**jffkguk**

**BAŞKENT ÜNİVERSİTESİ**

**ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ**

**EEM415 – MİKROİŞLEMCİLER ARAYÜZ TEKNİKLERİ**

**PROJE RAPORU**

PROJE ADI: İki Çıkışlı Kodlayıcının Verisi Kullanılarak Pozisyon Tespiti

ÖĞRENCİ AD,SOYAD: FARUK TÜRKOĞLU

NUMARA: 20594706

**Proje Konusunun Tanımı ve Amacı:**

2009-2010 Güz dönemi dersi olan EEM 415 Mikroişlemci Arayüz Teknikleri adlı derste öğrendiklerimizi bu proje yardımıyla teorik olarak öğrenme imkanını da bulduk. Projenin konusu kısaca, iki çıkışlı bir kodlayıcının verisi kullanılarak elimizde bulunan bir adet encoder bir motorun pozisyonunun tespitidir. Sisteme ait akış diyagramı EK-2’de verilmiştir.

Projede pozisyon tespiti için bir adet motor encoder kullanıldı. Motor encoder, motor ekseni üzerine monte edilen encoder ile motorun pozisyonu direkt olarak ölçülebilir. Encoderler, bir çeşit sinyal üreticileridir. Bağlı bulunduğu motorun veya şaftın hareketine karşılık, sayısal (dijital) bir elektrik sinyali üreten elektromekanik bir cihazdır.Çıkış Tiplerine göre ikiye ayrılırlar:

Mutlak Tip (Absolute) Encoderler:

Bu Encoderler, her pozisyonlarında farklı sayılardaki bitlerden oluşan dijital bit dizileri şeklinde birbirine benzemeyen çıkışlar üreterek, gerçek pozisyonlarını tam olarak gösterirler. Enerjisi kesilse bile mevcut durumunu korur.

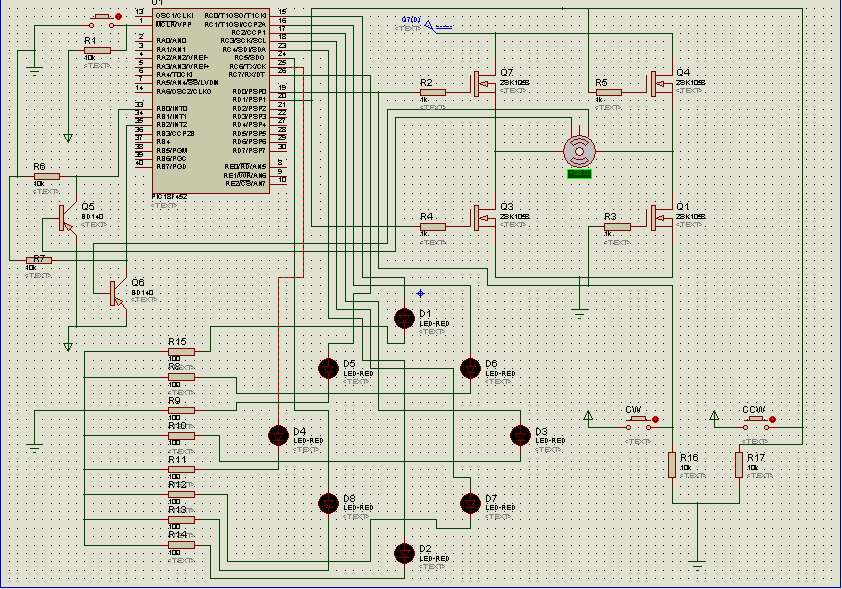
Artımsal Tip (Incremental) Encoderler:

Bu Encoderler, her pozisyonda benzer çıkış sinyalleri (Kare Dalga) üretirler. Bu sinyaller hız ölçümü (bir Takometre ile birlikte) ya da sayma işlemi için (bir Sayıcı ile birlikte) kullanılabilirler.  
 Bu tip encoderler abolute encoderlerdan farklı olarak enerjilendikleri anda bulundukları yeri sıfır kabul eder ve bunun üstüne işlem yapar.

512 / 1024 / 1048 / 4096 pulse ve üstü encoderler mevcuttur. Bu pulseler encoder ‘ın 1 tam tur attığında üreteceği pulse sayısıdır. Tabiki 4096 pulseli bir encoder daha hassastır. Çünkü nokta aralıkları daha azdır.

Projede motora bağlı bir encoderin pozisyonuna göre motorun kontrolünü sağlanıyor. Hangi konumda olduğu bulunuyor.

Projede bir adet motor encoder, motorun ileri ve geri dönmesi için gerekli transistörler, bir adet Pic18F452 ve pozisyonun tespiti için ledler kullanılmıştır. Projenin Proteus şeması aşağıdaki şekilde gösterilmiştir.



Projenin Proteus Şeması

**Sistemin Tanımlanması:**

Sistem, encoder motordan çıkan harici kesmelerin (External Interrupt), Pic18F452 ‘ye gitmesiyle çalışır. Sistemde İnterruptları aşağıda verilen tabloya göre ayarladık.

**INTCON REGISTER**

R/W-0 R/W-0 R/W-0 R/W-0 R/W-0 R/W-0 R/W-0 R/W-x

GIE/GIEH PEIE/GIEL TMR0IE INT0IE RBIE TMR0IF INT0IF RBIF

bit 7 bit 0

**bit 7** **GIE/GIEH:** Global Interrupt Enable bit

When IPEN = 0:

1 = Enables all unmasked interrupts

0 = Disables all interrupts

When IPEN = 1:

1 = Enables all high priority interrupts

0 = Disables all interrupts

**bit 6** **PEIE/GIEL:** Peripheral Interrupt Enable bit

When IPEN = 0:

1 = Enables all unmasked peripheral interrupts

0 = Disables all peripheral interrupts

When IPEN = 1:

1 = Enables all low priority peripheral interrupts

0 = Disables all low priority peripheral interrupts

**bit 5** **TMR0IE:** TMR0 Overflow Interrupt Enable bit

1 = Enables the TMR0 overflow interrupt

0 = Disables the TMR0 overflow interrupt

**bit 4** **INT0IE:** INT0 External Interrupt Enable bit

1 = Enables the INT0 external interrupt

0 = Disables the INT0 external interrupt

**bit 3** **RBIE:** RB Port Change Interrupt Enable bit

1 = Enables the RB port change interrupt

0 = Disables the RB port change interrupt

**bit 2** **TMR0IF:** TMR0 Overflow Interrupt Flag bit

1 = TMR0 register has overflowed (must be cleared in software)

0 = TMR0 register did not overflow

**bit 1** **INT0IF:** INT0 External Interrupt Flag bit

1 = The INT0 external interrupt occurred (must be cleared in software)

0 = The INT0 external interrupt did not occur

**bit 0** **RBIF:** RB Port Change Interrupt Flag bit

1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)

0 = None of the RB7:RB4 pins have changed state

İnterrupt tablosu

Sistemde, encoder motordan Pic18F452 ‘ye kare dalga sinyali gitmektedir. Sinyali de her yükselen kenar için bir interrupt gönderecek şekilde ayarlıyoruz.

Proteus’da motoru her devirde 8 pulse göndermek üzere ayarladık (Pulses Per Revolution=8); çünkü kullandığımız Pic18F452 8 bit çalışmaktadır. Yani motor her bir tur attığında sekiz adet pulse üretmektedir. Motorun pozisyonunun tespiti için ise 8 adet led kullandık; çünkü kullandığımız Pic’in çıkışında 8 adet bacak bulunmaktadır. Motor hangi pozisyonda durur ise o açıya yaklaşık olan led yanarak bize motorun pozisyonunu gösterecektir.

Motorun dönme yönü Pic18F452 tarafından kontrol edilmektedir. Motorun dönme yönü sistemde bulunan butonlar yardımıyla seçilmektedir. Butonlardan birine sürekli basıldığında, basılan buton motoru hangi yöne doğru döndürüyor ise o yöne sürekli olarak döndürmektedir. Motor dönerken ise sisteme entegre edilmiş olan ledler yardımıyla motorun yönü ve hızı görülebilmektedir. Yerleştirdiğimiz 8 adet led’in konumları ise sırasıyla 0, 45, 90, 135, 180, 225, 270, 315 derecede konumlanmıştır. Motor dönerken bu ledler sırasıyla yanar ve butondan elimizi çektiğimizde motor hangi konumda durmuşsa, o konuma karşılık gelen led yanarak bize motorun pozisyonunu göstermektedir.

İleri doğru (Clock Wise-CW) butonuna basıldığında PORT D=1 olur. PORT D=1 ve motordan pulse gittiğinde motor ileri doğru(saat yönünde) döner ve ledler de saat yönünde yanar. Geri butonuna (Counter Clock Wise-CCW) basıldığında PORT D=2 olur. PORT D=2 ve motordan pulse gittiğinde motor bu sefer saatin ters yönü döner. Ledler ise kodumuzda görüldüğü üzere X değişkenine bağlı olarak ileri ya da geriye doğru olmak üzere ve kodta yaptığımız sonsuz döngü yardımıyla sürekli olarak yanmaktadır. Sistem için kullanılan kod EK-1’de verilmiştir.

**Sonuç ve Öneriler:**

Bu proje çalışmasında sonuç olarak, PIC Mikrodenetleyici ile belli bir denetim sistemi geliştirdik. Bu sistemin kontrolünde, verilen bir encoder motorun, mikrodenetleyici yardımıyla hangi yöne doğru döneceğini seçip buna göre çıkışta bulunan 8 adet led kullanılarak motorun hangi konumda bulunduğunu tespit ettik.

Aynı sistem yardımıyla encoder kullanarak yine bir motorun hızını da tespit edebilirdik.

**EK-1**

#include "INT18XXX.H"

#pragma origin 0x8

#pragma config[1] = 0xF1 // Osilatör: XT

#pragma config[2] = 0xFE & 0xF9 // PWRT açık, BOR kapalı

#pragma config[3] = 0xFE // Watchdog Timer kapalı

int8 x;

interrupt int\_server(void) // KESME SUNUCU FONKSİYONU

{

int\_save\_registers //W, STATUS, ve BSR yazmaçlarının kesme gelmeden önceki değerlerini kaydeder

INT0IF = 0; // INT0 kesme bayrağını sıfırla(yeni kesmeler gelebilir)

if(x==-1)

x=7;

if(x==8)

x=0;

if(x==0)

PORTC=1;

if(x==1)

PORTC=2;

if(x==2)

PORTC=4;

if(x==3)

PORTC=8;

if(x==4)

PORTC=16;

if(x==5)

PORTC=32;

if(x==6)

PORTC=64;

if(x==7)

PORTC=128;

if(PORTD==1)

x++;

if(PORTD==2)

x--;

int\_restore\_registers //CC8E Macro to restore W, STATUS, and BSR registers

}

void main()

{

x=0;

TRISC=0;

PORTC=0;

TRISD=1;

PORTD=0;

INT0IE=1; // INT0 kesmesi açık

INTEDG0=1; // INT0 kesmesi yükselen kenarda aktif olacak

GIE=1; // Bütün kesmeler kullanılabilir

sonsuz:

goto sonsuz;

**}**

**EK-2**

HAYIR

EVET

EVET

HAYIR

EVET

EVET

HAYIR

HAYIR

EVET

HAYIR

Pozisyonun bulunduğu ledi yak

Pozisyonun bulunduğu ledi yak

Butona hala basılıyormu

Butona hala basılıyormu

Ledleri saatin tersi yönünde yak

Ledleri saat yönünde yak

CCW BUTONUNAMI BASILI

CW BUTONUNAMI BASILI

İNT VARMI