

**پیش گزارش و گزارش کار آزمایش هفتم**

سیاوش کاوسی 9231048

آرش تارافر 9131034

استاد: آقای حیدری

فروردین 1395

1. **پیش گزارش**
   1. **هدف آزمایش**

کار با وقفه خارجی، راه اندازی یک کیبورد ماتریسی، نمایش کاراکتر ها توسط نمایش دهنده 7 قطعه ای

* 1. **شرح آزمایش**

میخواهیم با نوشتن برنامه ای مقادیر کلید های فشرده شده در کیپد ماتریسی را بر روی یک نمایش دهنده 7 قطعه ای نمایش دهیم

برنامه به زبان اسمبلی avr:

.def A = r16

.def B = r17

.def ROW\_HOLDER = r18

.def COL\_HOLDER = r19

.def NUM\_HOLDER = r20

.org 0x00

reset:

jmp reset\_isr

.org 0x02

interrupt0:

jmp interrupt0\_isr

در برنامه روتین وقفه بازنشانی، خروجی و ورودی های پورت C را تنظیم و سپس وقفه INT0 را فعال کرده ایم

reset\_isr:

cli

ldi r16, LOW(RAMEND)

out SPL, r16

ldi r17, HIGH(RAMEND)

out SPH, r17

call column\_set

call pullup\_pd2

ldi A, (1 << INT0)

out GICR, A

ldi A, 0xFF

out DDRA, A

ldi A, 0x00

out PORTA, A

sei

jmp start

در برنامه روتین وقفه INT0، ابتدا محل کلید فشرده شده را پیدا میکنیم (با جارو کردن، یعنی همه پورت های خروجی high باشند و در یک بازه زمانی یکی از آنها را low می کنیم) و سپس با صدا زدن روتین key\_finder مقدار مورد نظر را یافته و بر روی نمایش دهنده 7 قطعه ای نمایش می دهیم

interrupt0\_isr:

cli

call column\_finder

call row\_set

nop

nop

nop

nop

call row\_finder

or ROW\_HOLDER, COL\_HOLDER

mov NUM\_HOLDER, ROW\_HOLDER

call key\_finder

sei

reti

start:

rjmp start

pullup\_pd2:

ldi A, (1 << PD2)

out PORTD, A

ret

column\_set:

ldi A, (1 << PC4) | (1 << PC5) | (1 << PC6) | (1 << PC7)

out DDRC, A

ldi A, (1 << PC0) | (1 << PC1) | (1 << PC2) | (1 << PC3)

out PORTC, A

ret

column\_finder:

in COL\_HOLDER, PINC

com COL\_HOLDER

andi COL\_HOLDER, 0x0F

ret

row\_set:

ldi A, (1 << PC0) | (1 << PC1) | (1 << PC2) | (1 << PC3)

out DDRC, A

ldi A, (1 << PC4) | (1 << PC5) | (1 << PC6) | (1 << PC7)

out PORTC, A

ret

row\_finder:

in ROW\_HOLDER, PINC

com ROW\_HOLDER

andi ROW\_HOLDER, 0xF0

ret

در روتین key\_finder ابتدا کلید زده شده را پیدا میکنیم و سپس آنرا بر روی 7 segment نمایش می دهیم

key\_finder:

zero:

cpi NUM\_HOLDER, 0x11

brne one

ldi A, 0x40

out PORTA, A

rjmp key\_finder\_end

one:

cpi NUM\_HOLDER, 0x12

brne two

ldi A, 0x79

out PORTA, A

rjmp key\_finder\_end

two:

cpi NUM\_HOLDER, 0x14

brne three

ldi A, 0x24

out PORTA, A

rjmp key\_finder\_end

three:

cpi NUM\_HOLDER, 0x18

brne four

ldi A, 0x30

out PORTA, A

rjmp key\_finder\_end

four:

cpi NUM\_HOLDER, 0x21

brne five

ldi A, 0x19

out PORTA, A

rjmp key\_finder\_end

.

.

.

key\_finder\_end:

call column\_set

call pullup\_pd2

ret

1. **گزارش کار**

در آزمایشگاه نیز باید برنامه بالا که به زبان اسمبلی نوشته شده را، به زبان شبه C! بنویسیم و بر روی بورد برنامه ریزی کنیم

برنامه به زبان C:

در این برنامه تابع key\_find با جارو کردن خروجی کلید (های) موردنظر را تشخیص می دهد و در تابع key\_find\_print آنرا بر روی نمایش دهنده 7 قطعه ای چاپ می کند

void key\_find()

{

PORTA.4 = 0;

key\_find\_print(4);

delay\_ms(5);

PORTA.4 = 1;

PORTA.5 = 0;

key\_find\_print(5);

delay\_ms(5);

PORTA.5 = 1;

PORTA.6 = 0;

key\_find\_print(6);

delay\_ms(5);

PORTA.6 = 1;

PORTA.7 = 0;

key\_find\_print(7);

delay\_ms(5);

PORTA.7 = 1;

clear\_portb();

}

void key\_find\_print(int row\_number)

{

switch(row\_number)

{

case 4:

if(!PINA.0)

PINB = 0xBF;

else if(!PINA.1)

PINB = 0x86;

else if(!PINA.2)

PINB = 0xDB;

else if(!PINA.3)

PINB = 0xCF;

break;

case 5:

if(!PINA.0)

PINB = 0xE6;

else if(!PINA.1)

PINB = 0xED;

else if(!PINA.2)

PINB = 0xFD;

else if(!PINA.3)

PINB = 0x87;

break;

case 6:

if(!PINA.0)

PINB = 0xFF;

else if(!PINA.1)

PINB = 0xE7;

else if(!PINA.2)

PINB = 0xF7;

else if(!PINA.3)

PINB = 0xFF;

break;

case 7:

if(!PINA.0)

PINB = 0xB9;

else if(!PINA.1)

PINB = 0xBF;

else if(!PINA.2)

PINB = 0xF9;

else if(!PINA.3)

PINB = 0xF1;

break;

default:

PINB = 0xFF;

}

}

void clear\_portb()

{

PORTB = 0xFF;

}