***EEM 415 – Mikroişlemci Arayüz Teknikleri  
Proje Çalışması***

***Adı Soyadı :*** *Evrim GÜNER*

***Numarası :*** *20894581*

***Proje Konusu :*** *Polling Tabanlı Klavyeden Basılan Butonu Algılama*

***Öğr. Elemanı :*** *Doç. Dr. Hamit ERDEM*

Projede proteus simulasyon ortamında hazırlanan (3x3) + 1 düğmelik bir matris klavye sisteminin, mikrodenetleyici tarafından polling yöntemi ile taranması ve basılan tuşun 7 segment göstergede gösterilmesi istenmiştir.

Projede kullanılması istenilen polling yöntemini kısaca özetleyecek olursak, mikrodenetleyici kendisine bağlı olan klavye düğmelerinin basılma durumlarını bir döngü içerisinde tek tek kontrol edecek ve basılma durumu söz konusu olduğunda 7 segment’ de düğme numarasını yazdıracaktır. Polling yöntemi, girişlerin devamlı olarak taranması sebebiyle mikrodenetleyiciyi gereksiz yere meşgul eden bir yöntemdir. Mikrodenetleyici bu tarama esnasında başka işlemlerle ilgilenememektedir. Polling yöntemi yazılımcı açısından yazması daha kolay olan bir yöntemdir ve tek bir işlem yapması için tasarlanan sistemlerde kullanılabilir. Mikrodenetleyicinin farklı ve çok adımlı işlemlerle ilgilenmeyeceği sistemlerde polling yöntemi başarılı olarak uygulanabilir. Polling yöntemine alternatif olan yöntem ise interrupt tabanlı algılamadır. İnterrupt tabanlı algılamada mikrodenetleyici kendi işlemlerini işlerken, interrupt pin’ ine bir sinyal gelmesi durumunda yaptığı işlemi devam etmek üzere durdurarak interrupt yordamına girer ve yapılması istenen işlemleri yapar. Bu sistem daha karmaşık ve mikrodenetleyici verimliliğinin önemli olduğu sistemlerde kullanılır.

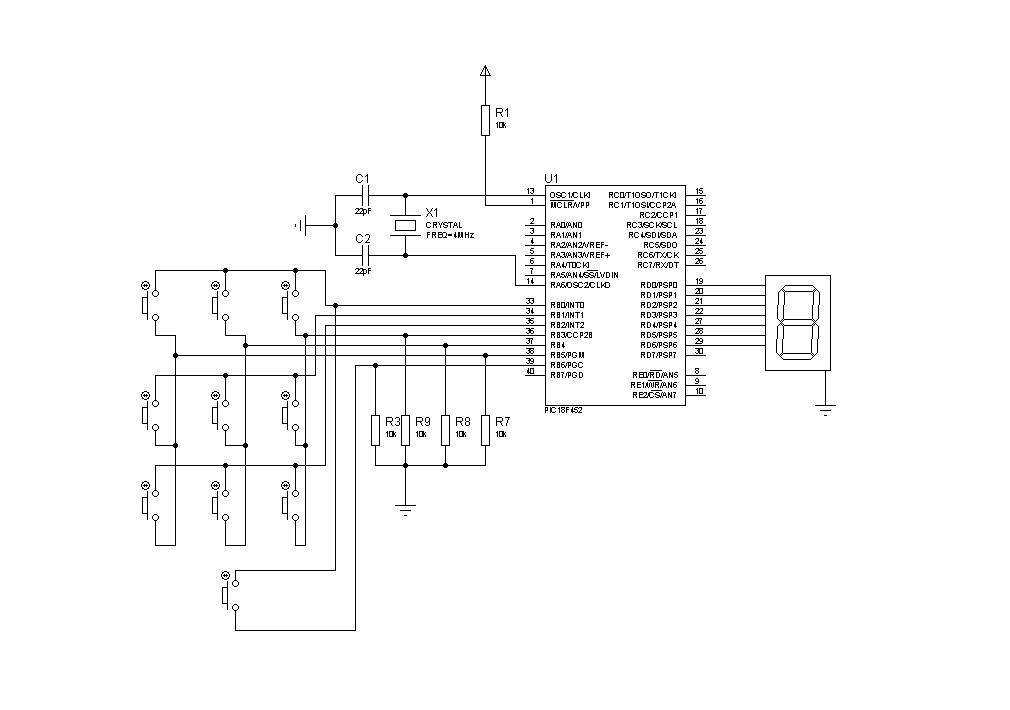
Hazırlan projede; (3x3) + 1 adet düğme bir mikrodenetleyiciye bağlanmıştır. Burada matris yapılı bir klavye üzerindeki 0 – 9 arası numaralanmış düğmelerin göstergede numaralarının gösterilmesi amaçlanmıştır. Mikrodenetleyici düğmeleri polling ile izler ve herhangi bir düğmenin basıldığı fark edildiğinde o düğmenin numarasını 7 segment göstergede gösterir.

Kurulan sistemde polling ile düğme algılama yapısını açıklayacak olursak; algılama işlevi için mikrodenetleyicinin B portu kullanılmıştır. B portunun ilk 3 biti çıkış, sonraki 5 biti ise giriş olarak belirlenmiştir. Giriş bitleri normal durumda pull down dirençleri topraklanmıştır, yani giriş bitleri normal durumda 0 okurlar. Çıkış bitleri sırasıyla high yapılarak giriş bitlerindeki değişim gözlenir. Çıkış biti high yapılmış bir satırda herhangi bir düğmeye basılması, o düğmenin bağlı olduğu girişi, 0 durumundan high durumuna getirir. Bu sayede mikrodenetleyici hangi çıkışın high olduğunu bildiğinden satır numarasını, hangi girişin high olduğunu bildiğinden de sütun numarasını hesaplamış olur ve çıkış portu olan D portuna düğmenin numarasını hexadecimal olarak gönderir ve görüntü elde edilir. Bu sistemde polling işlemi çıkış pinlerinin sıra ile high yapılması ile olur. Her çıkış biti için, bit high durumda iken sütunlardan gelen inputlarda high olan var mı diye bakılır, eğer inputlarda değişim olmamışsa high olan çıkış bitinin low a çekilerek bir sonraki bit in high yapılması ve sütunlardan gelen inputlara tekrar bakılması şeklinde tekrar eden bir döngü yapısı varıdır.

Sistemde kullanılan 7 segment gösterge ortak katotludur, yani gösterge içerisindeki ledlerin toprak bağlantıları ortaktır ve yanabilmeleri için mikrodenetleyici tarafından o led e giriş uygulanması gerekir.

Devrede herhangi bir düğmeye basılmamışsa 7 segment gösterge üzerinde yanmaz (boş görünür), herhangi bir düğmeye basıldığında ise düğme numarasına göre 0-9 arasında bir değer, düğmeye basılı tutulan süre boyunca göstergede gösterilir.

Kurulan sistem görünümü aşağıdaki resimde gösterilmiştir.



***Proje Kodu Aşağıdaki Gibidir :***

// Evrim Güner - 20894581 EEM415 Projesi, Polling Tabanlı Klavyeden Basılan Butonu Algılama

// Mikrodenetleyici Timer ve Clock Ayarlamaları

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

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

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

void ayarlar(); // Port ayarlamalarının yapıldığı fonksiyonların tanımı

void bekle(unsigned long t); // t milisaniye gecikme sağlayan fonksiyon tanımı

void main()

{

ayarlar(); // Port Ayarlamaları Programın ilk adımında yapılıyor.

//-----------------------------------------------

anadongu:

PORTB.0=1; // Port B' nin 0. biti high yapılıyor

PORTB.1=0; // Port B' nin 1. biti low yapılıyor

PORTB.2=0; // Port B' nin 2. biti low yapılıyor

if(PORTB.0==1){ // Port B' nin 0. biti high olduğu sürece, input bitlerine tek tek bakılacak

if(PORTB.5 == 1){ // Port B' nin 5. biti high olmuşsa göstergede 1 göster.

PORTD=0x06;

bekle(300);

PORTD=0;

}

if(PORTB.4 == 1){ // Port B' nin 4. biti high olmuşsa göstergede 2 göster.

PORTD=0x5B;

bekle(300);

PORTD=0;

}

if(PORTB.3 == 1){ // Port B' nin 3. biti high olmuşsa göstergede 3 göster.

PORTD=0x4F;

bekle(300);

PORTD=0;

}

if(PORTB.6 == 1){// Port B' nin 6. biti high olmuşsa göstergede 0 göster.

PORTD=0x03F;

bekle(300);

PORTD=0;

}

}

PORTB.1=1; // Port B' nin 1. biti high, 0 ve 2. bitleri low yapılır.

PORTB.0=0;

PORTB.2=0;

if(PORTB.1==1){ // Port B' nin 1. biti high olduğu sürece, input bitlerine tek tek bakılacak

if(PORTB.5 == 1){ // Port B' nin 5. biti high olmuşsa göstergede 4 göster.

PORTD=0x66;

bekle(300);

PORTD=0;

}

if(PORTB.4 == 1){ // Port B' nin 4. biti high olmuşsa göstergede 5 göster.

PORTD=0x6D;

bekle(300);

PORTD=0;

}

if(PORTB.3 == 1){ // Port B' nin 3. biti high olmuşsa göstergede 6 göster.

PORTD=0x7D;

bekle(300);

PORTD=0;

}

}

PORTB.2=1; // Port B' nin 2. biti high, 0 ve 1. bitleri low yapılır.

PORTB.0=0;

PORTB.1=0;

if(PORTB.2==1){ // Port B' nin 2. biti high olduğu sürece, input bitlerine tek tek bakılacak

if(PORTB.5 == 1){ // Port B' nin 5. biti high olmuşsa göstergede 7 göster.

PORTD=0x07;

bekle(300);

PORTD=0;

}

if(PORTB.4 == 1){ // Port B' nin 4. biti high olmuşsa göstergede 8 göster.

PORTD=0x7F;

bekle(300);

PORTD=0;

}

if(PORTB.3 == 1){ // Port B' nin 3. biti high olmuşsa göstergede 9 göster.

PORTD=0x6F;

bekle(300);

PORTD=0;

}

}

goto anadongu; // Döngü başına dönüş

//-----------------------------------------------

}

//////////////////////////////////////////////////////////////////////////////////////////////////

void ayarlar() // Bütün başlangıç ayarlarının tamamlandığı kısım

{

GIE=0; // Bütün kesmeleri kapat

TRISA=0xFF; // Port A input

TRISB=0xF8; // Port B' nin 0-3 arası bitleri output, 4-7 arası bitleri input

TRISC=0; // Port C output

TRISD=0; // Port D output

TRISE=0; // Port E output

// Program ilk çalıştırıldığında çıkışlarda değer görülmemesi için ilk çıkış değerleri 0 alınır.

PORTC=0;

PORTD=0;

PORTB=0;

}

void bekle(unsigned long t) //t milisaniye gecikme sağlar

{

unsigned x;

for(;t>0;t--)

for(x=140;x>0;x--)

nop();

}