محمد پویا خرسندی تمرین ۴ ریز پردازنده - ۹۳۳۱۹۱۰

تمرین ۱ قسمت الف)

در این بخش ما میخواهیم با زدن کلیدsw1 وقفه خارجی INT1 فعال شود و LED1 روشن شود. برای فعال کردن وقفه خارجی INT1 باید در کد بخش زیر را بگذاریم تا با آمدن وقفه مورد نظر به آن ادرس برود و روتین وقفه را انجام دهد.

```
.org 0x004
jmp ISR1
```

در مرحله بعد ما رجیستر های مربوط به فعال سازی وقفه ها را مشخص میکنیم و چگونه وقفه را بگیریم را با استفاده از رجیستر MCUCR باید طبق کد زیر این کار را انجام دهیم.

```
LDI r16, (1<<INT1)
OUT GICR, r16

LDI r16, (1 << ISC11) | (0 << ISC10) ;  10:falling edge
OUT MCUCR, r16
```

INT1 را در GICR فعال میکنیم که یعنی وقفه را بپذیرد و لبه را به صورت falling edge دریافت کند. و برای اینکه LED را خاموش روشن کنیم یک رجیستر را مقدار یک را در آن بارگذاری میکنیم و در هر دفعه با دستور CPI مقایشه میکنیم اگر مساوی بود به label ، LED_on میرویم و اگر مساوی نبود به LED_Off میرویم که کد آن به صورت زیر است.

```
CPI r24,0x01
BREQ LED_ON
RCALL LED_OFF
```

سوال ۱ قسمت ب) و ج)

در این بخش ما با 7-segment کار میکنیم و در ابتدا باید رجیستر های مربوط به port ها و وقفه ها را setup کنیم.

```
ldi r16,0b11111111;
out ddrb,r16;
ldi r16,0b00000000;
out portb,r16;

ldi r16,0b00000000;
out ddrd,r16;
ldi r16,0b00000100;
out portd,r16;

LDI r16, (1<<INT0)
OUT GICR, r16
;MCUCR
LDI r16, (1 << ISC01) | (0 << ISC00) ;  10:falling edge
OUT MCUCR, r16

SEI
```

و سپس باید با امدن وقفه به روتین keyfind برود و عدد مورد نظر را نمایش دهد که روتین keyfind را به این گونه پیاده سازی کردیم که هر موقع کاربر کلید را فشار داد پس یکی از پین های pc0 تا pc3 یک شده و ستون مورد نظر مشخص میشود حال برای مشخص کردن سطر ما خروجی های pc4 تا pc7 را یکی یکی صفر میکنیم تا بفهمیم کدام کلید فشار داده شده است.

```
ldi r17,(1<<PC7)|(1<<PC6)|(1<<PC5)|(0<<PC4);
out portc,r17;
sbis pinc,0
ldi r18,0x3F
sbis pinc,1
ldi r18,0x06
sbis pinc,2
ldi r18,0x5B
sbis pinc,3
ldi r18,0x4F
```

سوال ۲)قسمت الف)

با توجه به روتین های داده شده برای ارتباط برقرار کردن با LCD ما hello world را به صورت زیر نوشتیم و در LCD نمایش دادیم .

```
start:
  ldi  temp, low(RAMEND)
  out  SPL, temp
  ldi  temp, high(RAMEND)
  out  SPH, temp
    rcall  LCD_init
  rcall  LCD_wait
  ldi  argument, 'H'
  rcall  LCD_putchar
  ldi  argument, 'E'
  rcall  LCD_putchar
  ldi  argument, 'L'
  rcall  LCD_putchar
  ldi  argument, 'L'
  rcall  LCD_putchar
  ldi  argument, 'O'
  rcall  LCD_putchar
  ldi  argument, ' '
  rcall  LCD_putchar
  ldi  argument, 'W'
  rcall  LCD_putchar
  ldi  argument, 'O'
  rcall  LCD_putchar
  ldi  argument, 'R'
  rcall  LCD_putchar
  ldi  argument, 'L'
  rcall  LCD_putchar
  ldi  argument, 'D'
  rcall  LCD_putchar
```

قسمت ب) حال در این قسمت ما ابتدا اسم و فامیلی خود را در حافظه با استفاده از LABEL مینویسیم و سپس ادرس LCDTABLE را در رجیستر Z میریزیم و ابتدا تعداد کاراکتر را از اولین ادرس میخوانیم و با افزایش Z کاراکتر ها را یکی یکی میخوانیم و در LCD نمایش میدهیم و هر وقت مساوی عدد تعداد کاراکتر شد خاتمه میدهیم. که کد این بخش به صورت زیر نوشته شده است.

```
LCDTABLE : .db 14,'P','o','u','y','a','K','h','o','r','s','a','n','d','i'
rjmp start

start:
  ldi  temp, low(RAMEND)
  out  SPL, temp
  ldi  temp, high(RAMEND)
  out  SPH, temp
  rcall  LCD_init
  rcall  LCD_wait
  rcall  LCD
  rjmp start


LCD :
    LDI ZH,high(LCDTABLE)
    LDI ZL,low(LCDTABLE)
    LPM r22,Z+
    LDI r23,0
    LOOP:
        CP r22,r23
        BREQ ENDPRINT
        LPM argument,Z+
        RCALL LCD_putchar
        RCALL LCD_delay
        INC r23
        RJMP LOOP;
ENDPRINT:
    RET
```

قسمت ج)

مانند سوال یک کد های مربوط به 7segment را وارد میکنیم و کد های مربوط به روتین keyfind را هم به همان شکل مینویسیم با این تفاوت که در اینجا عدد ۱ تا ۱۵ را در رجیستر R18 میگذاریم و هر با R19 را افزایش میدهیم برای انکه تشخیص دهیم کی خط ما به پایان میرسد و به خط بعدی برویم ،سپس در LCD نمایش میدهیم اگر عدد ما دو

رقمی بود دو واحد افزایش میدهیم و گرنه یک واحد. کد این بخش به صورت زیر نوشته شده است.

```
LCD:
    inc r19
    inc r19
    mov r16,r0;
    subi r16,10
    brpl twoDigit;
    mov r16,r0
    ldi r17,0x30;
    add r17,r16;
    call lcd_putchar
    jmp end;
    twoDigit:
    inc r19
    ldi r17,'1'
    call lcd_putchar
    mov r17,r0;
    subi r17,10
    ldi r18,0x30;
    add r17,r18;
    call lcd_putchar
    jmp end
end:
    ldi r17,',';
    call lcd_putchar;
    mov r20,r19
    subi r20,16
    brpl NEWLINE
    ret
NEWLINE:
    ldi r17,0xC0
    call lcd_command
    ldi r19,0
    ret
```