

KOCAELİ ÜNİVERSİTESİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
BÖLÜMÜ



Analog Sensör ile Kalp Atış Hızını Ölçüp
Sonucu Bilgisayara Aktarma

PROJE ÖDEVİ

190208059 – Aslınur Serin

MİKROİŞLEMCİLER

MAYIS , 2022

İÇİNDEKİLER

ŞEKİL LİSTESİ	ii
1 PULSE SENSÖR	1
1.1 Pin Yapısı	
2 MSP430 ADC	2
2.1 MSP430 ADC Blog Diagram	
3 MSP430 HABERLEŞME	4
3.1 Kod Kütüphanesi	
3.2 Projenin Kodu	6
4 SİSTEMİN TASARIMI	6
4.1 Altium Şematik	4
4.2 PCB	
4 SİSTEMİN KODU	9
5 Kaynaklar	

ŞEKİL LİSTESİ

Şekil 1.1	Pulse Sensör	1
Şekil 2.1	MSP430 ADC Blog Diagram	2
Şekil 2.2	Çapraz Bağlantı	3
Şekil 4.1	Altium Şematik	4
Şekil 4.2	PCB Çizimi.....	5

MSP430 ile Kişi Sayma Sistemi

Elifnur Telli

Elektronik ve Haberleşme Mühendisliği Bölümü
Mikroişlemciler Projesi

Bu projemizde MSP430 kullanarak Çift Yönlü Ziyaretçi Sayacı yapılmıştır. Bu proje, bir engel algıladığında kesintiyi algılayan bir çift Kızılötesi Sensöre dayanmaktadır . Kızılötesi sensör çifti , ziyaretçiyi her iki yönden, yani giren ziyaretçi sayısı ve çıkan ziyaretçi sayısı gibi algılayabilir .

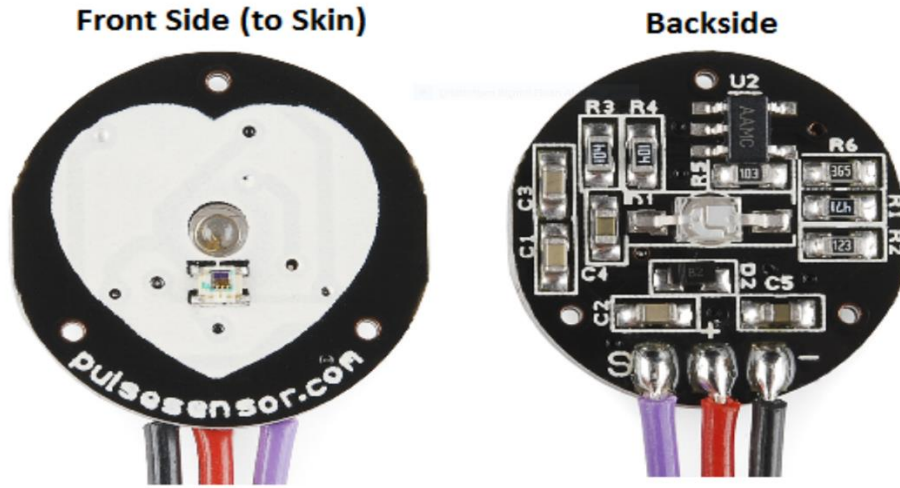
Bu MSP430 Çift Yönlü Ziyaretçi Sayacı Projesi , bir salona, alışveriş merkezine, ofise, giriş kapısındaki işlemlere giren kişi sayısını saymak için kullanılabilir. Park alanlarının kapılarında ve diğer halka açık yerlerde de kullanılabilir. Cihaz, kapıdan giren toplam kişi sayısını ve aynı kapıdan çıkan toplam kişi sayısını sayar. Ve son olarak, giren kişi sayısından çıkan kişi sayısını çıkararak ekranda odada ki toplam kişi sayısını göstermektedir.

1

PULSE SENSÖR

Mikrodenetleyici projelerinde kullanabileceğimiz bir nabız sensörüdür. Sensör parmağınızdaki kılcal kan damarlarının genişlemesinden kaynaklanan ışık değişimini ölçerek kalp atışını algılar.

Bir kalp atışı meydana geldiğinde, kan insan vücuduna pompalanır ve kılcal dokulara sıkışır. Bu kılcal dokuların hacmi, kalp atışının bir sonucu olarak artar. Ancak kalp atışları arasında (arka arkaya iki kalp atışı arasındaki süre) kılcal dokulardaki bu hacim azalır. Kalp atışları arasındaki hacimdeki bu değişiklik, bu dokulardan geçecek ışık miktarını etkiler.



Şekil 1.1 Pulse Sensör

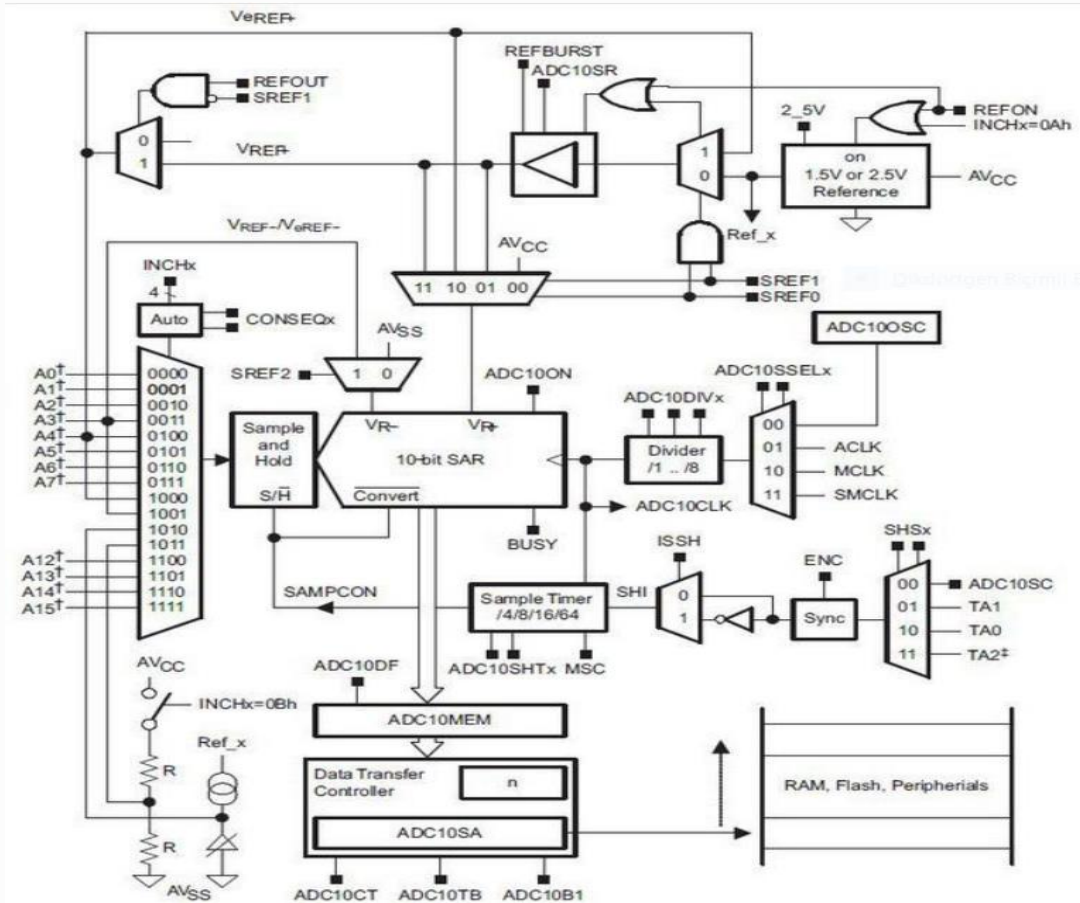
2.1 Pin Yapısı

Bu nabız sensörü üç pinden oluşur.

- Birincisi bu sensöre toprak sağlamak için kullanılan GND pinidir ve kaynak toprak pinine bağlanır.
- (+) pin, VCC pinidir. Bu sensör 3,3V ile 5V dc besleme voltajıyla çalışır.
- (S) Analog bir pin olan A0 pinidir ve analog sinyal almak için kullanılır.
- Sensörün merkezindeki LED, sensörün kalp atış hızını algılamasına yardımcı olur. Bunun yanında LED'in altında kalan devre ise kalp atış hızı nabız sensörünün okumasını etkileyen gürültüyü ortadan kaldırmak için kullanılır.

Analog sayısal dönüştürücüler mikrodeneleyici gibi sayısal sistemler ile analog sistemler arasında köprü görevi görür. köprü tek yönlü çalışır. Analog sayısal dönüştürücüler analog bir işaretin mikrodeneleyici gibi sayısal sistemler tarafından alınması için kullanılır. Mikrodeneleyici benzeri sayısal sistemlerden analog sistemlere analog bilgi göndermek için ise analog sayısal dönüştürücülerin tersi olarak çalışan sayısal analog dönüştürücüler kullanılır.

Analog sayısal dönüştürücüler, ADC, Analog to Digital Converter, Analog Dijital çevirici gibi isimlerle de anılabilir. Günümüzde çoğu mikrodeneleyicide ADC birimi bulunmaktadır. Aynı şekilde harici entegre olarak üretilen çeşitli özelliklere sahip ADC entegreleri de bulunmaktadır.



Şekil 2.1 MSP430 ADC Block Diagram

2.1 MSP430 ADC Blok Diagram

Şekil 2.1 ADC10 blok diyagramı görülmektedir. Görüldüğü üzere birimin ortasında 10 bit SAR(Successive Approximation Register/Ardışıl Yaklaşım Kaydedici) ADC bulunmaktadır. Onun dışında adcnin giriş kanalları ve referans girişleri görüyoruz.

Genel olarak ADC10 birimi harici veya dahili girişlerine uygulanan analog işareti referans gerilimlerini temel alarak 10 bit çözünürlükte sayısal değere dönüştürür. Referans gerilimleri denetleyicinin çalışma gerilim değerlerinin altında yada üstünde olmamalıdır. 10 bit değişken 0-1023 arasında değişen bir değişkeni ifade eder. Örneğin $V_{ref-}=0V$ ve $V_{ref+}=2.5V$ olarak seçersek, $2.5/1023=2.44mV$ aralıklar ile analog işaretimizi örnekleriz.

Sonuç olarak 0-2.5V gerilim değerini 0-1023 arası sayısal değere dönüştürürüz. Bu sayede analog değerimizi işlemcimizin anlayıp işleyebileceği sayısal değere çevirmiş oluruz.

ADC Saat Kaynak Seçimi

Saat kaynak seçimi ADC çevrim birimi hızını belirler.ADC10 modülü için işaret kaynağını ayarlamak için **ADC10SSEL** bitlerini istediğimiz kaynağa göre değiştirmemiz gerekiyor.İşaret kaynağı olarak **SMCLK, MCLK, ACLK ve ADC10CLK** seçilebilir. Genellikle yaygın olarak daha hızlı olduğu ve daha az güç harcadığı için **ADC10CLK** kullanılmaktadır. Bu kaynağın frekansı **1.5Mhz** ile **6.3 Mhz** arasında değişebilmektedir.

Hemen yanında görünen **ACD10DIV** bloğu ise seçtiğimiz işaret kaynağını kaç bölme için kullanılmaktadır. Bu frekans bölücü, seçilen kaynağı 1, 2, 3, 4, 5, 6, 7, 8e bölebilmektedir.

ADC10 Referans Gerilim Üretici

ADC10 Referans gerilimi üretici dahili olarak **2.5v** ve **1.5v** gerilim üretebilir. Eğer bu birimin ürettiği referans gerilimi kullanılmak isteniyorsa **REFON=1** yapılmalıdır. $REFON = 1$ iken **REF2_5V = 1** ise bu birimden alacağımız gerilim **2.5v** tur. Eğer **REF2_5V = 0** ise, bu birimden **1.5v** gerilim alınmaktadır.

ADC10 Düşük Güç Özellikleri

MSP430 ADC10 Modülündeki yukarıda bahsedilen dahili referans gerilimi üreten kısım, tamamen low power uygulamalar için tasarlanmıştır. Ve bu birimin otomatik olarak kapalı durumdan açık duruma gelmesi 30uS den daha az bir sürede gerçekleşmektedir.

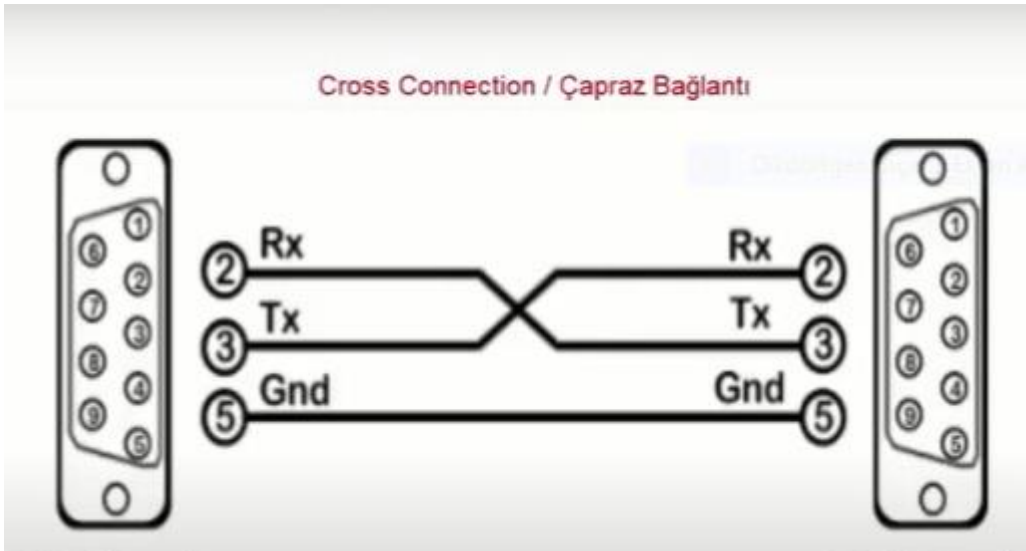
Ayrıca ADC10 modülünün tamamı, kullanılmadığı zaman da kendini pasif duruma getirmektedir. Eğer ADC10 modülü kullanılacaksa modül, otomatik olarak tekrar aktif hale gelmektedir. ADC10 modülü kapalı durumda iken hiç akım harcamamaktadır.

Kablolu haberleşme “Seri” ve “Paralel” olmak üzere iki sınıfta incelenir. Seri haberleşmede göndermek istediğimiz bitler bir hat üzerinden , paralel haberleşmede ise gönderilecek bitler birden fazla hat üzerinden aynı anda gönderilir. Paralel haberleşme seriye göre daha hızlı bir haberleşmedir, ancak birden fazla hat kullanılması sistemin karmaşıklığını arttırmaktadır. Yüksek hıza ihtiyaç duyulan yerlerde örneğin “Main Data Bus”ta, paralel haberleşme kullanılır. Hızın kritik seviyede önemli olmadığı yerlerde ise, örneğin sensör ile işlemci arasında, genellikle seri haberleşme tercih edilmektedir.

3.1 Asenkron Haberleşme

Seri haberleşme protokolü arasında en önemlileri USART, SPI, I2C, RS232 protokolleridir.

USART’da “Rx” ve “Tx” olmak üzere iki haberleşme hattı bulunmaktadır. Rx sistemin haberleşeceği sistemden bilgi almak için, Tx ise haberleşilecek sisteme bilgi göndermek için kullanılan hattır. Haberleşme için çapraz bağlantı kullanılır.



Şekil 3.1 Çapraz Bağlantı

Projemde Msp430 rx pinini ttl tx ve MSP430 tx pinini ttl rx bacağına bağlayarak çapraz bağlantı kullandım .

3.2 Asenkron RS232 Haberleşme

RS232 bir asenkron (yani haberleşme esnasında bir saat sinyali gönderilmez.) haberleşme standartıdır . Kısa mesafede haberleşme sağlamak için 1960'ların başlarında ortaya çıkmıştır.

RS232, genelde farklı cihazlar arasında haberleşmeyi sağlayan bir seri haberleşme protokolüdür. Bu protokol sayesinde denetleyicimiz ile bilgisayarımızı haberleştirebilir, bilgisayar ile kontrol edilen devreler yapabiliriz.

RS232'nin +15 – -15V gerilimleriyle çalıştığını biliyoruz. Normalde MSP430 ile bu gerilim değerlerini elde edemiyoruz. Ancak launchpad bize çok güzel bir özellik sağlıyor. Herhangi bir entegre kullanmadan, bilgisayar tarafında oluşturduğu sanal seri port aracılığıyla, denetleyicinin P1.1 (TXD) ve P1.2 (RXD) pinleri ile RS232 haberleşmesi yapabiliyoruz.

3.3 RS232 Kullanımı

RS232, **TTL** seviyesinde (Transistor – Transistor Logic) iletim yapmaz. Gerilim aralığı +12v ve -12v arasındadır. Bilgisayarın binary sayı sisteminde 0 değeri 0V'u, 1 değeri ise 5V temsil etmektedir. Buna aynı zamanda TTL seviyesi de denir. Ancak veri iletimini TTL seviyesinde gerçekleştirmek oldukça risklidir. Çünkü TTL seviyesi gürültülerden çok çabuk etkilenip bozulmaya uğrayabilir. Burda devreye – kısa mesafeler için – RS232 giriyor. RS232, TTL seviyesinde iletilmiş olan veriyi kendi gerilim aralığı olan +12V ve -12V aralığına çekerek dışardan gelecek olan gürültülerden etkilense bile veri kaybolması yaşanmamasını sağlıyor.

3.4 Usb - TTL Dönüştürücü (PL2303)

PL2303 USB-TTL seri dönüştürücü , USB birimini TTL arayüzünde kullanmanıza imkan sağlayan USB-seri dönüştürücüdür. TTL arayüzüne sahip herhangi bir sensör birimini veya deneme kartını bu kablo aracılığı ile USB portu üzerinden bilgisayarınıza bağlayarak iki birim arasında haberleşme sağlamak mevcuttur.

3.5 Realterm Kullanımı

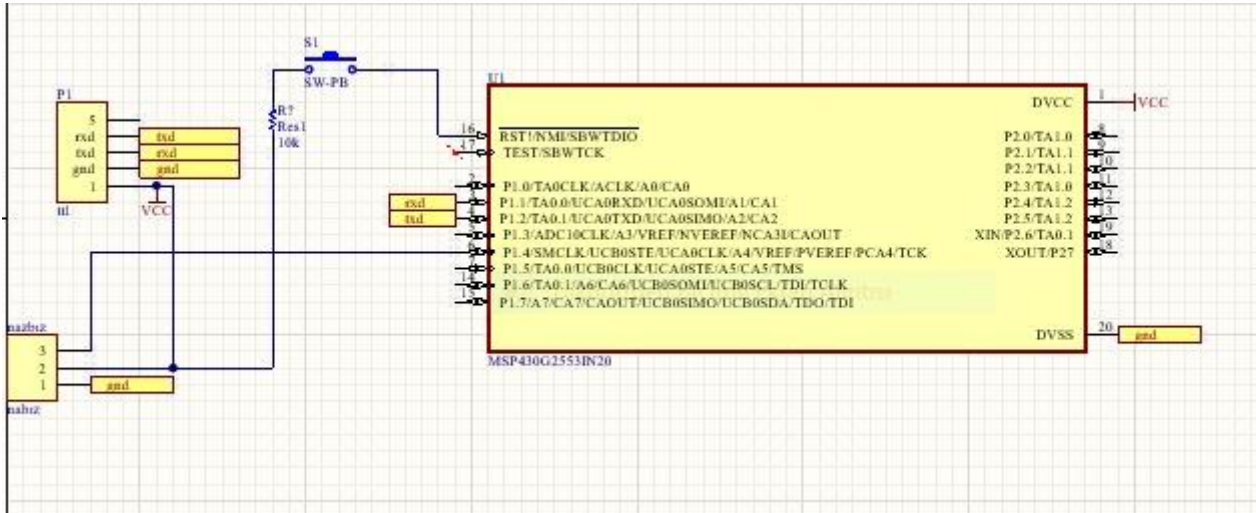
Programı açtığımızda karşımıza haberleşme parametrelerini kolayca ayarlayabileceğimiz bir ekran çıkıyor.

unit8” seçeneğinin seçersek seri ekrana yazdırılacak tüm karakterlerin ASCII karşılığını görebiliriz. Bu da bize seri ekrana giden karakterleri daha açık yorumlama imkanı verir.

Port” sekmesinden ayarlamak istediğimiz haberleşme parametrelerini ayarlayabiliriz. Ayarladıktan sonra “Open” kısmına tıklayarak veri iletimi için programı aktif hala getirebilirsiniz.

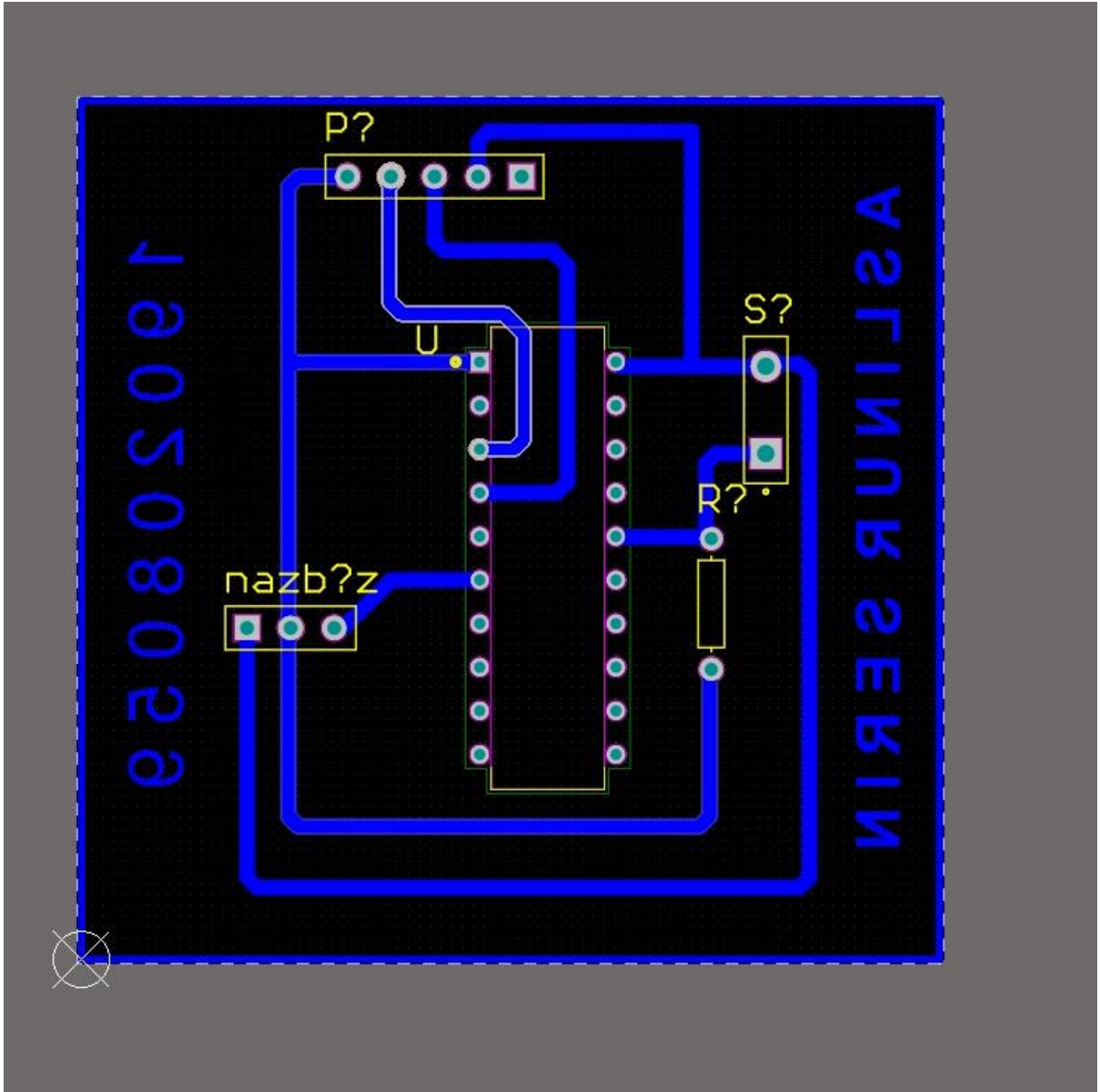
Sistem MSP430 bağlanmış kılcal kan damarlarının genişlemesinden kaynaklanan ışık değişimini ölçerek kalp atışını algılayan sensör ile ttl yardımıyla rs232 haberleşme protokolünü kullanarak analog veriyi sayasala çevirmeyi hedeflemektedir.

4.1 Altium Şematik Çizimi



Şekil 4.1 Altium Şematik Çizimi

4.2 PCB ÇİZİMİ



Şekil 4.2 PCB Çizimi

5

Sistemin Kodu

```
#include <intrinsics.h>
#include <io430.h>
```

```
int gelendeger = 0;
void uart_init(void);
void adc_init(void);
```

```
int main(void)
{
    WDTCTL = WDTPW | WDTHOLD; // stop watchdog timer
```

```
    DCOCTL = 0;
    BCSCTL1 = CALBC1_1MHZ; // Clock saatini 1 MHZ ayarlamak
    DCOCTL = CALDCO_1MHZ;
    __delay_cycles(1000000);
```

```
    P1SEL = BIT1 + BIT2; // P1.1/P1.2 rx/tx olarak kullanmak için yapılan ayar görevleri seçiyor
    ikisinde aktif edince haberleşme pinlerini kullanıyoruz
    P1SEL2 = BIT1 + BIT2;
```

```
    adc_init();
    uart_init();
```

```
    __bis_SR_register(LPM1_bits+GIE); // lmpx modundayken receivden bir şey gelirse uyanıyor
```

```
    while(1);
}
```

```
void uart_init(void)
{
    UCA0CTL1 |= UCSSEL_2; //SMCLK işlemciden gelen clocku aldı
    UCA0BR0 = 104; // 1000000/9600 BAUDRATE AYARLAMAK baud rate msw bit
    UCA0BR1 = 0; // lsw biti baud hızı ayarlamak için
    UCA0MCTL = UCBRS2 + UCBRS1; // tam çıkmayan sonuçlarda hata payı düzeltmek için hata
    payını gidermek için modülasyon katsayısı seçildi
    UCA0CTL1 &= ~UCSWRST; //ucaoctl1 de ucswrst resetleme biti bulunmaktadır yazılımsal
    olarak temizlenmezse uart çalışmayacaktır
}
```

```

void adc_init(void)
{
    ADC10CTL0 &= ~ENC; //Can be modified only when ENC = 0
    ADC10CTL0 = ADC10SHT_3 + ADC10ON + REF0N + ADC10IE; // ADC10ON, interrupt
    enabled , X64 örnekleme kaç adc clock darbesi alınacağını söyler
    ADC10CTL1 = INCH_4 ; // input p1.4 analog giriş ayarı
    ADC10AEO |= BIT4; // Analog (input) enable control register satlayıcısına bit 4 girişe seçtiğimiz
    pini aktif etmek için
    ADC10CTL0 |= ENC + ADC10SC; // Sampling and conversion start

}
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void)
{
    gelendeger = ADC10MEM; // sensörden alınan veri adc memory ye kaydoluyor biz de bu veriyi
    gelen deger degiskenine atıyoruz
    while (!(IFG2&UCAoTXIFG)); //TX müsait olana kadar bekle
    UCAoTXBUF = gelendeger; // bu degiskende tutulan degeri tx bufferina aktararak uartla
    gönderilecek veriyi söylemiş oluyoruz.
    ADC10CTL0 |= ENC + ADC10SC; // enc biti adc10sc biti ile kullanılır adc çevrşm tek
    çevrim modundayken adc10sc biti ile bu bitte logic bir yapıldığında başlamaktadır
}

```

Kaynaklar

<https://www.instructables.com/Ultrasonic-Sensor-with-MSP430-and-IARCCS/>

https://www.ti.com/lit/an/slaa274b/slaa274b.pdf?ts=1652964543833&ref_url=https%253A%252F%252Ftr.search.yahoo.com%252F

