

Project Report (گزارش پروژه)

Ali Naghiloo

40010093



دانشگاه صنعتی خواجه نصیرالدین طوسی

دانشکده مهندسی برق

1402/10/19

—

Digital Systems 2

—

Dr. Darmani

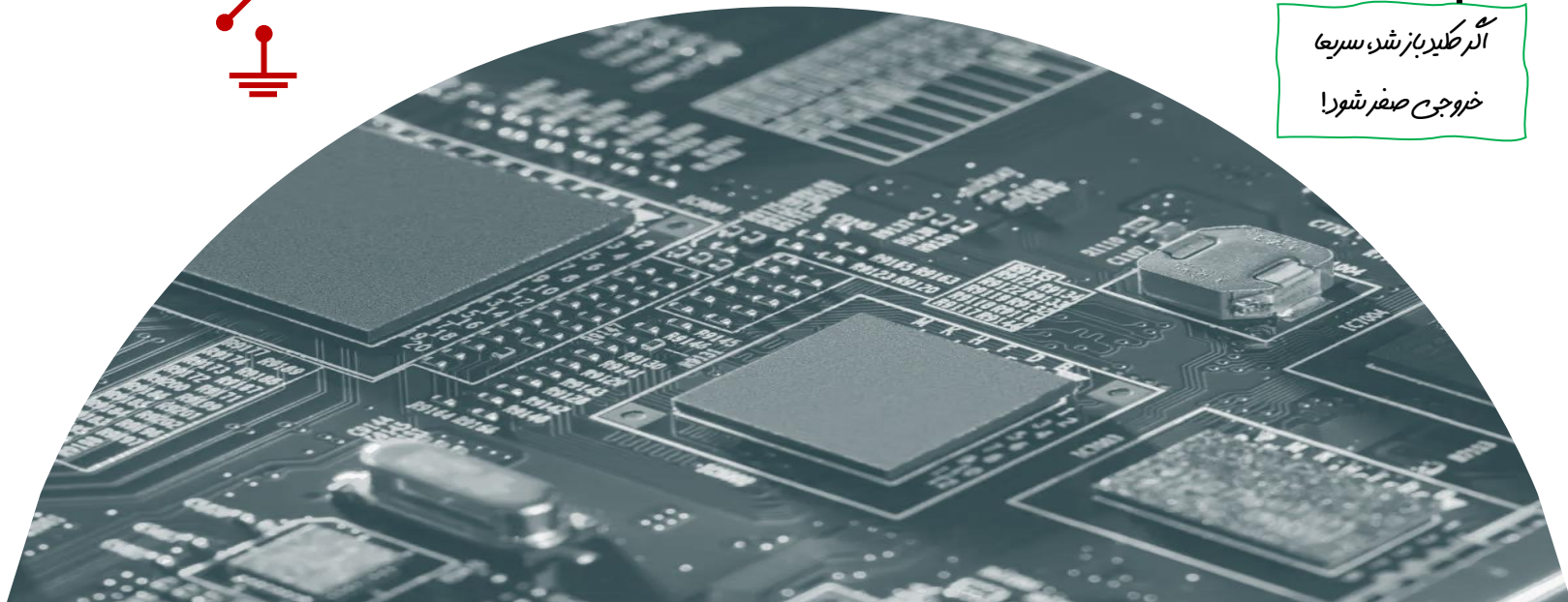
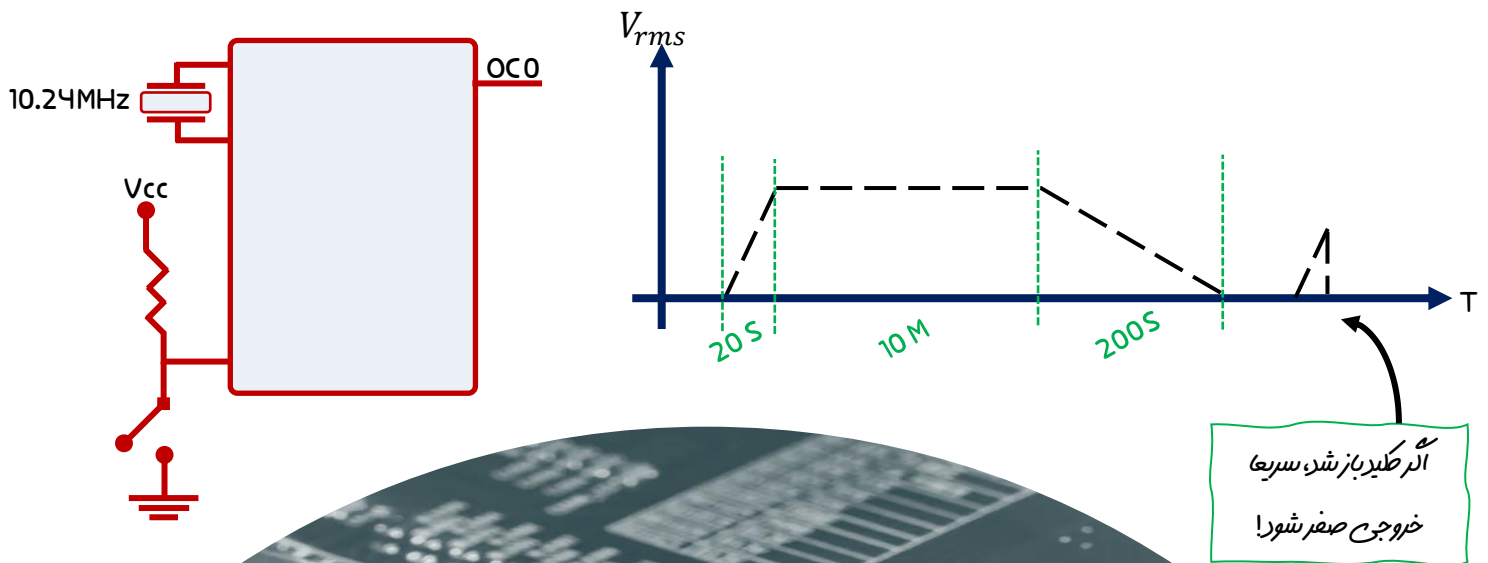
بهره

نکات قبل شروع گزارش پروژه:

لطفاً حتماً به کامنت‌گذاری‌های کلاه لای کرد، توضیحات این فایل و شماتیک‌ها و تصاویر توجه کنید زیرا اولاً زحمات فراوانی برایشان کشیده شده است، ثانیاً گاهی در بعضی قسمت‌های کد، خلاقیت‌هایی استفاده شده که ممکن است در آن‌ها به تنهایی و فقط با توجه به خود کد، سخت و گاهی غیر ممکن باشد!

نکات سوال پروژه:

وقتی کلید بسته شود در هر ثانیه، ۵٪ به مونتر خروجی اضافه شود. وقتی به حداکثر رسید، ۱۰ دقیقه ثابت بماند. بعد از هر ۱۰ ثانیه، ۵٪ کم شود تا به صفر برسد.



```

;=====
; DEFINITIONS
;=====

.DEF VAR_OCR = R20
.DEF SECOND = R22
.DEF MINUTE = R23
.DEF TEMP = R24
.DEF TEN = R25
.DEF NUM = R18

;=====
; RESET and INTERRUPT VECTORS
;=====

.INCLUDE "M64DEF.INC"

.ORG 0X0000
    JMP MAIN

.ORG 0X0014
    JMP TIMER2_OVF_ISR

;=====
; MAIN CODE SEGMENT
;=====

.ORG 0X0050
MAIN:
    LDI R16,HIGH(RAMEND)
    OUT SPH,R16
    LDI R16,LOW(RAMEND)
    OUT SPL,R16

    CBI DDRA,4      ;MAKE PORTA.4 AS AN INPUT(<=:0)
    SBI DDRB,4      ;MAKE PORTB.4 AS AN OUTPUT(==>:1)
    CBI PORTB,4     ;INITIAL STATE OF PORTB.4 WILL BE "0" VOLT FOR SAFTEY AT STARTING !

; f_tcmt0 = (10.24M/1) = 10.024 MHz
; T_tcmt0 ~ 25 us (I CHOSD PRESCALER=1 BECAUSE of UPDATING OCR0 FROM "OCR0 BUFFER" VERY VERY QUICKLY !)

    LDI R16,0
    OUT TCNT0,R16
    CLR VAR_OCR      ;AT FIRST MOMENT, OCR0 VALUE SHOULD BE 0 WHEN WE CLOSE SWICTH!
    OUT OCR0,VAR_OCR
    LDI R16,0X49      ;FAST PWM {OC0 DISCONNECTED} & n=1 (0100 1001)
    OUT TCCR0,R16
    LDI R16,0X40      ;JUST OVF2. IS ENABLE! (0100 0000)
    OUT TIMSK,R16

; f_tcmt2 = (10.24M/1024) = 10 KHz , T_tcmt2 = 100 us

```

```

LDI R16,6          ;(256-6)*100us => EVERY 25ms OVF2. INTRUPTS!
OUT TCNT2,R16
LDI R16,0X07       ;NORMAL MODE & n=1024 (0000 0111)
OUT TCCR2,R16

CLR R21            ;;
CLR SECOND        ;;    MEASURING TIME
CLR MINUTE        ;;
CLR TEMP
CLR TEN

;THE FIRST INTEGER MULTIPLE OF 12.75 IS NUMBER 51. (51 == 12.75 * 4)
;                (12.75 IS 5% OF 255)
LDI NUM,51
SEI

LOOP:
SBIS PINA,4
CALL CLOSE
SBIC PINA,4
CALL OPEN
JMP LOOP

OPEN:
LDI R16,0X49
OUT TCCR0,R16      ;OC0 DISCONNECTED(NORMAL I/O PORT)! (0100 1001)
CBI PORTB,4        ;HOWEVER PORTB.4 HAS BEEN CLEARED (AT INITIALIZING SECTION)!

CLR VAR_OCR
OUT OCR0,VAR_OCR
CLR R21            ;;{{
CLR SECOND        ;;    MEASURING TIME
CLR MINUTE        ;;
CLR TEMP          ;;
CLR TEN
RET

CLOSE:
LDI R16,0X69
OUT TCCR0,R16      ;OC0 CONNECTED & NON-INVERTING! (0110 1001)
OUT OCR0,VAR_OCR
RET

;=====
; INTERRUPT ISRs
;=====

TIMER2_OVF_ISR:    ;INTRUPTS EVERY 25ms
LDI R17,6
OUT TCNT2,R17

```

```

        CALL EV_25ms
        RETI
;=====
EV_25ms:
    INC R21
    CPI R21,40
    BRNE END_EV_25ms
    CLR R21
    CALL EV_1s
END_EV_25ms: RET
;=====
EV_1s:
    INC SECOND
    CPI SECOND,60
    BRNE ENDING
    CLR SECOND
    CALL EV_1m
ENDING: CALL OCR_CHANGER
    RET
;=====
EV_1m:
    INC MINUTE
    CPI MINUTE,60
    BRNE END_EV_1m
    CLR MINUTE
END_EV_1m: RET
;=====
; SUB PROGRAMS:
;=====

OCR_CHANGER:

INCREASE:
    CPI MINUTE,0
    BRNE STAY_HIGH
    CPI SECOND,21
    BRSH STAY_HIGH

    MUL NUM,SECOND
    LSR R1
    ROR R0
    LSR R1
    ROR R0           ;DEVIDED TO 4

    MOV VAR_OCR,R0
    JMP END_OCR_CHANGER

STAY_HIGH:
    CPI MINUTE,10
    BRLO END_OCR_CHANGER
    BRNE DECREASE    ;WHEN WE ARE IN THIS LINE OF CODE, MINUTE IS DEFINITELY "10 OR HIGHER" !
    CPI SECOND,21

```

BRLO END_OCR_CHANGER

DECREASE:

CPI MINUTE,13

BRLO PROG

BRNE STAY_LOW ;WHEN WE ARE IN THIS LINE OF CODE, MINUTE IS DEFINITELY "13 OR HIGHER" !

CPI SECOND,41

BRLO PROG

STAY_LOW:

LDI MINUTE,15 ;FREEZING TIME TO DO NOTHING AFTER 00:13:40 (for example I freeze it on a quarter)!

END_OCR_CHANGER: RET

PROG:

INC TEMP

CPI TEMP,10

BRNE END_OCR_CHANGER

CLR TEMP

INC TEN ;EVERY 10s

MUL NUM,TEN

LSR R1

ROR R0

LSR R1

ROR R0 ;DEVIDED TO 4

COM R0

MOV VAR_OCR,R0

JMP END_OCR_CHANGER

فلسفه خلاقیت در استفاده از عدد ۵۱ در رجیستر NUM یا همان R17 چیست؟؟؟

برای مثال در ۲۰ ثانیه ابتدایی، در واقع ۵٪ عدد ۲۵۵ میشود ۱۲.۷ که ما آنرا خواهیم آن را با عدد نزدیکش یعنی ۱۳ (یا ۱۲) تقریب بزنیم، واضح هست که در مرحله اول OCR ما ۰.۲۵ (یا ۰.۷۵) خط خواهد داشت که خیلی عدد بزرگی نیست و فعلا خوب هست ولی مثلاً اگر هر ثانیه OCR را با عدد ۱۳ جمع کنیم تا OCR به اصطلاح ۵٪ رشد کند و OCR جدید حاصل شود، خطاها روی هم انباشته و تلنبار شده و در نهایت مراحل آخر خیلی خطا خواهیم داشت، ضمن اینکه OCR یا به ۲۵۵ نخواهد رسید یا آن را رد می کنند که این بدان معنی است که هرگز ولتاژ موثر ۱۰۰ درصد خود را نخواهد دید که این برخلاف خواسته سوال هست!

مشابه همین قضیه ی انباشته شدن خطاها، در ۲۰ ثانیه آخر که در حال کاهش ولتاژ موثر هستیم نیز رخ خواهد داد

پس چاره چیست؟؟

صفحه بعد!

من آمدیم و اولین مضرب صحیح عدد مذکور یعنی ۱۲.۷۵ (همان ۵ درصد عدد ۲۵۵) را که عدد ۵۱ (همان ۴*۱۲.۷۵) را پیدا کردیم و در هر مرحله بجای مکانیزم ۱۳ × مرحله = OCR که خطای زیادی خواهد داشت، از زیر استفاده میکنیم:

$$OCR = \frac{51 \times \text{مرحله}}{4}$$

12.75

اینجا عدد ۱۳ تقریب عدد ۱۲.۷۵ هست و با ضرب در عددهای بزرگ در مراحل بالاتر، خطاهای زیادی تلنبار میشوند.

در هر مرحله به طور جداگانه ای فقط کمی خطا دارد که آن هم مربوط به LSR و ROR برای تقسیم بر ۴ هست و خطاهای روری هم انباشته نمیشود تا خطای بزرگی شود

