



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

پیش گزارش و گزارش کار آزمایش هفتم

سیاوش کاوسی ۹۲۳۱۰۴۸

آرش تارافر ۹۱۳۱۰۳۴

استاد: آقای حیدری

فروردین ۱۳۹۵

۱. پیش گزارش

۱,۱. هدف آزمایش

کار با وقفه خارجی، راه اندازی یک کیبورد ماتریسی، نمایش کاراکتر ها توسط نمایش دهنده ۷ قطعه ای

۱,۲. شرح آزمایش

میخواهیم با نوشتن برنامه ای مقادیر کلید های فشرده شده در کیپد ماتریسی را بر روی یک نمایش دهنده ۷ 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 ۷ 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
```

۲. گزارش کار

در آزمایشگاه نیز باید برنامه بالا که به زبان اسمبلی نوشته شده را، به زبان شبه C! بنویسیم و بر روی برد 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 ۷ 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;
}

```