۱- در نظر بگیرید که بخواهیم خروجی PWM به صورت تابعی از ورودی آنالوگ معادل $y = x(Vin)^3$ تولید شود که در آن :

- فرکانس PWM معادل ۱۲/۵kHz در نظر گرفته شود
- ورودی آنالوگ ۷۲۰ بار در ثانیه بروز شود
- Y نسبت زمان high به پریود PWM بوده (duty cycle) که مقدار آن بین صفر تا یک تغییر می‌کند
- Vin ولتاژ آنالوگ ورودی که بین صفر تا ۲/۵۶ ولت است
- x ضریبی که بر اساس داده‌های قبلی بدست می‌آید

هستند. فرضیات (مثلاً فرکانس پالس ساعت میکرو)، محاسبات، مجموعهٔ ارتباطها، تنظیمات اولیه (شامل آماده‌سازی تایمر و ADC) و نرم‌افزار مربوطه را مشخص نمایید. در هر بخش توضیح کافی نیز ارائه شود.

۲- در برنامهٔ صفحه‌کلید–نمایشگر که نمونهٔ آن در ذیل آمده است تغییرات را بگونه‌ای ایجاد کنید تا اگر کلید با کد ۱۱ زده شود، از آخرین مقدار وارد شده (در حال نمایش بر روی نمایشگرها) ۳ واحد کاسته شود (۳ مقدار دیگر در حال نمایش بر روی نمایشگرها هیچ تغییری نمی‌کنند). در مورد فرآیند مورد نظر و خطوطی که تغییر می‌دهید یا می‌افزایید توضیح کافی بدهید.

```
void main(void){
.
.
.
  while (1) {
    while((i=keyscan())==16)dispscan();
    instodisp(keyto7seg(i));
    while(keyscan()!=16)dispscan();
    }
 }

void dispscan(void){
    char digsel = 0xfe;
    int i=0 , j;
    for(i=0; i<4; i++){
        PORTC = digsel;
        PORTD = dispbuf[i];
        for(j=0; j<2000; j++);
        PORTD = 0xff;
        digsel=(digsel<<1)+1;
        }
    }

char keyscan(void){
    int keycode=0, i, j;
    char colsel = 0xfe, t;
    for(i=0; i<4; i++){
        PORTC = colsel;
        for(j=0; j<2; j++);
        t = PINC;
        for(j=0; j<4; j++){
          if (!(t & 0x10)) return (keycode);
          keycode++;
          t>>=1;
          }
        colsel=(colsel<<=1)+1;
        }
```

```
    return(keycode);
    }

char keyto7seg(char keycode){
    char table[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0x88,0x83,0xc6,0xa1,0x86,0x8e};
    return(table[keycode]);
    }

void instodisp(char segcode){
    int i;
    for(i=3; i>0; i--) dispbuf[i]=dispbuf[i-1];
    dispbuf[i]=segcode;}
    }
```

۳- در برنامهٔ ارسال–دریافت سریال آسنکرون (۷بیتی پیاده شده به روش نرم‌افزاری که کد آن ذیلاً آمده) تغییراتی ایجاد نمایید (اسمبلی) تا در بخش دریافت، تشخیص خطای پریتی (فرد) هم انجام شده و نتیجهٔ آن در بیت ۵ از پورت C قرار داده شود : مقدار آن در صورت وقوع خطای پریتی '0' و در غیر این صورت '1' گردد. در مورد فرآیند مورد نظر و خطوطی که تغییر می‌دهید یا می‌افزایید توضیح کافی بدهید.

```
#pragma warn-
#asm
  .equ   PORTA = $1b
  .equ   PINA = $19
  .equ   PORTB = $18
  .equ   PINB = $16
  .equ   PORTC = $15
  .equ   PINC = $13
  .equ   PORTD = $12
  .equ   PIND = $10
  .equ   DISPBUF = $800
  .equ   KCODETAB = $810

  .def   XREGLOW = r26
  .def   XREGHIGH = r27
  .def   YREGLOW = r28
  .def   YREGHIGH = r29
  .def   ZREGLOW = r30
 .def   ZREGHIGH = r31
#endasm

char receive7(void){
#asm
receive_7bit:
    ldi     r18,0
till_reset:
    sbic    PINC,5       //1-2
    cbi     PORTC,4      //1
    in      r17,PIND     //1
    andi    r17,1        //1
    brne    till_reset   //2-1
    call    hbit_delay   //4
    ldi     r16,7        //1
rnxt_bit:
    clc                  //1
    sbic    PIND,0       //1-2
    sec                  //1
    ror     r18          //1
    call    bit_delay    //4
    dec     r16          //1
```

```
    brne   rnxt_bit    //2
    clc
    ror    r18
    mov    r30,r18
#endasm
    }

//formula for delays (both hbit_delay and bit_delay
//       3*r17*(r16+1)+23

#asm
bit_delay:
    push   r16       //2
    push   r17       //2
    ldi    r17,1     //1
bdel2:
    ldi    r16,26    //1    //133
bdel1:
    dec    r16       //1
    brne   bdel1     //2 on jump - 1 for next instruction
                     // loop1 length = 3*r16-1
    dec    r17       //1
    brne   bdel2      //2 on jump - 1 for next instruction
                     // loop2 length = (4+loop1)*r17-1
    pop    r17       //2
    pop    r16       //2
    ret              //4
                     //total delay = 13+loop2
hbit_delay:
    push   r16       //2
    push   r17       //2
    ldi    r17,1     //1
hbdel2:
    ldi    r16,43    //1    //203
hbdel1:
    dec    r16       //1
    brne   hbdel1      //2 on jump - 1 for next instruction
                     // loop1 length = 3*r16-1
    dec    r17       //1
    brne   hbdel2      //2 on jump - 1 for next instruction
                     // loop2 length = (4+loop1)*r17-1
    pop    r17       //2
    pop    r16       //2
    ret              //4
                     //total delay = 13+loop2
#endasm
#pragma warn-
```

۴- یک مکانیزم اندازه‌گیری فاصله بر اساس اندازه‌گیری زمان رفت و برگشت سیگنال (ماوراء صوت) ناشی از انعکاس سیگنال ارسالی کار می‌کند :

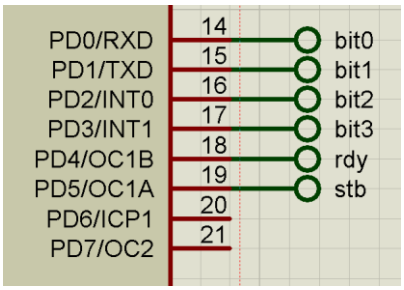
- بخش فرستندهٔ سیگنال دارای ورودی منطقی (’0‘ یا ’1‘ معادل خاموش یا روشن) است که به صورت پیش‌فرض بمدت ۱ms تحریک می‌شود
- بخش دریافت کنندهٔ سیگنال نیز دارای ورودی منطقی (’0‘ یا ’1‘ معادل خاموش یا روشن) است که پس از خاموش شدن فرستنده بکار می‌افتد و قرار است تا ۶۰ms فعال بماند
- فرستنده و دریافت کننده همزمان فعال نمی‌شوند
- خروجی دریافت کننده هنگام فعال بودن در صورت دریافت انعکاس ’1‘ می‌شود (در غیر این صورت ’0‘ است)

- سرعت حرکت سیگنال در محیط ۳۳۰m/s ثابت فرض می‌شود
- اندازه‌گیری با لبهٔ بالارونده سیگنال st که از بیرون به میکروکنترلر وارد می‌گردد شروع و انجام می‌شود

هدف اندازه‌گیری فاصله تا مانع (یا جسم روبرو) بر حسب سانتیمتر (عدد صحیح باینری) از ۳۰cm تا ۹۹۹cm و قرار دادن آن در متغیر صحیح با نام dist است. فرضیات (مثلاً فرکانس پالس ساعت میکرو)، محاسبات، مجموعهٔ ارتباطها، تنظیمات اولیه و نرم‌افزار مربوطه را مشخص نمایید. در هر بخش توضیح کافی نیز ارائه شود.

۵- نمونه‌ای از انتقال داده به صورت موازی بین دو واحد (دو میکرو در اینجا) در انتقال مقدار دریافت شده به صورت سریال به میکرو سوم در مثال پیاده‌سازی انتقال دادهٔ سریال به صورت نرم‌افزاری را مشاهده نموده‌اید (استفاده از سیگنالهای handshake). بخشی از برنامهٔ میکرو سوم و ارتباطهای مربوطه را ملاحظه می‌نمایید. اینک فرض کنید در میکرو دوم داده (۷بیتی) پس از دریافت در متغیر rdata (global تعریف شده) قرار داشته باشد. برای میکرو دوم، مجموعهٔ ارتباطهای پیشنهادی و تابعی که در صورت صدا زده شدن عمل تبادل داده با میکرو سوم را شروع و کامل می‌کند را بنویسید. توضیح کافی نیز ارائه شود.

```
while (1) {
    dispscan();
    if(PIND & 0x10){
      instodisp(keyto7seg(PIND & 0x0F));
      PORTD.5=1;
      }
    else{
      PORTD.5=0;
      }
  }
```



۶- در تابع bit_delay که در متن سؤال سوم آمده است اگر بخواهیم میزان تأخیر تا حدود ۵ میلیون پالس ساعت (۵ ثانیه برای clock سیستم ۱MHz) افزایش یابد، لازم خواهد بود حلقهٔ جدیدی به دو حلقهٔ موجود بیافزاییم. متن تابع bit_delay تغییر یافته (اسمبلی) و رابطهٔ میزان تأخیر بدست آمده بر حسب رجیسترهای مربوط (r16 و r17 در وضعیت موجود و رجیستر سومی که شما در نظر می‌گیرید از ابتدا تا انتهای تابع) را مشخص کنید. توضیح کافی نیز ارائه شود.

سؤال اول	سؤال دوم	سؤال سوم	سؤال چهارم	سؤال پنجم	سؤال ششم
۵	۴	۴	۴	۴	۳

موفق باشید

مجید شرکاء