-۱

الف)

با مشاهده $2^7 * 3$ میتوان اطمینان حاصل کرد تقریبا 3sec سپری شده است. چون فرکانس clock اصلی برابر $2Mhz=2^{21}$ تنظیم شده بنابراین فرکانس clock_I/O برابر 2Mhz میباشد. پس از عبور از Prescaler، همین فرکانس بر 64 تقسیم شده و برابر 2^{15} خواهد شد. پس clock period زمانسنج/شمارنده۰ برابر $1/2^{15}$ خواهد بود. بنابراین $1/2^7 = 2^8 / 2^{15}$ sec طول میکشد تا یک سرریز رخ دهد چون سرریز هرclock 2^8 یکبار رخ می دهد. حال $2^7 = 1/(1/2^7)$ سرریز معادل 1sec خواهد بود و مشابها سه برابر آن 3sec را به ما خواهد داد.

به دلیل محدودیت های دستورات مقایسه در ریزپردازنده Atmega16 از دو متغیر کمکی استفاده کرده ایم. متغیر cnt1 بعد از هر $2^5 = 32$ بار سرریز باعث می شود یکی به cnt2اضافه شود. حال هر $2^2 * 3 = 12$ بار اضافه شدن cnt2 نحوه روشن بودن LED1 و LED2 عوض میشود که همان ۳ ثانیه ای است که به دنبالش بودیم. کد دقیق تر مطلب گفته شده در این خطوط به چشم میخورد:

```
ovf:
inc cnt1
;cp cnt1, cmp
;brne DO_NOT_SET_cnt2
cpi cnt1, 32
brne DO_NOT_SET_cnt2
;modify cnt2 if cnt = 128
inc cnt2
ldi cnt1, 0
DO_NOT_SET_cnt2:

cpi cnt2, 13
brlt TURN_ON_PLAN_A
rjmp TURN_ON_PLAN_B


TURN_ON_PLAN_A:
sbi portd, pd5;turn on led1
cbi portd, pd4;turn off led2
rjmp AFTER

TURN_ON_PLAN_B:
cbi portd, pd5;turn off led1
sbi portd, pd4;turn on led2
cpi cnt2, 24
brne AFTER
clr cnt2
AFTER:
reti
```

ب)

مشابه الف) میباشد با این تفاوت که اولا بیت های WGM و COM باید به حالت درستی تنظیم شوند که به مد CTC برویم، آن هم در حالتی که با رخداد های متوالی compare match خروجی OC0، toggle کند. در ثانی دیگر به روتین وقفه نیاز نیست. در اینجا prescaler را روی حالت تقسیم بر $2^{10} = 1024$ و مقدار رجیستر OCR0 را برابر 127 قرار داده ایم. بنا به رابطه فرکانس در حالت CTC خواهیم داشت:

$$f_{OC0} = f_{I/O}/2.N.(1 + OCR0) = 2^{20}/2.2^{10}.(1 + 127) = 2$$

۲- الف) در این سئوال و همچنین سئوال بعد از حالت inverting استفاده شده است. دلیل این کار، آنست که خواسته ایم در ابتدا که هیچ سوئیچی فشار داده نشده است، duty cycle برای موتور DC حداقل مقدار خود باشد. بدین منظور در روتین reset مقدارOCR را برابر 255=MAX قرار داده ایم که دائما مقدار 0 به پایه OC0 منتقل شود. البته باز هم در این حالت موتور، چرخش اندکی خواهد داشت و با سرعت بسیار کمی خواهد چرخید. با توجه به این که در مود PWM حالت Inverting، یک شدن OC0 بعد از compare match رخ میدهد پس مقدار دهی OCR0 تا حدودی با حالت های قبل فرق دارد. در تکه کد زیر این فرق مشهود است. صرفا کافی است به جای 100 از 100 - 255 استفاده کنیم تا مطمئن شویم در 100 کلاک خروجی 1 داده می شود و در نتیجه duty cycle برابر 100/256 خواهد بود:

```
sw1_pressed:
ldi r16, 255 - 120
out ocr0, r16
ret

sw2_pressed:
ldi r16, 255-240
out ocr0, r16
ret
```

ب)دقیقا مثل حالت الف) میباشد و تفاوت صرفا در تنظیم بیت های WGM برای تنظیم حالت مربوطه است.