

```
.include "m32def.inc"
```

```
.ORG 0
```

```
RJMP START
```

```
.ORG OC1Aaddr
```

```
RJMP ROUTINE
```

```
.ORG OC0addr
```

```
RJMP ROUTINE2
```

```
; initialize stack pointers
```

```
START: LDI R16, HIGH(RAMEND)
```

```
OUT SPH, R16
```

```
LDI R16, LOW(RAMEND)
```

```
OUT SPL, R16
```

} interrupt

در ابتدا لیبل مربوط به هر یک از دو interrupt مورد نیاز برای تایمر ۰ و تایمر ۱ را به routine و routine2 جامب زدم و با جامپ کردن به START در ابتدای برنامه از اجرا شدن آنها در همان ابتدا اجتناب کردم.

```
; initialize ports
```

```
LDI R16, 0xFF
```

```
OUT DDRB, R16 ; PORTB as output
```

```
OUT DDRC, R16 ; PORTC as output
```

```
OUT DDRD, R16 ; PORTD as output
```

```
OUT PORTA, R16 ; Enabling Pullup Resistor on PORTA
```

```
LDI R16, 0x00
```

```
OUT DDRA, R16 ; PORTA as input
```

سپس پورت A را به عنوان ورودی (برای دیپسویچ) و پورت های B و C و D را به عنوان خروجی تنظیم کردم.

```
; initialize timers
```

```
; timer0  
LDI R16, 0x0C ; 0 00 1 100 (CTC clk/256)
```

```
OUT TCCR0, R16
```

```
LDI R16, 0x00 ; 0
```

```
OUT TCNT0, R16
```

```
LDI R16, 0x63
```

```
OUT OCR0, R16
```

```
; timer1
```

```
LDI R16, 0x00 ; 0
```

```
OUT TCCR1A, R16
```

```
LDI R16, 0x0C ; CTC clk/256
```

```
OUT TCCR1B, R16
```

```
; 7A12 = 31250 in decimal
```

```
LDI R16, 0x7A
```

```
OUT OCR1AH, R16
```

```
LDI R16, 0x12
```

```
OUT OCR1AL, R16
```

```
LDI R16, 0x12
```

```
OUT TIMSK, R16 ; OCIE0 and OCIE1A enabled
```

```
SEI
```

در مرحله ی بعدی رجیسترهای مربوط به تایمر ۰ و تایمر ۱ را مطابق آنچه در توضیح آزمایش ۳ آمده بود تنظیم کردم و در نهایت با SEI اینترپت ها را فعال کردم.

```
MAIN_LOOP:
```

```
; send LEDs output to PORT B
```

```
OUT PORTB, R24
```

```
RJMP DISP
```

در واقع در برنامه ی من CPU همواره در حال اجرا کردن حلقه ی روبرو است و R24 را روی LED ها نمایش می دهد و R20 تا R23 را روی 7segment ها نمایش می دهد که کنترل مقدارهای درست این رجیسترها را interrupt های مربوط به تایمر ۰ و تایمر ۱ برعهده دارند.

```
ROUTINE:
```

```
LSL R24
```

```
CPI R24, 0
```

```
BRNE CONT
```

```
LDI R24, 1
```

```
CONT: IN R16, PINA ; read portA digit0
```

```
MOV R17, R16 ; store as R17
```

```
ANDI R17, 0b00000001 ; remove other digits
```

```
CPI R17, 0 ; detect mode
```

```
BRNE DECREMENT
```

```
RJMP INCREMENT
```

```
CONT3: RETI
```

```
ROUTINE2:
```

```
INC R30
```

```
CPI R30, 4
```

```
BRLO CONT2
```

```
LDI R30, 0
```

```
CONT2: RETI
```

در interrupt تایمر ۱ پس از گذشت ۱ ثانیه با توجه به مقدار دیپسویچ یا یکی به عدد 7segment اضافه می کنیم یا یکی از آن کم می کنیم.

در interrupt تایمر ۰ هر بار R30 یکی زیاد می شود که رقم بعدی نمایش داده شود و اگر ۴ شود دوباره صفر می شود (چون رقم ۴م نداریم)