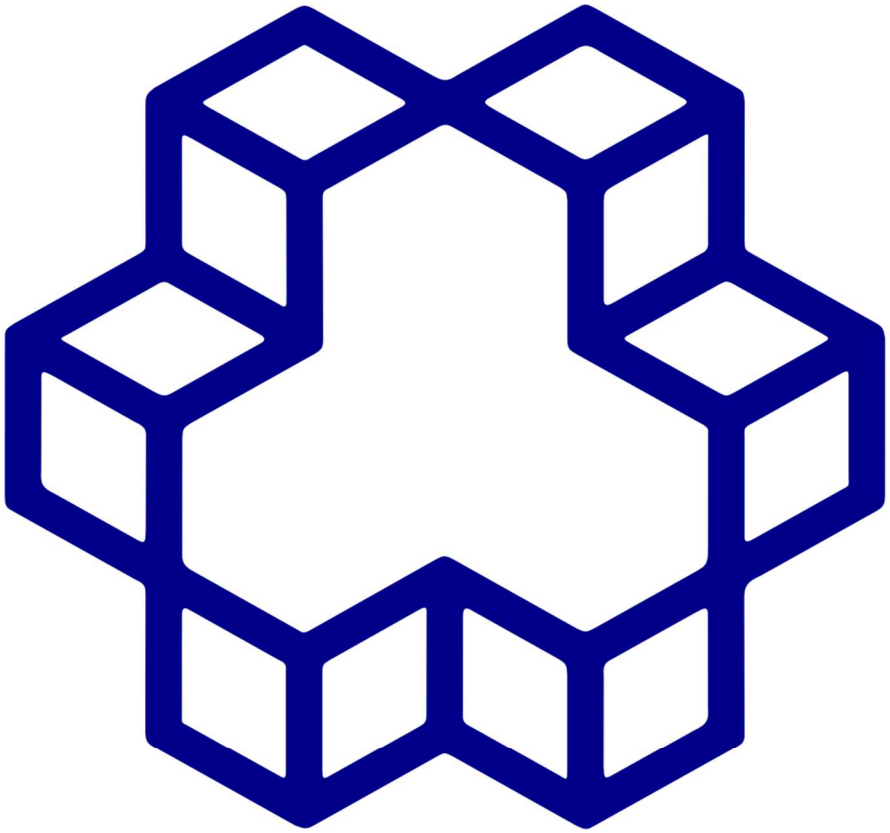


دانشگاه صنعتی خواجه نصیرالدین طوسی

درس سیستم دیجیتال 2

پروژه : قفل هوشمند رفتاری

نام استاد : استاد دلیر

نام تهیه کننده : سیاوش ابراهیمی کمال

مقدمه :

در این پروژه قصد داریم Atmega64 را به نحوی برنامه ریزی کنیم که مانند یک قفل هوشمند عمل کند که وارد کردن رمز به صورت اندازه گیری طول فشار (مانند کد مورس) است
از **الگوی فشردن کلید توسط کاربر** برای آموزش و احراز هویت استفاده می‌شود. این الگو شامل **توالی بیت‌هایی از نوع فشردن (کوتاه/بلند)** به‌علاوه **فاصله‌های زمانی بین فشردن‌ها** است.

سیستم در چهار حالت اصلی کار می‌کند:

1. اولیه : چراغ سبز به صورت چشمک زن است و منتظر دریافت دستور از کاربر .
2. **آموزش رمز :(Learning)** کاربر الگوی دلخواه خود را به‌عنوان رمز وارد می‌کند.
3. **احراز هویت :(Authentication)** کاربر تلاش می‌کند رمز را تکرار کند، و سیستم تطابق را بررسی می‌کند.
4. **حالت قفل :(Lock Mode)** در صورت وارد کردن سه رمز اشتباه، سیستم قفل شده و فقط با **رمز اضطراری** قابل آزادسازی است.

نمایش وضعیت سیستم از طریق **LEDهای متصل به PORTA** انجام می‌شود. از تایمرها برای اندازه‌گیری مدت فشار (تعیین صفر یا یک بودن بیت)، زمان‌بندی‌های مربوط به چشمک زدنLED ها و کنترل فاصله بین فشردن‌ها استفاده شده است.

از **وقفه‌های خارجی INT0 تا INT4** و **تایمرهای Timer0 و Timer2** برای مدیریت ورودی‌ها، زمان‌ها و حالت‌ها استفاده شده است. حافظه‌ی RAM داخلی برای ذخیره‌سازی رمز آموزش داده‌شده و رمز ورودی کاربر به کار رفته است.

از Timer0 برای اندازه گیری فاصله ها در هر فشار دادن int0 استفاده میشود
و از timer2 برای مشخص کردن نوع کار کرد Led ها استفاده میشود .

از portc برای مشخص کردن مقدار رمز در زمان ورودی گرفتن استفاده شده است . که به صورت اتصال 7 led است که به صورت باینری عدد نمایش داده شود .

لازم به ذکر است که فرکانس پردازنده برابر $1.024\ Mhz$ است .

آماده سازی این پروژه در نرم افزار ATMEL STUDIOانجام گرفته است و همچنین شبیه سازی از نرم افزار proteuos استفاده شده .

یک فیلم در کنار فایل های پروژه آماده شده است که انجام شبیه سازی و توضیحات کد و اجرای کد است .

در این بخش کتابخانه مشخص شده

```
.Include "M64DEF.INC"
```

ین بخش آدرس‌های وقفه ها و تایمر ها را مشخص می‌کند. به‌محض وقوع هر وقفه، پردازنده به روتین مربوطه می‌پرد.

```
.org 0x0000
    jmp main
.org 0x0002
    jmp INT0_ISR
.org 0x0004
    jmp INT1_ISR
.org 0x0006
    jmp INT2_ISR
.org 0x0008
    jmp INT3_ISR
.org 0x000A
    jmp INT4_ISR
.org 0x0014
    jmp TIMER2_OVF_ISR

.org 0x0020
    jmp TIMER0_OVF_ISR

.org 0x0050
```

این آدرس‌ها در حافظه داخلی برای ذخیره‌سازی داده‌های رمز استفاده می‌شوند

که شامل مقدار فشار در هر رقم رمز ، نوع 0 یا 1 بودن و طول رمز است

```
; TIMER OF HOLDING :
; 0X1001 ==> LENGTH OF PASSWORD IN LEARNING
; 0X1002 ==> BIT1
; 0X1003 ==> BIT2
; 0X1004 ==> BIT3
; 0X1005 ==> BIT4
; 0X1006 ==> BIT5
; 0X1007 ==> BIT6
; 0X1008 ==> BIT7
; 0X1009 ==> BIT8
; 0X1010 ====> PASSWORD 0 1
; 0X1051 ==> LENGTH OF PASSWORD IN AUTHENTICATION
; 0X1052 ==> BIT1
; 0X1053 ==> BIT2
; 0X1054 ==> BIT3
; 0X1055 ==> BIT4
; 0X1056 ==> BIT5
; 0X2057 ==> BIT6
; 0X2058 ==> BIT7
; 0X1059 ==> BIT8
; 0X1060 ====> PASSWORD 0 1
```

بخش initialشروع میشود که در ابتدا آدرس stack مشخص شده است

```
main:
    ;stack
    LDI R16, HIGH(RAMEND)
    OUT SPH, r16
    LDI R16, LOW(RAMEND)
    OUT SPL, r16
```

در اینجا نوع وقفه ها را مشخص کردیم
همچنین در ابتدا تمام وقفه ها به جز وقفه 4 را فعال کرده ایم .

```
;INT0-4 --> LOW EDGE
LDI R16, 0b10101010
STS EICRA, R16
; falling edge  INT4
LDI R16 , 0X02
OUT EICRB, R16
; Enable INT0 -- INT4
LDI R16, 0b00001111
OUT EIMSK, R16
```

در این بخش نوع کار کرد کلاک را مشخص کردیم
هر دو کلاک با پیش تقسیم کننده 1024 کار میکنند
همچنین با توجه به فرکانس خارجی کلاک میدانیم که با هر بار پر شدن تایمر مقدار 50ms گذشته است
یعنی واحد هایمان به صورت 50ms است .

در ابتده فقط تایمر 2 روشن میشود که نوع LED مورد نظر روشن شود

```
   ;Timer0:  Prescaler 1024
   LDI R16, 0x07
   OUT TCCR0, R16
   LDI R16 , 206
   OUT TCNT0 , R16
   ; Timer2: Prescaler 1024 (for LED)
   LDI R16, 0x05
   OUT TCCR2, R16
   LDI r16, 206
   OUT TCNT2, R16
   ; ENABLE TIMERS  ==> timer 2
   LDI R16 , 0X40
   OUT TIMSK, R16
   LDI R16, 0b11111111  ; PA0-PA4 as outputs
   OUT DDRA, R16
   OUT DDRC, R16; TEST
```

پورت های A , B , Cبه عنوان خروجی تعیین شده است . که A برای LED ها و B تعداد خطا و C مقدار روز وارد شده در حالت آموزش یا احراز هویت است .

```
;PORTA (LEDs )- PORTB (wrong counter)
LDI R16, 0b11111111
OUT DDRA, R16
OUT DDRC, R16; TEST
OUT DDRB, R16
```

```
    CLR R0  ; LENGTH OF LEARNING PASSWORD
    CLR R1  ; LENGTH OF AUTHENTICATION PASSWORD
    CLR R3  ; CODE LEARNING
    CLR R4  ; CODE AUTHENTICATION
    CLR R16 ; TEMP
    CLR R17 ; OUTPUT PORT A
    CLR R18 ; TEMP TIMER2
    CLR R19 ; MODE A
    CLR R20 ; TEMP TIMER0
    CLR R21 ; END OF LED
    clr R22 ; TYPE EDGE
    clr R23 ; TYPE OF LEARNING OR AUTHENTICATION
    clr R25 ; WRONG COUNTER
    LDI R26 , 0X02
    LDI R27 , 0X10  ; X POINTER ==> 0X0102 ### R27 = 01 & R26 = 02
    LDI R28 , 0X52
    LDI R29 , 0X10  ; Y POINTER ==> 0X0202 ### R29 = 02 & R28 = 02
    CLR R30 ;  TYPE LEVEL OUT
    CLR R31 ;  FLAG OF
    SEI
```

در این رجیستر های که در برنامه به آن نیاز داریم مقدار اولیه داده شده همچین کار برد هر کدام در جلوی آنها کامنت شده است از پونتر های Y , X برای ذخیره در حافظه استفاده شده .

نکته بسیار مهم استفاده از **SEI** در آخر کد است که برای فعال کردن وقفه ها است .

```
    SEI
    ;================================
MAIN_LOOP:
    OUT PORTA , R17
    JMP MAIN_LOOP
    ;================================
```

حلقه اصلی که رجیستر r17 را در پورت A نمایش میدهد با انیکار LED مورد نظرمان روشن میشود نوع LED در تایمر 2 مشخص میشود

```
INT2_ISR:
    LDI R19 , 0X01 ; led yellow
    RETI
```

وقفه ی دوم که با زدن آن به حالت آموزش میرویم و نوع LED که میخوایم روشن کنیم هم از طریق رجیستر R19 مشخص میشود

```
INT0_ISR:
    CPI R23 , 0
    BREQ  LEARNING
    CPI R23   1
AUTHENTICATION :
    CPI R22 , 0  ; LOW EDGE OR HIGH EDGE
    BREQ low_edge_int1
    CPI R22 , 1  ; LOW EDGE OR HIGH EDGE
    BREQ high_edge_int1
low_edge_int1 :
    LDI R16  ,206
    OUT TCNT0 , R16
    LDI R16 ,0X01
    OUT TIMSK , R16
    LDI R16 , 0b10101011
    STS eicra , R16
    LDI R22 , 1
    RETI
high_edge_int1 :
    ST Y+ , R20
    INC R1
    CPI R20 , 8
    BRCC ITS_ONE1
    BRCS ITS_ZERO1
    ITS_ONE1 :

    LSL R4
    INC R4
    JMP CONTINUE1
    ITS_ZERO1 :
    LSL R4
    CONTINUE1 :
    CLR R20
    LDI R16 ,0X01
    OUT TIMSK , R16
    LDI R16  ,206
    OUT TCNT0 , R16
    LDI R16 , 0b10101010
    STS EICRA , R16
    LDI R22 , 0
    RETI
```

اینتراپت صفر که برای ورودی گرفتن از کاربر است که برای انتخاب نوع کارکرد (آموزش یا احراز هویت) توسط رجیستر R23 استفاده شده است

R22 مشخص می‌کنه لبه فعلی پایینه یا بالاست (Low/High Edge).

بسته به این مقدار به توابع مربوطه می‌ره.

این بخش با شروع فشار کلید اجرا می‌شه:

- مقدار تایمر صفر (Timer0) ریست می‌شه تا طول فشار اندازه‌گیری بشه.
- وقفه تایمر فعال می‌شه.
- لبه بعدی به بالا تنظیم می‌شه (برای تشخیص پایان فشار).
- R22 = 1یعنی منتظر لبه بالا هستیم.

این قسمت با **رها کردن کلید** اجرا می‌شه:

- مقدار مدت زمان فشار (R20) را در RAM ذخیره می‌کنه (یعنی بیت رمز وارد شده.)
- اگر بیشتر از ۸ بوده، بیت را ۱ حساب می‌کنه (فشار بلند).
- اگر کمتر، بیت را ۰ حساب می‌کنه (فشار کوتاه).

R4 رمز وارد شده توسط کاربر در حالت احراز.

R1 تعداد بیت‌هایی که کاربر وارد کرده.

پس از ثبت بیت، دوباره سیستم را برای شروع فشردن بعدی آماده می‌کنه.

در حالت learning عملکرد مشابه حالتAUTH ، فقط داده‌ها به مکان دیگری در حافظه می‌رن.

مشابه حالت بالا، اما:

+Xذخیره رمز آموزش

R3رمز آموزش‌شده

R0طول رمز آموزش

```
    ;=============================================================================================
INT1_ISR:
    LDI R19 , 3
    LDI R23 , 1
    RETI
    ;=============================================================================================
```

وقفه ی 1 برای رفتن به حالت احراز هویت و روشن کردن led آبی

وقفه - INT3 رمز اضطراری

بررسی می‌کنه فشرده شدن کلید INT3در حالت Low یا High بوده.

```
INT3_ISR:
    CPI R22 , 0  ; LOW EDGE OR HIGH EDGE
    BREQ low_edge_int3
    CPI R22 , 1  ; LOW EDGE OR HIGH EDGE
    BREQ high_edge_int3
low_edge_int3 :
    LDI R16, 0b00010000
    OUT EIMSK , R16
    LDI R22, 0X00
    RETI
high_edge_int3 :
    LDI R16 , 0b10101010
    STS EICRA , r16
    LDI R22 , 0
    clr R25
    OUT PORTB , R25
    ldi R19  , 0
    ldi R16  , 0x40
    out TIMSK , r16
    RETI
```

فقط INT4 فعاله می‌مونه (برای اجرای مرحله دوم رمز اضطراری)

اگر دکمه اضطراری درست وارد بشه:

- سیستم از حالت قفل درمیاد.
- شمارنده خطا صفر می‌شه(R25 = 0)
- LEDها ریست می‌شن.

وقفه ی چهارم : مرحله ی دوم برای رمز اضطراری :

```
INT4_ISR:
    CPI R22 , 0  ; LOW EDGE OR HIGH EDGE
    BREQ low_edge_int4
    CPI R22 , 1  ; LOW EDGE OR HIGH EDGE
    BREQ high_edge_int4
low_edge_int4 :
    SEI
    LDI R16  ,206
    OUT TCNT0 , R16
    LDI R16 ,0X01
    OUT TIMSK , R16
    LDI R16 , 0b00000011
    STS EICRB , r16
    LDI R22 , 1
    RETI
high_edge_int4 :
    CPI R20 ,10
    BRCS IN_LOCK
    CPI R20 , 20
    BRCC IN_LOCK
    LDI R16 , 0b11101010
    STS EICRA , R16
    LDI R16 , 0b00001000
    OUT EIMSK , R16
    RETI
IN_LOCK :
    LDI R16 , 0b00001000
    OUT EIMSK , R16
    CLR R22
    RETI
```

مثل بقیه‌ی وقفه‌ها، بررسی می‌کنه که لبه‌ی فعلی پایین (شروع فشردن کلید) هست یا بالا (رها کردن کلید).

این بخش، اندازه‌گیری مدت فشردن کلید INT4را شروع می‌کنه تا مشخص بشه آیا شرایط رمز اضطراری رعایت شده یا نه.

اگر R20(مدت فشردن کلید (INT4 بین **۱۰ تا ۲۰ واحد زمانی** بود، یعنی رمز اضطراری صحیح وارد شده.

در این حالت فقط INT3 فعال می‌مونه. یعنی قفل باز شده و سیستم به حالت آماده برمی‌گرده.

اگر شرایط رمز اضطراری رعایت نشه، سیستم همچنان در حالت قفل باقی می‌مونه و تنها INT3 فعاله.

```
TIMER2_OVF_ISR :
    CPI R19 , 0
    BREQ  LED_GREEN_DUCY_50
    CPI R19 , 1
    BREQ LED_YELLOW
    CPI R19 , 2
    BREQ LED_WHITE
    CPI R19 , 3
    BREQ LED_BLUE
    CPI R19 , 4
    BREQ CORRECT
    CPI R19 , 5
    BREQ WRONG_INPUT1
    CPI R19 , 6
    BREQ LOCK_MODE1
    RETI
```

رجیستر R19مشخص می‌کنه که سیستم در چه حالتی هست و کدوم الگوی LED باید اجرا بشه. برای هر حالت یک روتین LED جداگانه تعریف شده

```
LED_GREEN_DUCY_50 :
    LDI R16, 206
    OUT TCNT2, R16
    CPI R30,1
    BREQ HIGH_LEVEL0
    LDI R17 , 0X00
    INC R18
    CPI R18,10
    BRNE NEXT0
    CLR R18
    LDI R30,1
    RETI
HIGH_LEVEL0:
    LDI R17, 0x01
    INC R18
    CPI R18,10
    BRNE NEXT0
    CLR R30
    CLR R18
    RETI
NEXT0:
    RETI
```

LEDسبز با ٪۵۰ دیوتی‌سایکل چشمک می‌زنه – یعنی هر ۱۰ واحد زمانی خاموش و روشن به تناوب .

```
LED_YELLOW :
    LDI R17 , 0X02
    OUT PORTA, R17
    RETI
```

LED_YELLOW: فقط LED زرد روشن می‌شه. به معنی آن است که ما در حالت آموزش هستیم .

```
LED_BLUE:
    LDI R17 , 0X08
    OUT PORTA, R17

    RETI
```

LED_BLUE: LEDآبی روشن می‌شه (در حالت ورود به احراز).

LED_WHITE سفید چشمک می‌زنه تا حالت اتمام آموزش رو نشون بده.

که شرایط آن طبق دستور پروژه ا نجام شده است

```
LED_WHITE :
    LDI R16, 206
    OUT TCNT2, R16
    CPI R21 , 40
    BRNE NOT_END_WHITE
    CLR R21
    LDI R19 , 0
    JMP LED_GREEN_DUCY_50
NOT_END_WHITE:
    CPI R30,1
    BREQ HIGH_LEVEL2
    LDI R17 , 0X00
    INC R18
    INC R21
    CPI R18,5
    BRNE NEXT2
    CLR R18
    LDI R30,1
    RETI
```

```
HIGH_LEVEL2:
    INC R21
    LDI R17, 0x04
    INC R18
    CPI R18,5
    BRNE NEXT2
    CLR R30
    CLR R18
    RETI
NEXT2:
    RETI
```

CORRECT, WRONG_INPUT, LOCK_MODE:همگی حالت‌های خاص برای نتیجه رمز هستند.

```
CORRECT:
    LDI R16, 206
    OUT TCNT2, R16

    CPI R21 , 60
    BRNE NOT_END_GREEN
    LDI R21 , 0X00
    ldi R19 , 0
    jmp LED_GREEN_DUCY_50
NOT_END_GREEN:
    CPI R30,1
    BREQ HIGH_LEVEL4
    LDI R17 , 0X00
    INC R18
    INC R21
    CPI R18 ,3
    BRNE NEXT4
    CLR R18
    LDI R30,1
    RETI
```

در این حالت برای زمانی است که رمز توسط کاربر درست وارد شده و LED سبز با تناوب 500ms و دیوتی سایکل 70 درصد به مدت 3 ثانیه روشن میشود

```
HIGH_LEVEL4:
    INC R21
    LDI R17, 0x01
    INC R18
    CPI R18,7
    BRNE NEXT4
    CLR R30
    CLR R18
    RETI
NEXT4:
    RETI

;###################################
WRONG_INPUT:

    LDI R16, 206
    OUT TCNT2, R16

    CPI R21 , 40
    BRNE NOT_END_WRONG_INPUT
      LDI R19 , 0X08
      jmp LED_BLUE
      NOT_END_WRONG_INPUT:
      INC R21
      LDI R17 , 0X10
      OUT PORTA, R17
      RETI
 ;#########################
 LOCK_MODE:
      LDI R16, 156
      OUT TCNT2, R16
      CPI R30,1
      BREQ HIGH_LEVEL6
      LDI R17 , 0X00
      INC R18
      CPI R18,8
      BRNE NEXT6
      CLR R18
      LDI R30,1
      RETI


 HIGH_LEVEL6:
     LDI R17, 0x10
     INC R18
     CPI R18,12
     BRNE NEXT6
     CLR R30
     CLR R18
     RETI
 NEXT6:
     RETI
```

برای زمانی است که کاربر رمز را اشتباه میزند

برای زمانی است که کاربر 3 بار رمز را اشتباه زده

و LED قرمز به صورت گفته شده روشن خاموش شود

```
TIMER0_OVF_ISR :
SEI
    LDI R16  ,206
    OUT TCNT0 , R16
    INC R20
    CPI R20 , 60
    BREQ CHOOSING_MODE_OF_CORNOMETER
    RETI
    CHOOSING_MODE_OF_CORNOMETER :
    CPI R23 , 0
    BREQ END_OF_LEARNING
    CPI R23 , 1
    BREQ END_OF_AUTHENTICATION
    RETI
```

هدف این تایمر اندازه‌گیری زمان بین فشردن‌ها و پایان دوره آموزش یا احراز هویت است. هر سرریز 1 واحد به رجیستر شمارنده R20 اضافه می‌کند. اگر مقدارش به ۶۰ برسد، یعنی ۳ ثانیه) با prescaler مناسب (گذشته

←انتخاب حالت بعد از اتمام زمان

```
END_OF_LEARNING :
    LDI R16, 206
    OUT TCNT2, R16
    LDI R16, 0X40
    OUT TIMSK, R16
    OUT PORTC , R3 ; DEBUG
    LDI R19 , 2
    CLR R18
    CLR R20

    END_OF_AUTHENTICATION:
        CLR R18
        CLR R20
        LDI R16, 206
        OUT TCNT2, R16
        LDI R16, 0X40
        OUT TIMSK, R16
        OUT PORTC , R4 ; DEBUG
        CP R0 ,R1
        BRNE DIFFRENT_SIZE
        BREQ SAME_SIZE
    DIFFRENT_SIZE :
        LDI R17  , 0X10
        OUT PORTA , R17
        CLR R1
        INC R25
        CLR R4
        OUT PORTB , R25
        CPI R25 , 3
        BREQ LOCK
        LDI R19 , 0
```

برای زمانی است که 3 ثانیه از آخرین فشار در حالت آموزش گذشته است

و دستور لازم را برای روشن شدن چراغ سفید انجام میدهد .

در ادامه، سیستم ابتدا **تعداد بیت‌های رمزها** R0و R1را مقایسه می‌کند؛

در صورت برابر بودن سپس **مقدار نهایی رمزها** R3و R4را بررسی می‌کند تا صحت ورود رمز مشخص شود.

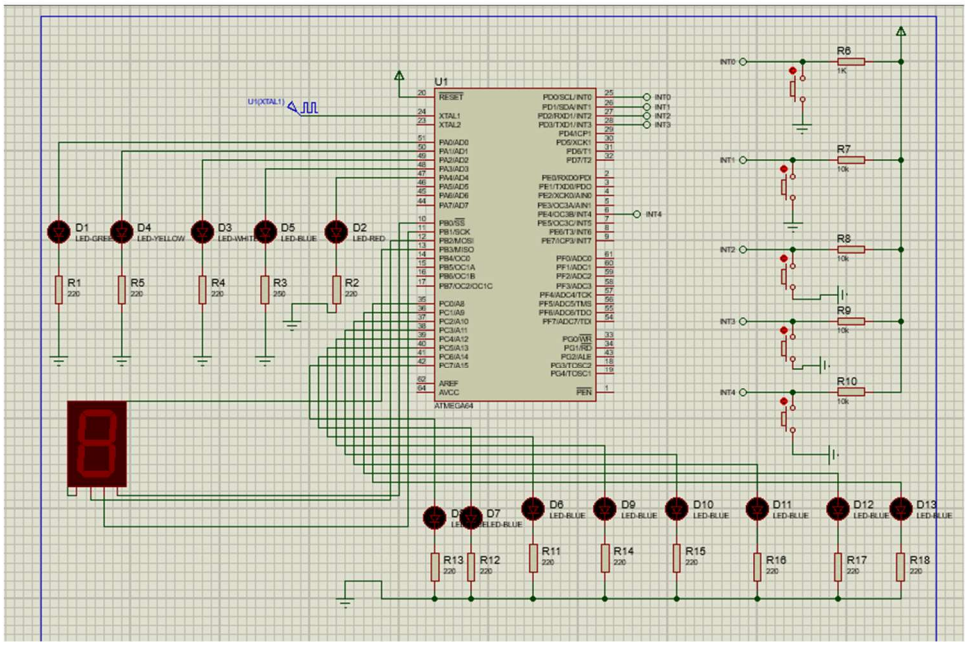
اگر اشتباه بود مقدار اشتباه ها در سون سگمنت (پورت B) یک واحد زیاد میشود

در هر مرحله نیز LED مخصوص به حالت خودش در صورت اشتباه شدن روشن میشود

```
SAME_SIZE :
    CP R3 , R4
    BREQ is_equal
    INC R25
    CLR R4
    OUT PORTB , R25
    LDI R19 , 5
    JMP WRONG_INPUT
    CPI R25 , 3
    BREQ LOCK
    LDI R19 , 0
    RETI

is_equal :
    LDI R19, 4
    LDI R16 , 0X0FF
    OUT PORTC , R16
    LDI R25 , 0X00
    OUT PORTB , R25
    CLR R4

RETI

LOCK :
    LDI R16 , 0b00001000
    OUT EIMSK , R16
    LDI R19 , 6
    LDI R26  ,0X01
    LDI R27 , 0X10
    OUT PORTB , R25

    RETI
```

در صورت برابر بودن اجازه ی عبور صادر میشود و تعداد اشتباه ها پاک میشود

برای زمانی است که کاربر 3 بار رمز را اشتباه زده و تمام اینتراپت ها غیر فعال میشوند (به جز 3 برای حالت اضطراری)



با تشکر از توجه و همراهی شما