



## PICPLC'ler için Programlama Arayüzü Geliştirilmesi Developing of a Programming Interface for PICPLCs

Serhat KILIÇARSLAN<sup>1</sup>, Gökhan GELEN<sup>2</sup>

<sup>1</sup> Enformatik Bölümü  
Gaziosmanpaşa Üniversitesi, Tokat  
serhat.kilicarslan@gop.edu.tr

<sup>2</sup> Mekatronik Mühendisliği Bölümü  
Gaziosmanpaşa Üniversitesi, Tokat  
gokhan.gelen@gop.edu.tr

### Özet

PLC'ler endüstriyel otomasyon sistemlerinde yaygın olarak kullanılan kontrol cihazlarıdır. Çeşitli üreticilerin farklı modellerde birçok PLC'si mevcuttur. Son zamanlarda, ticari PLC'lere alternatif mikrodenetleyici temelli ucuz PLC'ler tasarlanmıştır. Bu PLC'ler diğerleri gibi komut listesi dili kullanılarak programlanabilmektedir. Bu çalışmada, PIC mikrodenetleyicisi temelli bir PLC için merdiven diyagramı programlarını PIC koduna dönüştüren programlama arayüzü geliştirilmiştir. Gerçekleştirilen arayüz web sayfası sitilinde olduğundan çeşitli platformlarda çalıştırılabilir. Arayüz çeşitli PLC kod örnekleri kullanılarak test edilmiştir. Testler sonucunda arayüzün sorunsuz çalıştığı görülmüştür.

### Abstract

PLCs are widely used in industrial automation system as control devices. Many PLCs from various manufacturers are available in different models. Recently, microcontroller based inexpensive PLCs are designed as alternative to commercial PLCs. These PLCs can be programmed as others by using instruction list language. In this study, a programming interface is developed to convert Ladder diagrams to PIC codes for a PIC based PLC. The realized interface can be run on different platforms because it is in the style of a webpage. The interface has been tested for various sample PLC codes. According to test results, it is observed that the interface works without any problems.

### 1. Giriş

Programlanabilir Lojik Denetleyiciler (PLC) fabrikalardaki makine veya imalat hatlarının kontrolü için gerekli işlemleri yerine getiren bilgisayarlar olarak tanımlanmaktadır [1]. PLC'ler günümüz endüstrisinde otomasyon alanında yaygın olarak kullanılmaktadır. Bunun en önemli sebebi PLC'lerin endüstriyel ortamlarda (nem, ısı, mekanik v.b.) çalışmak üzere tasarlanmış olması ve güvenilir olmasıdır. PLC'ler yaygın olarak merdiven (Ladder) diyagramı dili veya komut listesi (STL) dili kullanılarak programlanmaktadır. Merdiven (Ladder) diyagramı dili, kullanımının kolay olması ve alışılmış kumanda devrelerine benzerliği nedeniyle en yaygın kullanılan dildir.

Günümüzde çok sayıda firmanın üretmiş olduğu farklı boyut, donanım ve fiyatlarda ticari PLC'ler mevcuttur. Ticari PLC üreticilerinin neredeyse tamamı ürünleri ile birlikte kendi programlama yazılımlarının lisansını satmaktadırlar. Bu durum ticari PLC'lerin fiyatlarının yüksek olmasının temel nedenidir [2]. Her ne kadar ücretsiz yazılımlar bulunsun da bu yazılımların genel olmasından kaynaklı uyum ve kullanıcı zorluğu problemleri mevcuttur.

Ticari PLC'lere alternatif olarak mikroişlemci veya mikrodenetleyici temelli daha ucuz sistemler çözüm olarak sunulmuştur [3-8]. Atmega128 entegresi kullanarak 14 adet dijital giriş/çıkış ve 5 adet sadece dijital çıkışlı PLC Tongur tarafından gerçekleştirilmiştir. Ancak, tasarlanan PLC kod üretebilen bir arayüz programı mevcut değildir [5]. Yapılan bir diğer çalışmada ise tasarlanan PLC sadece giriş çıkış ve temel işlemleri yerine getirebilmektedir [8]. Rafat tarafından yapılan çalışmada ise tasarlanan PLC için arayüz mevcuttur ancak PLC'nin giriş çıkış sayısı çok kısıtlıdır [4]. Literatürdeki bazı çalışmalarda ise simülasyon amaçlı programlar ve donanımlar tasarlanmıştır [9-10].

Mikrodenetleyici entegreleri incelendiğinde ücretsiz derleyici programı sunulması, fiyatlarının uygun olması ve kaynak kod desteğinin fazla olması gibi sebeplerden dolayı Microchip firmasının üretmiş olduğu PIC serisi mikrodenetleyicilerin yaygın olarak kullanıldığı görülmektedir. PIC mikrodenetleyicileri kullanılarak ticari ürün olarak satışı gerçekleştirilen PICPLC16 isimli bir PLC de mevcuttur [11]. Ancak bu ürün içinde alışılmış PLC komutları ile programlama yapılmasına olanak sağlayan bir arayüz mevcut değildir.

PIC mikrodenetleyicileri kullanılarak gerçekleştirilen en ciddi tasarımlar Uzam tarafından yapılmış ve raporlanmıştır [12-14]. Bu çalışmalarda PIC16F877A entegresi kullanılarak 16 giriş ve 16 çıkışlı bir PLC ile PIC16F648A entegresi kullanılan ve giriş çıkış sayısı modül eklenerek arttırılabilen iki PLC tasarımı önerilmiştir. Bu PLC'leri programlamada kullanmak üzere yaygın kullanılan PLC programlama dili olan STL diline benzer makro tabanlı komutlar önerilmiştir. Ancak, bu çalışmalarda da kullanıcının görsel olarak merdiven (Ladder) diyagramı programı girebileceği bir arayüz mevcut değildir.

Bu çalışmada Uzam tarafından önerilen tasarım ve makroları kullanan PIC16F877A mikrodenetleyicisi temelli PICPLC için kullanıcıların merdiven (Ladder) diyagramı olarak girdikleri programı ilgili entegrenin koduna dönüştüren bir programlama arayüzü geliştirilmiştir. Tasarlanan arayüz web sayfaları altında çalışabilir nitelikte olduğundan farklı işletim sistemleri yüklü bilgisayarlarda çalışabilmektedir yani platformdan bağımsızdır denilebilir.

Bildirinin organizasyonu şu şekildedir. İkinci bölümde PICPLC'nin donanımı ve temel yazılımı ile makrolar kısaca açıklanmaktadır. Üçüncü bölümde gerçekleştirilen arayüz tanıtılmaktadır. Arayüzün testlerinde kullanılan örnek uygulamalar dördüncü bölümde açıklanmıştır. Son olarak elde edilen sonuçlar sonuç bölümünde sunulmaktadır.

## 2. PICPLC Donanımı ve Temel Yazılımı

Bu kısımda PIC16F877A ile tasarlanan 16 Girişli/16 Çıkışlı PICPLC Donanımı ve yazılımı kısaca açıklanmaktadır. PIC mikrodenetleyicisi temelli PICPLC beş kısımdan meydana gelir. Bunlar;

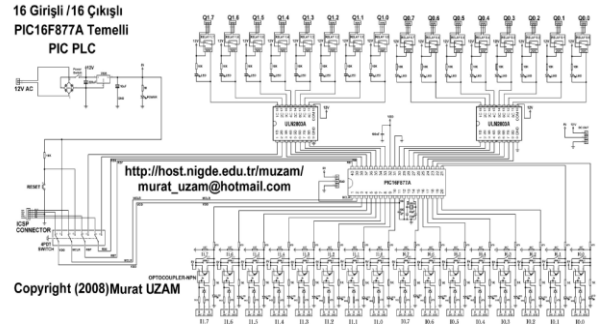
- Güç Birimi
- Program Yükleme Birimi
- Merkezî İşlem Birimi (CPU)
- Giriş Birimi
- Çıkış Birimidir

PICPLC'nin her bir alt biriminin genel özellikleri Çizelge 1'de sunulmuştur.

Çizelge 1: PICPLC alt birimlerinin genel özellikleri

| Alt Birim   | Genel Özellikler  |
|-------------|---|
| Girişler    | 16 giriş mevcuttur. Her biri optoisolator ile yalıtılmıştır. 5V ve 24V standardına uygundur. Ani temaslı ve sürgülü anahtar simülatörü mevcuttur. Her bir girişin durumu giriş ledlerinde kontrol edilmektedir. |
| Çıkışlar    | 16 röle çıkışı mevcuttur. Her bir rölenin kontak uçları klemensler ile bağlantıya hazır haldedir. Her bir çıkışın durumu çıkış ledlerinde kontrol edilebilmektedir.   |
| Programlama | PIC'e program aktarılması için kullanılmaktadır.  |
| İşlemci     | 20 Mhz frekansta çalışan PIC16F877A mevcuttur.  |
| Güç Birimi  | Kart üzerinde PIC için Regüleli 5V ve röleler için 12V DC gerilim kaynağı mevcuttur.  |

PIC16F877A kullanılan PICPLC'nin devre şeması Şekil 1'de görülmektedir [14]. PICPLC'nin programlanması, Şekil 2'de görülen temel yazılıma ait makine kodunu içeren dosya ve PIC'in üreticisinin sunmuş olduğu MPASM derleyicisi kullanılarak yapılabilmektedir. Kullanıcı klasik STL diline benzer olarak oluşturulmuş PICPLC makrolarını kullanarak program yazmakta ve ilgili derleme işlemleri ardından işlemciyi programlamaktadır. PICPLC'leri programlamada kullanılan ve merdiven (Ladder) diyagram işlemlerini yerine getiren Kontak ve Röle Temelli Makrolar, Filip-Flop Makrolar, Zamanlayıcılar, Sayıcılar, Karşılaştırma ve Aritmetik makrolardan bazıları Çizelge 2'de gösterilmektedir.



Şekil 1: PICPLC'nin Devre Şeması [14]

```

list           pic16f877a           ;işlemciyi tanımlamak için listede direktifli
#include <pic16f877a.inc>          ;işlemciye has değişken tanımlar:
#include <PICPLC_877.inc>         ;temel tanımlar ve makrolar

_CONFIG _CP_OFF & _WDT_ON & _BODEN_OFF & _PWRTE_ON & _HS_OSC & _LVP_OFF & _DEBUG_OFF & _CPD_OFF

main
org 0x00           ;Reset vektörü
initialize
scan

scan
    get_inputs      kullanıcı programı burada başlar -----
;1. Basamak
ld I0.0
TON_8 0,T1.1,.50
;2. Basamak
ld TON8_Q0
out Q0.0
;3. Basamak
ld I0.2
TOF_8 1,T1.1,.50
;4. Basamak
ld TOF8_Q1
out Q0.2
;----- kullanıcı programı burada bitir -----
send_outputs
goto scan
end
    
```

Şekil 2: PICPLC Temel Yazılım Dosyası

Çizelge 2: PICPLC'ye ait makrolar

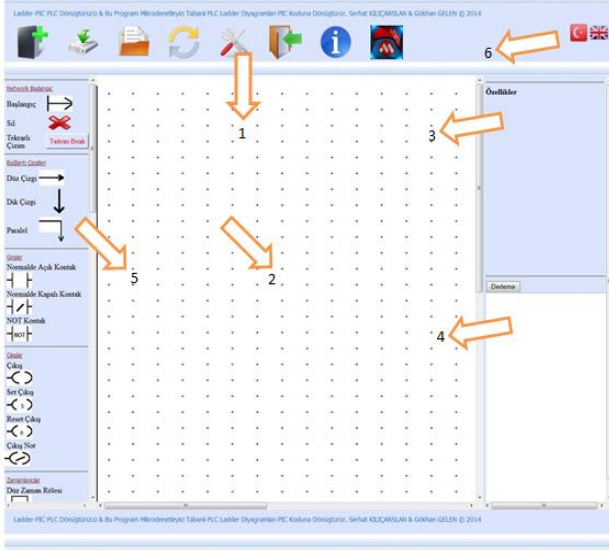
|                       | Makro Adı | PICPLC Assembly Örnek Kodu |
|-----------------------|-----------|----------------------------|
| Kontaklar             | LD        | ld I0.0                    |
|                       | LD_NOT    | ld_not I0.1                |
| Bobinler              | OUT       | out Q0.1                   |
|                       | OUT_NOT   | out_not Q0.2               |
| Zamanlayıcılar        | TON       | TON_8 0, T1.1, .50         |
|                       | TOF       | TOF_8 0, T1.1, .50         |
| Sayıcılar             | CTU       | CTU_8 0, I0.0, I0.1, .15   |
|                       | CTD       | CTD_8 0, I0.0, I0.1, .15   |
| Flip-Flop Makrosu     | r_edge    | r_edge 1                   |
|                       | f_edge    | f_edge 1                   |
| Karşılaştırma Makrosu | EQ        | R_EQ_K I1, 0Fh             |
|                       | GT        | R_GT_K I1, 0Fh             |
| Aritmetik Makrolar    | ADD       | RaddK Q1, .2, Q1           |
|                       | SUB       | RsubK Q1, .3, Q1           |

## 3. Geliştirilen Programlama Arayüzü

Geliştirilen arayüzle kullanıcıların merdiven (Ladder) diyagramı kodu çizip PICPLC için PIC kodu oluşturmasına olanak verilmiştir. Arayüzün tasarımında, kullanıcılara web uygulaması alanında günümüz teknolojilerinde ihtiyaç duyacakları birçok metot ve yöntemi sunan ASP.NET kullanılmıştır. Çalışmada .NET kullanılışının temel nedeni

diğer script dilleriyle yapılamayan bir çok program özelliklerinin rahatça kullanabilme avantajı sağlamasıdır.

Şekil 3’ de, tasarlanan programlama arayüzünün genel görüntüsü verilmiştir. Sayfa birçok bölümden oluşmaktadır. Hangi işlem yapılacaksa o işleme ait paneller aktif edilerek ilgili paneller üzerinde işlemler yapılabilir.



Şekil 3: PIC Temelli PLC Yazılımı Arayüzü

Yazılımda merdiven (Ladder) diyagramına ait programlamanın yapılabilmesi için ilk adım olarak çizim alanı oluşturulmalıdır. Çizim alanının daha belirgin olması için noktalarla alt alan oluşturularak kullanıcının rahatlıkla programlama yapması sağlanmıştır. Çizim alanına ait resim Şekil 3 de 2. madde olarak görülmektedir.

Geliştirilen yazılım çeşitli bileşenlerden meydana gelmektedir. Bu bileşenler Şekil 3 üzerinde rakamlarla gösterilmektedir ve aşağıda açıklanmaktadır.

1. **Menüler Bölümü:** Bu alanda bulunan ikonlar kullanılarak programa ait menülere ulaşılabilir. Burada yeni çizim alanı oluşturma, çizimi kaydetme, açma, derleme, donanım seçme ve yardım işlemlerini yerine getirecek ikonlar mevcuttur.
2. **Merdiven Diyagramı Çizim Bölümü:** Merdiven (Ladder) diyagramı çizimi için oluşturulan alandır. Bu alanda araç menüsünden alınan objeler tıklanarak çizim yapılmaktadır.
3. **Özellikler Bölümü:** Bu bölüm kullanılarak çizilen objelere değer atanabilmektedir.
4. **Derleme Bölümü:** Çizilen merdiven (Ladder) diyagramına ait PIC kodunu oluşturmak için kullanılacak bölümdür.
5. **Araçlar Bölümü:** Çizim yapılırken kullanılacak kontakların, zamanlayıcıların, sayıcıların vb. bileşenlere ulaşılacak bölümdür.
6. **Dil Ayarı Bölümü:** Geliştirilen arayüzün dil ayarlarını düzenlemek için kullanılan bölümdür. Türkçe ve İngilizce olmak üzere iki dil desteklenmektedir.

Program menüler bölümünden yeni çalışma alanı açılmasıyla aktif hale gelmektedir. Çalışma alanı oluşturulduktan sonra araçlar bölümünden hangi merdiven (Ladder) diyagramı bileşeni kullanılacaksa o bileşen seçilerek çalışma alanında istenilen yere tıklanarak yerleştirilir. Çizim alanındaki objenin üzerine tıklanarak özellikler bölümünden değer atanır. Örneğin bir sayıcı için, sayıcı etiketi, sayma değeri, reset girişi vb parametreler bu ekran kullanarak atanır.

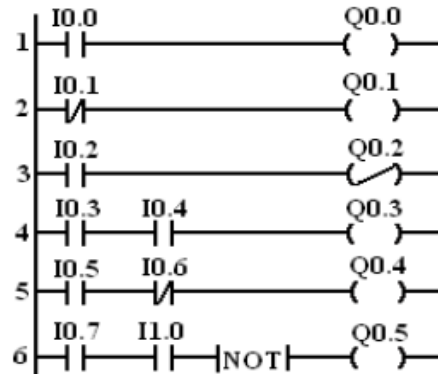
Çizim işlemi tamamlandıktan sonra derleme bölümündeki butona basılarak çizilen merdiven (Ladder) diyagramı PIC komutlarına dönüştürülür. Bu aşamadan sonra seçilen donanım uygun olan temel kod dosyası içerisine arayüzün elde ettiği PIC komutları yerleştirilir ve artık derlenmeye hazır olan \*.asm dosyası elde edilir. Bu işlemin ardından arayüze entegre edilen Microchip firmasının ücretsiz sunduğu MPASM derleyici programı çalıştırılarak \*.asm dosyası PIC’e yüklenilecek olan \*.hex dosyasına dönüştürülür. Artık kod PIC’e yüklenmeye hazır hale gelmiştir. Herhangi bir PIC programlayıcısı kullanılarak bu dosya PIC’e aktararak PICPLC için merdiven (Ladder) diyagram olarak çizilen PLC kodu gerçekleştirilmiş olur.

## 4. Uygulamalar

Bu kısımda tasarlanan arayüzün çalışmasını ve dönüşüm işleminin doğruluğunu test etmek amacıyla gerçekleştirilen uygulama örnekleri açıklanmaktadır. Örnek olarak ele alınan merdiven (Ladder) diyagramları arayüz ile PIC koduna dönüştürülmüş ve donanıma yüklenerek ilgili lojik fonksiyonu yerine getirip getirmediği denenerken test edilmiştir.

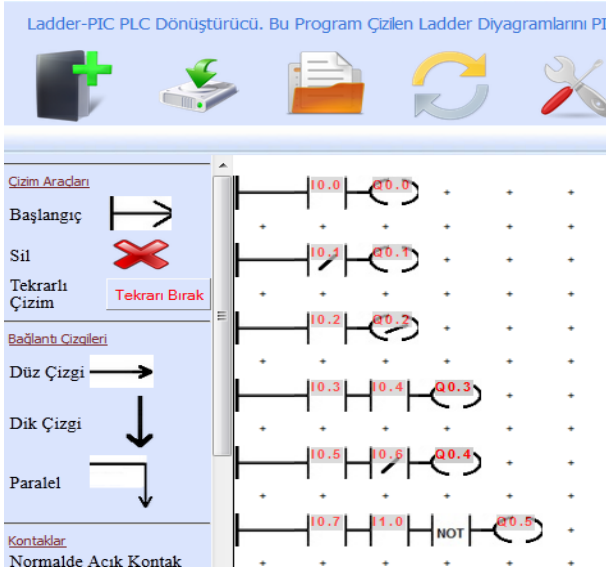
### 4.1. Kontak ve Röle Temelli Ladder Diyagramları

İlk olarak Şekil 4’te görülen ve temel merdiven(Ladder) diyagramı bileşenlerinden yani açık-kapalı kontak ve rölelerden oluşan kumanda devresi test edilmiştir.

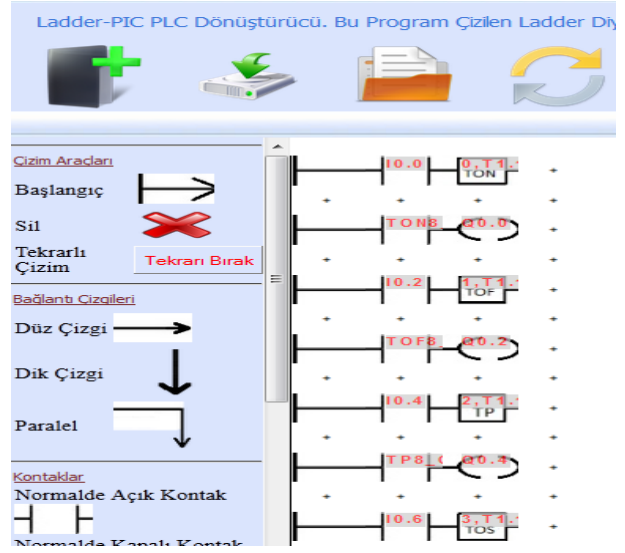


Şekil 4: Merdiven(Ladder) Diyagramı Örneği

Şekil 4’te görülen merdiven (Ladder) diyagramının arayüz çizim alanında oluşturulmuş görüntüsü Şekil 5’de görülmektedir. Oluşturulan bu merdiven (Ladder) diyagramının arayüz kullanılarak derlenmesinin ardından elde edilen PICPLC makro temelli kodları ve donanım seçimi ardından elde edilen .asm dosyası Şekil 6’da görülmektedir.



Şekil 5: PIC Temelli PLC Merdiven (Ladder) Diyagram



Şekil 7: PIC Temelli PLC Zamanlayıcı Merdiven (Ladder) Diyagram

```

Ilist p=I6F877A ;işlemciyi tanımlamak için listede direksiyon
#include <pl6F877A.inc> ;işlemciye has değişken tanımları
#include <PICPLC_877_161_160_tmmlr.inc> ;temel tanımlar ve makrolar
#include <PICPLC_877_161_160_kontak.inc> ;kontak ve röle temelli makrolar

_CONFIG _CP_OFF & _WDT_ON & _BODEN_OFF & _PWRTD_ON & _HS_OSC & _LVP_OFF & _DEBUG_OFF & _CPD_OFF

main
org 0x00 ;Reset vektörü

initialize

scan

get_inputs
;----- kullanıcı programı burada başlar -----
:1. Basamak
ld I0.0
out Q0.0

:2. Basamak
ld_not I0.1
out Q0.1

:3. Basamak
ld I0.2
out_not Q0.2

:4. Basamak
ld I0.3
and I0.4
out Q0.3

:5. Basamak
ld I0.5
and_not I0.6
out Q0.4

:6. Basamak
ld I0.7
nand I1.0
out Q0.5

;----- kullanıcı programı burada biter -----
send_outputs

goto scan
end
    
```

Şekil 6: Donanım seçimi ardından elde edilen .asm dosyası

Denemeler sonucunda girişlere uygulanan bilgiler ile merdiven diyagramda oluşması gereken çıkış bilgileri gözlemlenerek merdiven (Ladder) diyagramın doğru biçimde gerçekleştirildiği görülmüştür.

#### 4.2. Zamanlayıcı Uygulaması

Bu uygulamada zamanlayıcıları içeren örnek bir merdiven (Ladder) diyagramı dikkate alınmıştır. Şekil 7’de merdiven (Ladder) diyagramının arayüz çizim alanında oluşturulmuş görüntüsü sunulmuştur.

Oluşturulan bu merdiven (Ladder) diyagramının arayüz kullanılarak derlenmesinin ardından elde edilen PICPLC makro temelli kodları ve donanım seçimi ardından elde edilen .asm dosyası Şekil 8’de görülmektedir.

```

Ilist p=I6F877A ;işlemciyi tanımlamak için listede direksiyon
#include <pl6F877A.inc> ;işlemciye has değişken tanımları
#include <PICPLC_877_161_160_tmmlr.inc> ;temel tanımlar ve makrolar
#include <PICPLC_877_161_160_kontak.inc> ;kontak ve röle temelli makrolar

_CONFIG _CP_OFF & _WDT_ON & _BODEN_OFF & _PWRTD_ON & _HS_OSC & _LVP_OFF & _DEBUG_OFF & _CPD_OFF

main
org 0x00 ;Reset Vektörü

initialize

scan

get_inputs
;----- kullanıcı programı burada başlar -----
:1. Basamak
ld I0.0
TON_S 0,T1.1,.50

:2. Basamak
ld TON_Q0
out Q0.0

:3. Basamak
ld I0.2
TOF_S 1,T1.1,.50

:4. Basamak
ld TOFF_Q1
out Q0.2

:5. Basamak

:6. Basamak
ld I0.4
TP_S 2,T1.1,.50

:7. Basamak
ld I0.6
TON_S 3,T1.1,.50.50

:8. Basamak
ld TOSS_Q3
out Q0.6

;----- kullanıcı programı burada biter -----
send_outputs

goto scan
end
    
```

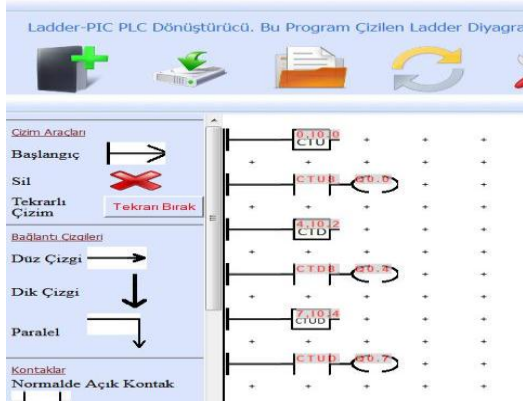
Şekil 8: Donanım seçimi ardından elde edilen .asm dosyası

Denemeler sonucunda girişlere uygulanan bilgiler ile merdiven (Ladder) diyagramda tanımlanan zamanlama fonksiyonlarının doğru biçimde gerçekleştirildiği görülmüştür.



### 4.3. Sayıcı Uygulaması

Son uygulama olarak sayıcıları içeren merdiven (Ladder) diyagramı örneği test edilmiştir. Şekil 9’da bu merdiven (Ladder) diyagramının arayüz çizim alanında oluşturulmuş görüntüsü görülmektedir.



Şekil 9: PIC Temelli PLC Sayıcı Merdiven Diyagram

Oluşturulan bu merdiven (Ladder) diyagramının arayüz kullanılarak derlenmesinin ardından elde edilen PICPLC makro temelli kodları ve donanım seçimi ardından elde edilen .asm dosyası Şekil 10’da görülmektedir.

```

lisr      p=16F877A                               ;işlemciyi tanımlamak için liste direkrifi
#include <pic16f877a.inc>                          ;işlemciye has değişken tanımları
#include <PICPLC_877_161_160_cmlr.inc> ;temel tanımlar ve makrolar
#include <PICPLC_877_161_160_kontak.inc>;kontakt ve röle temelli makrolar

_CONFIG CP_OFF & _WDT_ON & _BOREN_OFF & _PWRT_ON & _HS_OSC & _LVP_OFF & _REBUS_OFF & _CPD_OFF

org      0x000                                ;Reset Vektörü
main
    initialize
scan
    get_inputs
;----- Kullanıcı programı burada başlar -----
;1. Basamak
CTUD_8 0,10,0,10,1,,15

;2. Basamak
ld CTUD_Q0
out Q0:0

;3. Basamak
CTUD_8 4,10,2,10,3,,10

;4. Basamak
ld CTUD_Q4
out Q0:4

;5. Basamak
CTUD_8 7,10,4,10,5,10,6,,10,7,20

;6. Basamak
ld CTUD_Q7
out Q0:7

;----- Kullanıcı programı burada biter -----
send_outputs
goto scan
end
    
```

Şekil 10: Donanım seçimi ardından elde edilen .asm dosyası

Denemeler sonucunda girişlere uygulanan bilgiler ile merdiven (Ladder) diyagramda tanımlanan sayma fonksiyonlarının doğru biçimde gerçekleştirildiği görülmüştür.

Tüm uygulamalarda donanım seçimi ardından elde edilen .asm dosyası arayüz üzerinden MPASM programı ile derlenmiş ve elde edilen kaynak kodu PICPLC’ye yüklenerek denemeler yapılmıştır. Tasarlanan arayüz aritmetik işlem vb. gibi daha kompleks merdiven diyagramı örnekleri kullanılarak ta test

edilmiştir. Bu örneklerde de arayüzün sorunsuz çalıştığı gözlemlenmiştir.

## 5. Sonuçlar

Bu çalışmada, ticari PLC’lere ucuz maliyeti ile alternatif olan PIC mikrodenetleyicisi temelli bir PLC için programlama arayüzü gerçekleştirilmiştir. Tasarlanan arayüz web sayfaları altında çalışabilir nitelikte olduğundan farklı işletim sistemleri yüklü bilgisayarlarda çalışabilecek niteliktedir. Bu açıdan platformdan bağımsızdır denilebilir. Geliştirilen arayüz ile merdiven diyagramı olarak çizilen PLC kodu ilgili mikrodenetleyicinin koduna dönüştürülmektedir. Arayüz çeşitli PLC kod örnekleri kullanılarak test edilmiştir. Testler sonucunda arayüzün sorunsuz çalıştığı görülmüştür. Gerçekleştirilen arayüzün kullanıcı dostu olması için işlemler gayet basit olarak düzenlenmiştir.

## 6. Kaynaklar

- [1]. Wikipedia, <http://tr.wikipedia.org/wiki/PLC>, 2014
- [2]. Karwoski Chuck, “How to lower PLC software costs”, Control Engineering, 2013
- [3]. Harmanda, A., 2011. 16 Bitlik Bir PIC Mikrodenetleyicisi ile Bir Programlanabilir Lojik Denetleyici Tasarımı ve Uygulaması.(Yüksek Lisans Tezi), Niğde Üniversitesi. Fen Bilimleri Enstitüsü, Niğde.
- [4]. Rafat, Ö.F., 2010. PIC16F877 Mikrodenetleyicisi ile Bir PLC Tasarımı.(Yüksek Lisans Tezi), Selçuk Üniversitesi. Fen Bilimleri Enstitüsü, Konya.
- [5]. Tongur, V., 2008, Atmega128 tabanlı PLC tasarımı.(Yüksek Lisans Tezi), Selçuk Üniversitesi. Fen Bilimleri Enstitüsü, Konya.
- [6]. Erkol H., 2008. Mikrodenetleyici tabanlı PLC donanımı ve yazılımının gerçekleştirilmesi. (Yüksek Lisans Tezi), Erciyes Üniversitesi. Fen Bilimleri Enstitüsü, Kayseri.
- [7]. Kitiş, Ş., 2007. PIC16F84 Mikrodenetleyicisi ile bir programlanabilir lojik denetleyici tasarımı ve uygulaması.(Yüksek Lisans Tezi), Niğde Üniversitesi. Fen Bilimleri Enstitüsü, Niğde.
- [8]. Başkan, M. ve Gazi, M.E., 2012. PIC 16F877A Mikro denetleyicisinin PLC olarak kullanılması PIC LDR Programlama. Proje Tabanlı Mekatronik Eğitim Çalıştayı, 25-27 Mayıs, 2012, Çankırı Karatekin Üniversitesi Meslek Yüksekokulu. Çankırı.
- [9]. Mun, N., and Haron, Z. A., 2010. Graphical microcontroller programming tool based on extended S-System Petri net. In Proceedings of the 4th WSEAS international conference on Computer engineering and applications. World Scientific and Engineering Academy and Society (WSEAS).
- [10]. Al Mashhadany, Y.I., 2012. Design and Implement of a Programmable Logic Controller (PLC) for Classical Control Laboratory. Intelligent Control & Automation, (2153-0653), 3(1).
- [11]. Mikroelektronica PICPLC16, <http://www.mikroe.com/picplc16/>, 2014
- [12]. Uzam M., “PIC 16F877 A Temelli PLC”, Birsan Yayınevi, İstanbul,
- [13]. Uzam M., “Building a Programmable Logic Controller with PIC16F648A Microcontroller” Taylor and Francis, August 2013.
- [14]. Uzam M., “PIC16F877A Mikro-Denetleyicisi Temelli PIC PLC Donanım”, Endüstri & Otomasyon, sayı 153, sayfa 22-24, Aralık 2009.