نام و نام خانوادگی: یزدان بیات

شماره دانشجویی: 40116453

گزارش کار پروژه 3 درس سیستم دیجیتال 2

ابتدا اینیشالایز های اولیه رو انجام میدهیم:

```
.include "C:\Users\Yazdan\Desktop\project\m64def.inc"
.ORG 0X0000
JMP MAIN
.ORG 0X0020
JMP EXT_TIMER0
.ORG 0X0014
JMP EXT_TIMER2
.ORG 0X0002
JMP EXT_INT0
.ORG 0X0004
JMP EXT_INT1
.ORG 0X0006
JMP EXT_INT2
.ORG 0X0008
JMP EXT_INT3

.ORG 0X0050
```

```
MAIN:
        LDI R16, low(RAMEND)
        OUT SPL, R16
        LDI R16, high(RAMEND)
        OUT SPH, R16

        LDI R16, 0X01
        OUT TIMSK, R16
        LDI R16, 0X07
        OUT TCCR0, R16
        LDI R16, 156
        OUT TCNT0, R16
        OUT TCNT2, R16
        LDI R16, 0X05
        OUT TCCR2, R16

        LDI R16, 0X0AA
        STS EICRA, R16
        LDI R16, 0X0F
        OUT EIMSK, R16

        LDI R16, 0X00
        OUT DDRB, R16
        LDI R16, 0XFF
        OUT DDRC, R16
        OUT DDRA, R16
        OUT DDRE, R16
        STS DDRG, R16

        LDI R16, 0X00
        MOV R20, R16
        MOV R21, R16
        MOV R22, R16
        MOV R23, R16
        MOV R31, R16
        LDI R30, 10



        SEI
```

در این اینیشالایز ها ابتدا ادرس اینتراپت ها و سپس تایمر را ست میکنیم. سپس در main بعد از مشخص کردن stack pointer به سراغ پایین رونده کردن وقفه های 0 تا 3 میکنیم. همچنین فرکانس cpu را 4.096Mhz فرض میکنیم و مقدار prescaler را 1024 قرار میدهیم که باعث میشود کانتر تایمر 0 و 2 که مورد استفاده قرار گرفتند را از 156 تا 256 شمارش کند و وقفه بزند.

** برای PORT G نمیتوان از دستور OUT استفاده کرد چون از ادرس 3F خارج است و مجبوریم از STS استفاده کنیم.

```
EXT_TIMER0:
        LDI R16, 156
        OUT TCNT0, R16

        CPI R31, 0
        BRNE MARHALE2
        JMP MARHALE_1
```

سپس وارد تایمر صفر میشویم که رجیستر R31 به عنوان فلگ برای ما مشخص میکند که ایا از حالت اولیه چراغ ها – سبز بودن مسیر شمال و جنوب و قرمز بودن مسیر شرق و غرب – خارج شده ایم یا خیر. طبق مقدار R31 وارد مرحله 1 یا 2 میشویم که در ادامه توضیح داده شده.

```
MARHALE_1:
        LDI R16, 0X01
        OUT PORTC, R16
        OUT PORTA, R16
        SBI PORTE, 2
        LDI R16, 0X04
        STS PORTG, R16
```

در marhale 1 حالت اولیه چراغ های خود را داریم که مسیر شمال و جنوب را سبز میکنیم و شرق و غرب را قرمز. پس باید بیت صفر پورت های c و A یک شود و بیت دوم پورت های E و G یک شود.

```
MARHALE2:
        INC R20
        CPI R20, 40
        BRNE END1
        CLR R20
        CALL COUNTING_1S
```

اما مرحله 2 شروع به شمارش زمان میکند تا به یک ثانیه برسد. پس از رسیدن به یک ثانیه تابع COUNTING_1S کال میشود.

```
COUNTING_1S:
        INC R21
        CPI R31, 0
        BREQ NEXT1
        CPI R31, 1
        BREQ NEXT2
        CPI R31, 2
        BREQ NEXT1
        CPI R31, 3
        BREQ NEXT2
        JMP END2
```

در این حالت به مقدار R21 به منظور گذر یک ثانیه یکی اضافه میکنیم. حال طبق مقدار فلگ که 4 حالت دارد انتخاب میکنیم مقصد بعدی ما کجاست. فلگ R31 در واقع یکی از چهار حالت زیر را برای ما مشخص میکند که رفتار برنامه متفاوت خواهد بود:

چرخه عادی:

۱. چراغ‌ها به صورت زیر تغییر می‌کنند:

- مرحله ۱: مسیر شمالی-جنوبی سبز، شرقی-غربی قرمز.
- مرحله ۲: مسیر شمالی-جنوبی زرد، شرقی-غربی قرمز.
- مرحله ۳: مسیر شمالی-جنوبی قرمز، شرقی-غربی سبز.
- مرحله ٤: مسیر شمالی-جنوبی قرمز، شرقی-غربی زرد.

حال به NEXT ها میپردازیم:

```
NEXT1:
        CP R21, R30
        BRNE END2
        CLR R21
        CALL COUNTING_10S

NEXT2:
        CPI R21,2
        BRNE END2
        CLR R21
        CALL COUNTING_2S
```

در NEXT1 مقدار 10 را به صورت پیش فرض قرار دادیم. اگر وقفه 2 فشرده شود مقدار R30، 2 ثانیه کم میشود و اگر وقفه 3 فشرده شود، 2 ثانیه زیاد میشود تا مقدار زمان سبز بودن چراغ تغییر کند.

در NEXT 2 هم 2 ثانیه شمرده میشود که برای چراغ زرد میباشد و زمان این چراغ ثابت است.

حال که میدانیم در کدام مرحله از زمانبندی چراغ ها هستیم، اگر زمان چراغ سبز به 10 و زمان زرد به 2 برسد باید حالت چراغ ها عوض شود. عوض شدن ان به صورت زیر میشود:

```
MARHALE1_BE_MARHALE2:
        CBI PORTC, 0
        CBI PORTA, 0
        SBI PORTC, 1
        SBI PORTA, 1
        INC R31
```

```
MARHALE2_BE_MARHALE3:
        CBI PORTC, 1
        CBI PORTA, 1
        SBI PORTC, 2
        SBI PORTA, 2
        LDI R16, 0X00
        STS PORTG, R16
        ;CBI PORTG, 2
        CBI PORTE, 2
        LDI R16, 0X01
        STS PORTG, R16
        ;SBI PORTG, 0
        SBI PORTE, 0
        INC R31

MARHALE3_BE_MARHALE4:
        LDI R16, 0X00
        STS PORTG, R16
        ;CBI PORTG, 0
        CBI PORTE, 0
        LDI R16, 0X02
        STS PORTG, R16
        ;SBI PORTG, 1
        SBI PORTE, 1
        INC R31
        JMP END3
```

```
MARHALE4_BE_MARHALE1:
        LDI R16, 0X00
        STS PORTG, R16
        ;CBI PORTG, 1
        CBI PORTE, 1
        LDI R16, 0X04
        STS PORTG, R16
        ;SBI PORTG,2
        SBI PORTE, 2
        CBI PORTC, 2
        CBI PORTA, 2
        SBI PORTC, 0
        SBI PORTA, 0
        CLR R31
        JMP END4
```

کاری که در هرکدام از این کد های انجام میشود این است که چراغ مرحله قبل خاموش میشود و چراغ مرحله بعدی روشن میشود.

برای مثال مرحله 1 چراغ شمال و جنوب سبز است و شرق و غرب قرمز. حال وقتی به ثانیه 10 برسیم، چراغ شمال و جنوب به مدت 2 ثانیه زرد میشود و شرق و غرب همچنان قرمز میمانند.

بقیه چراغ ها نیز به همین روند انجام میشود. در اخر هر مرحله فلگ نیز تغییر میکند.

تا اینجای کار مدار اصلی چراغ های راهنمایی را طراحی کردیم. اما باید کلید ریست، رد شدن امبولانس و زیاد و کم شدن مدت زمان چراغ سبز را نیز طراحی کنیم.

ابتدا به وقفه 2 و 3 میپردازیم که با فشردن انها تایم تغییر میکند.

```
EXT_INT2:
        DEC R30
        DEC R30
        RETI

EXT_INT3:
        INC R30
        INC R30
        RETI
```

پیش تر گفتیم که مقدار 10 را به صورت پیشفرض در R30 قرار دادیم. اگر اینتراپت 2 فشرده شود، 2 تا از مقدار R30 کاهش می یابد. همچنین اگر اینتراپت 3 فشرده شود، 2 تا به مقدار R30 اضافه میشود که حکم کاهش یا افزایش 2 ثانیه ای برای مقدار زمان سبز داریم که متعاقبا زمان قرمز نیز کاهش یا افزایش میابد.

```
EXT_INT1:
        CLR R16
        OUT PORTA, R16
        OUT PORTC, R16
        OUT PORTE, R16
        STS PORTF, R16

        LDI R30, 10
        CLR R20
        CLR R21

        CLR R31
        LDI R16, 0X01
        OUT TIMSK, R16
        RETI
```

در اینتراپت 1 باید سیستم ریست شود. پس همه چراغ ها را خاموش میکنیم و زمان به همراه فلگ ریست میشود.

حال به عبور امبولانس میپردازیم. اگر اینتراپت 0 فشرده شود، باید همه چراغ ها قرمز شوند و تنها مسیری که امبولانس در ان است سبز باشد.

برای اینکار برای سادگی در اینتراپت 0 تایمر 0 را خاموش میکنیم و تایمر 2 روشن میشود:

```
EXT_INT0:
        LDI R16, 0X40
        OUT TIMSK, R16
        RETI
```

حال وارد تایمر 2 میشویم:

```
EXT_TIMER2:
        LDI R16, 156
        OUT TCNT2, R16

        IN R17, PINB
        CPI R17, 14
        BREQ SHOMAL_SABZ
        CPI R17, 13
        BREQ JONOUB_SABZ
        CPI R17, 11
        BREQ SHARGH_SABZ
        CPI R17, 7
        BREQ GHARB_SABZ;???
        JMP END5
```

مقدار پورت B را میگیریم که برای ما مسیر امبولانس را مشخص میکند. برفرض اگر این پورت به صورت 1110 باشد یعنی از مسیر شمال،1101 از مسیر جنوب،1011 از مسیر غرب و اگر 0111 باشد یعنی از مسیر شرق عبور میکند.

```
SHOMAL_SABZ:
        CLR R16
        OUT PORTC, R16
        OUT PORTA, R16
        OUT PORTE, R16
        STS PORTG, R16

        LDI R16, 0X01
        OUT PORTC, R16
        CBI PORTA, 0
        SBI PORTA, 2
        SBI PORTE, 2
        LDI R16, 0X04
        STS PORTG, R16
        JMP WAIT_10_SECOND


JONOUB_SABZ:
        CLR R16
        OUT PORTC, R16
        OUT PORTA, R16
        OUT PORTE, R16
        STS PORTG, R16

        LDI R16, 0X04
        OUT PORTC, R16
        SBI PORTA, 0
        SBI PORTE, 2
        LDI R16, 0X04
        STS PORTG, R16
        JMP WAIT_10_SECOND
```

```
SHARGH_SABZ:
        CLR R16
        OUT PORTC, R16
        OUT PORTA, R16
        OUT PORTE, R16
        STS PORTG, R16

        CBI PORTC, 0
        SBI PORTC, 2
        CBI PORTA, 0
        SBI PORTA, 2
        SBI PORTE, 2
        LDI R16, 0X01
        STS PORTG, R16
        JMP WAIT_10_SECOND
```

```
GHARB_SABZ:
        CLR R16
        OUT PORTC, R16
        OUT PORTA, R16
        OUT PORTE, R16
        STS PORTG, R16

        CBI PORTC, 0
        SBI PORTC, 2
        CBI PORTA, 0
        SBI PORTA, 2
        CBI PORTE, 2
        SBI PORTE, 0
        LDI R16, 0X04
        STS PORTG, R16
        ;SBI PORTG, 2
        JMP WAIT_10_SECOND;
```

همانطور که مشاهده میکنید روند همه حالت ها یکسان هستند. یعنی طبق اطلاعات دریافتی از پورت B یکی از این حالات اجرا میشود که بیت دوم همه مسیر ها روشن میشود الا مسیری که امبولانس در ان قرار دارد.

پس از انجام این کار، تابع WAIT_10_SECOND کال میشود که 10 ثانیه DELAY ایجاد میکند تا امبولانس عبور کند.

```
WAIT_10_SECOND:
        INC R22
        CPI R22, 40
        BRNE END5
        CLR R22
        CALL TIMER2_1S

END5:
        RETI

TIMER2_1S:
        INC R23
        CPI R23, 10
        BRNE END6
        CLR R23
        CALL TIMER2_10S; ????? 1 ?????

END6:
        RET
```

نحوه شمارش این تایم دقیقا مانند تایمر 0 میباشد. و پس از رسیدن به 10 ثانیه RET میکند.

اما هنوز یک مرحله باقی مانده. وقتی امبولانس عبور کرد باید از تایمر 2 خارج شویم و وارد تایمر 0 شویم. به منظور این کار، مقدار TIMSK را تغییر میدهیم.

```
TIMER2_10S:
        CLR R16
        OUT PORTC, R16
        OUT PORTA, R16
        OUT PORTE, R16
        STS PORTG, R16

        LDI R16, 0X01
        OUT TIMSK, R16 ;????? 10 ?????
```

حال تایمر 0 روشن میشود و ادامه چراغ شکل میگیرد.

** نسخه کامل کد اسمبلی و پروتیوس اپلود شده اما برای راحتی و فهم بهتر روند کد، در این فایل نیز گذاشته شده:

```
.include "C:\Users\Yazdan\Desktop\project\m64def.inc"
.ORG 0X0000
JMP MAIN
.ORG 0X0020
JMP EXT_TIMER0
.ORG 0X0014
JMP EXT_TIMER2
.ORG 0X0002
JMP EXT_INT0
.ORG 0X0004
JMP EXT_INT1
.ORG 0X0006
JMP EXT_INT2
.ORG 0X0008
JMP EXT_INT3

.ORG 0X0050

        MAIN:
LDI R16, low(RAMEND)
OUT SPL, R16
```

```
LDI R16, high(RAMEND)
OUT SPH, R16

LDI R16, 0X01
OUT TIMSK, R16
LDI R16, 0X07
OUT TCCR0, R16
LDI R16, 156
OUT TCNT0, R16
OUT TCNT2, R16
LDI R16, 0X05
OUT TCCR2, R16

LDI R16, 0X0AA
STS EICRA, R16
LDI R16, 0X0F
OUT EIMSK, R16

LDI R16, 0X00
OUT DDRB, R16
LDI R16, 0XFF
OUT DDRC, R16
OUT DDRA, R16
OUT DDRE, R16
```

```
STS DDRG, R16

LDI R16, 0X00
MOV R20, R16
MOV R21, R16
MOV R22, R16
MOV R23, R16
MOV R31, R16
LDI R30, 10

SEI

LOOP:
JMP LOOP

EXT_TIMER0:
LDI R16, 156
OUT TCNT0, R16
```

```
CPI R31, 0
BRNE MARHALE2
JMP MARHALE_1

MARHALE_1:
LDI R16, 0X01
OUT PORTC, R16
OUT PORTA, R16
SBI PORTE, 2
LDI R16, 0X04
STS PORTG, R16


MARHALE2:
INC R20
CPI R20, 40
BRNE END1
CLR R20
CALL COUNTING_1S

END1:
RETI
```

```
COUNTING_1S:
INC R21
CPI R31, 0
BREQ NEXT1
CPI R31, 1
BREQ NEXT2
CPI R31, 2
BREQ NEXT1
CPI R31, 3
BREQ NEXT2
JMP END2

NEXT1:
CP R21, R30
BRNE END2
CLR R21
CALL COUNTING_10S

NEXT2:
CPI R21,2
BRNE END2
CLR R21
CALL COUNTING_2S
```

```
END2:
RET

COUNTING_10S:
CPI R31,0
BREQ MARHALE1_BE_MARHALE2
CPI R31,2
BREQ MARHALE3_BE_MARHALE4
JMP END3

MARHALE3_BE_MARHALE4:
LDI R16, 0X00
STS PORTG, R16
;CBI PORTG, 0
CBI PORTE, 0
LDI R16, 0X02
STS PORTG, R16
;SBI PORTG, 1
SBI PORTE, 1
INC R31
JMP END3
```

```
MARHALE1_BE_MARHALE2:
CBI PORTC, 0
CBI PORTA, 0
SBI PORTC, 1
SBI PORTA, 1
INC R31

END3:
RET

COUNTING_2S:
CPI R31, 1
BREQ MARHALE2_BE_MARHALE3
CPI R31, 3
BREQ MARHALE4_BE_MARHALE1
JMP END4

MARHALE4_BE_MARHALE1:
LDI R16, 0X00
STS PORTG, R16
;CBI PORTG, 1
CBI PORTE, 1
LDI R16, 0X04
STS PORTG, R16
```

```
;SBI PORTG,2
SBI PORTE, 2
CBI PORTC, 2
CBI PORTA, 2
SBI PORTC, 0
SBI PORTA, 0
CLR R31
JMP END4

MARHALE2_BE_MARHALE3:
CBI PORTC, 1
CBI PORTA, 1
SBI PORTC, 2
SBI PORTA, 2
LDI R16, 0X00
STS PORTG, R16
;CBI PORTG, 2
CBI PORTE, 2
LDI R16, 0X01
STS PORTG, R16
;SBI PORTG, 0
SBI PORTE, 0
INC R31
```

```
END4:
RET


EXT_INT2:
DEC R30
DEC R30
RETI

EXT_INT3:
INC R30
INC R30
RETI

EXT_INT1:
CLR R16
OUT PORTA, R16
OUT PORTC, R16
OUT PORTE, R16
STS PORTF, R16

LDI R30, 10
```

```
CLR R20
CLR R21

CLR R31
LDI R16, 0X01
OUT TIMSK, R16
RETI

EXT_INT0:
LDI R16, 0X40
OUT TIMSK, R16
RETI


EXT_TIMER2:
LDI R16, 156
OUT TCNT2, R16

IN R17, PINB
CPI R17, 14
BREQ SHOMAL_SABZ
CPI R17, 13
BREQ JONOUB_SABZ
CPI R17, 11
```

```
BREQ SHARGH_SABZ
CPI R17, 7
BREQ GHARB_SABZ
JMP END5

SHOMAL_SABZ:
CLR R16
OUT PORTC, R16
OUT PORTA, R16
OUT PORTE, R16
STS PORTG, R16

LDI R16, 0X01
OUT PORTC, R16
CBI PORTA, 0
SBI PORTA, 2
SBI PORTE, 2
LDI R16, 0X04
STS PORTG, R16
JMP WAIT_10_SECOND

JONOUB_SABZ:
CLR R16
OUT PORTC, R16
```

```
OUT PORTA, R16
OUT PORTE, R16
STS PORTG, R16

LDI R16, 0X04
OUT PORTC, R16
SBI PORTA, 0
SBI PORTE, 2
LDI R16, 0X04
STS PORTG, R16
JMP WAIT_10_SECOND

SHARGH_SABZ:
CLR R16
OUT PORTC, R16
OUT PORTA, R16
OUT PORTE, R16
STS PORTG, R16

    CBI PORTC, 0
SBI PORTC, 2
CBI PORTA, 0
SBI PORTA, 2
SBI PORTE, 2
```

```
LDI R16, 0X01
STS PORTG, R16
JMP WAIT_10_SECOND

GHARB_SABZ:
CLR R16
OUT PORTC, R16
OUT PORTA, R16
OUT PORTE, R16
STS PORTG, R16

CBI PORTC, 0
SBI PORTC, 2
CBI PORTA, 0
SBI PORTA, 2
CBI PORTE, 2
SBI PORTE, 0
LDI R16, 0X04
STS PORTG, R16
JMP WAIT_10_SECOND

WAIT_10_SECOND:
INC R22
CPI R22, 40
```

```
BRNE END5
CLR R22
CALL TIMER2_1S

END5:
RETI

TIMER2_1S:
INC R23
CPI R23, 10
BRNE END6
CLR R23
CALL TIMER2_10S

END6:
RET

TIMER2_10S:
CLR R16
OUT PORTC, R16
OUT PORTA, R16
OUT PORTE, R16
STS PORTG, R16
```

```
LDI R16, 0X01
OUT TIMSK, R16

END7:
RET
```