

# HPFBU 2014

## LabVIEW: Tanıtım

Görkem Türemen  
Ankara Üniversitesi

03/02/14

# LabVIEW: Tanıtım

## -İçerik-

- » LabVIEW,
  - » Nedir?
  - » Kullanım Alanları
  - » Nasıl Çalışır?
    - » Veri Akışı Programlama
  - » Program Çevresi
    - » VI
    - » Programlama Ortamı
      - » Ön Pencere ve Akış Çizelgesi
      - » Tahtalar
      - » Veri Türleri
      - » Uçbirim Simgeleri
      - » Teller
      - » Yapılar
      - » Dizi ve Kümeler
      - » Kaydırma Yazgacı
      - » Çizimler
      - » Alt VI & Çabuk VI
  - » Örnek VI
  - » Bilgisayar ile Bağlantı Yolları
  - » Donanım Sürücüleri

# LabVIEW: Tanıtım

## - Terimler ve Kısaltmalar -

- » LabVIEW: Laboratory Virtual Instrument Engineering Workbench
- » NI: National Instruments
- » VI: Virtual Instrument
- » GPIB (IEEE-488): General Purpose Interface Bus
- » LAN: Local Area Network
- » PCI: Peripheral Component Interconnect
- » VISA: The Virtual Instrument Software Architecture
- » IVI: Interchangeable Virtual Instrumentation

# LabVIEW: Tanıtım

## -Sözlük-

- » Data flow programming: Veri akışı programlama
- » Virtual instrument: Sanal donanım
- » Front panel: Ön pencere
- » Block diagram: Akış çizelgesi
- » Control: Yönlendirme
- » Indicator: Gösterge
- » Palette: Tahta
- » Terminal: Uçbirim
- » Wire: Tel
- » Structure: Yapı
- » Array: Dizi
- » Cluster: Küme
- » Shift register: Kaydırma yazgacı
- » Graph: Çizim
- » Express: Çabuk
- » Node: Düğüm
- » String: Dizgi
- » Dynamic: Devinimsel

# LabVIEW

## -Nedir?-

- » NI firması tarafından geliştirilen bir görsel programlama dili/ortamı/yazılımı 'dır.
- » Kullanılacak herhangi bir donanımın (Ör: Osiloskop, Güç Kaynağı, Algıç, vb.) yada donanımlar topluluğunun yönlendirilmesinin oluşturulan bir akış çizelgesi içinde bilgisayar destekli olarak yapılmasını sağlar.
  - » Kısaca fiziksel bir donanımı programlanabilir bir sanal donanıma (VI) çevirir.
- » İlk olarak 1986'da Apple'ın MAC'i için geliştirildi.
- » Kullandığı görsel programlama diline "G" deniliyor.
- » Mac OS X, Linux/UNIX ve MS Windows'da çalışabiliyor.
- » Ücret: 2.400 - 13.100 TL arasında değişiyor.

# LabVIEW

## -Kullanım Alanları-

» Üç ana iş için kullanılabilir:

- » Veri toplama
- » Veri işleme (Çözümleme)
- » Donanım yönlendirme



» Kullanıldığı büyük ölçekli tesisler:

- » Parçacık hızlandırıcıları (CERN, ISIS, SPring8, LANL, LNLS)
- » Reaktörler (ITER)
- » Teleskoplar (JWST, E-ELT)
- » ...

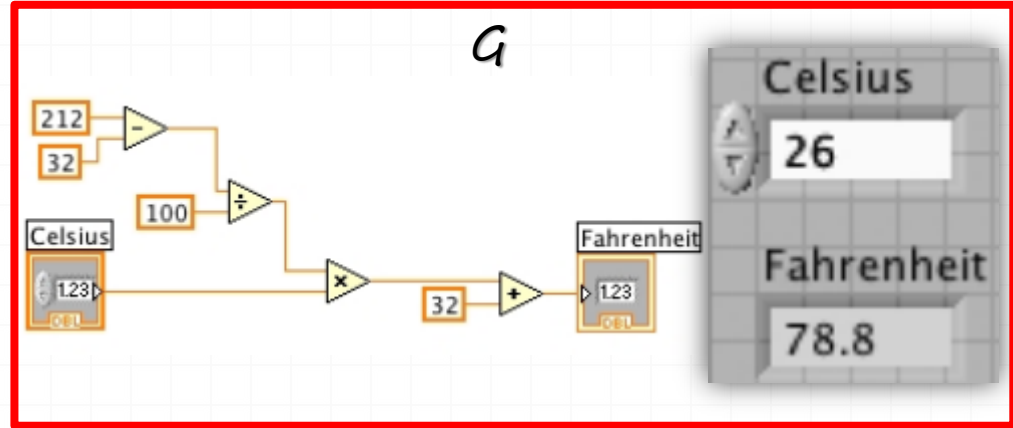
# LabVIEW

## -Nasıl Çalışır-

- » Görsel programlama yöntemi sayesinde C/C++, Fortran, vb. programlama dillerinindeki komutlar yerine simgeler ile program yazılır.
- » Metin tabanlı dillere kıyasla çok daha kısa sürede programlama yapılabilir.

```
float celsius(float fahrenheit);  
  
main()  
{  
    float DegreeF=76;  
    float DegreeC=celsius(DegreeF);  
    printf("The Temperature is %f F and %f C\n", DegreeF, DegreeC);  
}  
  
float celsius(float fahrenheit)  
{  
    float celsius = (5.0/9.0)*(fahrenheit-32);  
    return celsius;  
}
```

C

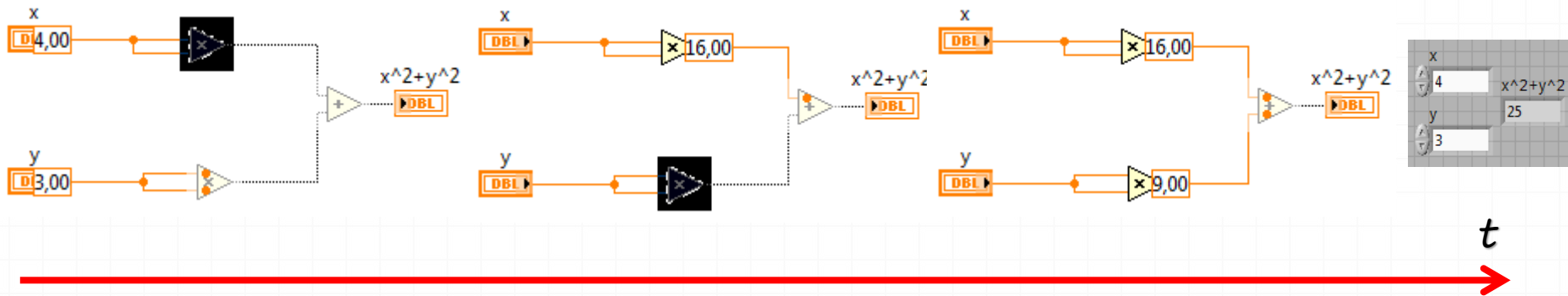




# Veri Akışı Programlama

## -Nedir?-

- » Programın çalışma sırasını komutların belirlediği metin tabanlı programlama dillerinin tersine, veri akışı programlamada çalışma sırasını akış çizelgesi üzerindeki yollardan verinin akma sırası belirler.
- » Akış çizelgesindeki herhangi bir bileşenin çalıştırılması için, gereken tüm girdi verilerinin o bileşene ulaşmış olması şarttır.
- » Bu sayede çoklu işlemler eşzamanlı olarak gerçekleştirilebilir.





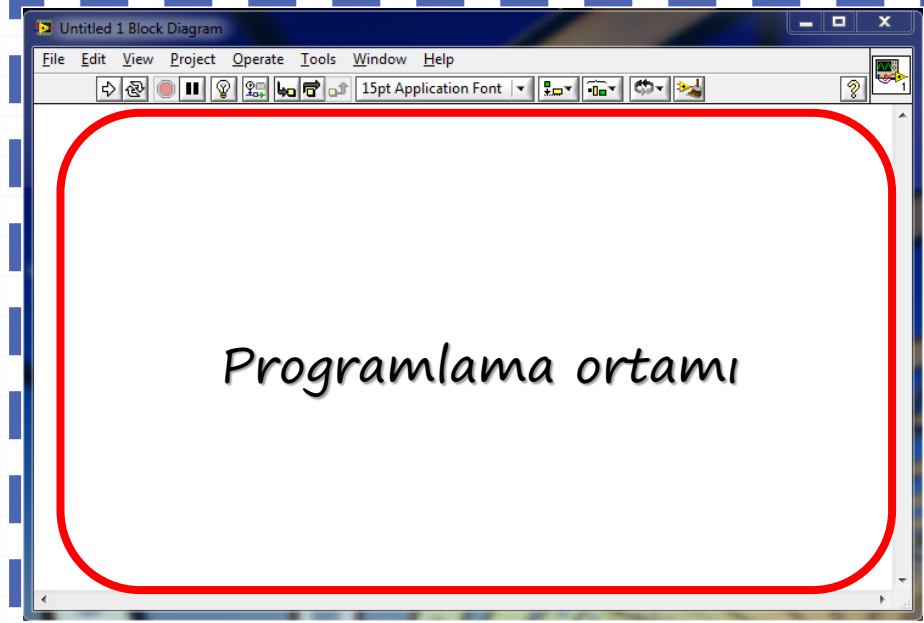
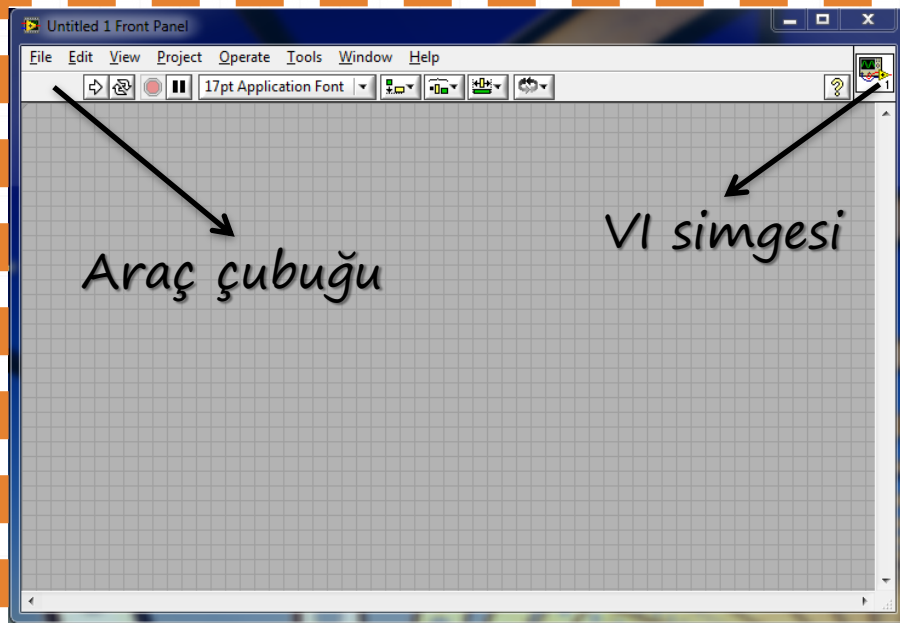
# Program Çevresi



# Program Çevresi

-VI-

- » LabVIEW programlarının her birine VI denir.
- » Ön pencere ve akış çizelgesi olmak üzere iki kısımdan oluşur.
- » Her bir programın kütük uzantısı → \*.vi



# VI

## - Araç Çubuğu -

Çizelge temizleyici










içerik yardımı



Hata ayıklama araçları

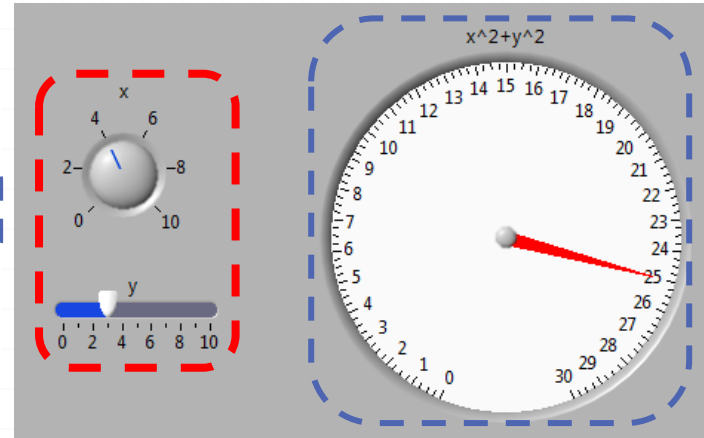
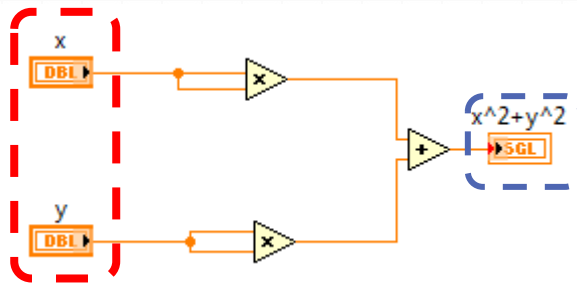
Hizalama araçları

-  VI'nın çalıştırılmaya uygun olduğunu gösterir ve tıklandığında çalışmasını sağlar.
-  VI'nın çalışmakta olduğunu gösterir.
-  VI'da bir hata olduğunu ve çalıştırılmaya uygun olmadığını gösterir.
-  VI'nın durdurulmasını sağlar. Son çare olarak kullanılır !!!
-  VI'nın duraklatılmasını sağlar. Duraklatılan konumdan devam edilebilir.
-  VI'ın durdur yada duraklata tıklanmadıkça tekrar tekrar çalışmasını sağlar.
-  VI'nın çalışması sırasında verinin akışını hareketli olarak gösterir. (Sadece akış çizelgesinde kullanılabilir.)

# Programlama Ortamı

## -Ön Pencere-

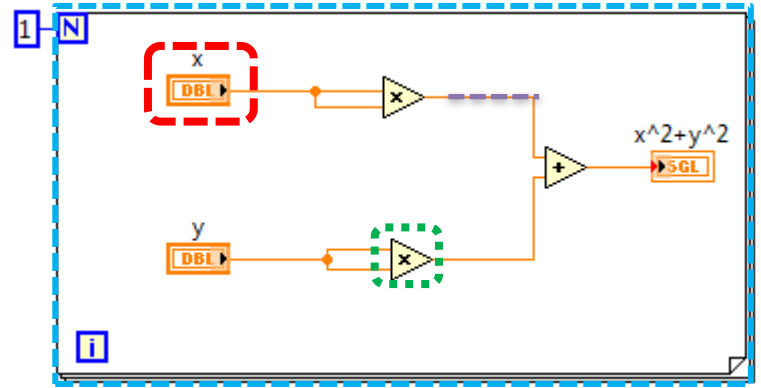
- » Ön pencere, akış çizelgesinde oluşturulan VI'nın kullanıcı arayüzlerini içerir.
- » **Yönlendirme:** VI'ya verilecek girdileri yönlendirmeye/ayarlamaya yararlar. Ör: Çevirmeli düğme, basmalı düğme, kadran, vb...
- » **Göstergeler:** VI'dan alınacak çıktıları görüntülemeye/yazdırmaya yararlar. Ör: Çizim, LED gösterge, sayaç, vb...



# Programlama Ortamı

## - Akış Çizelgesi -

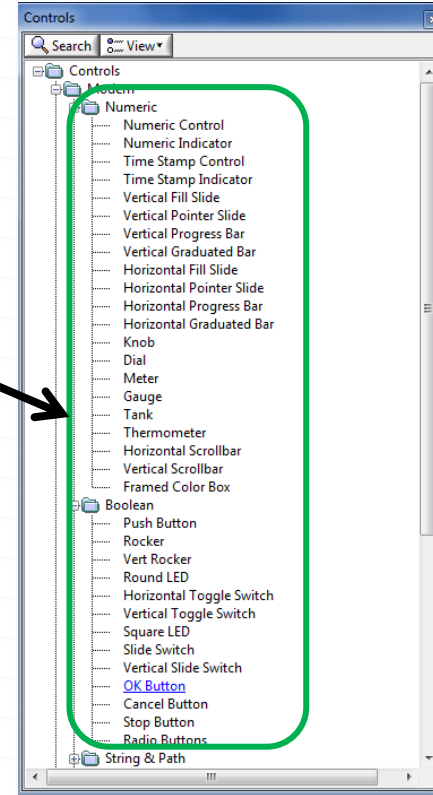
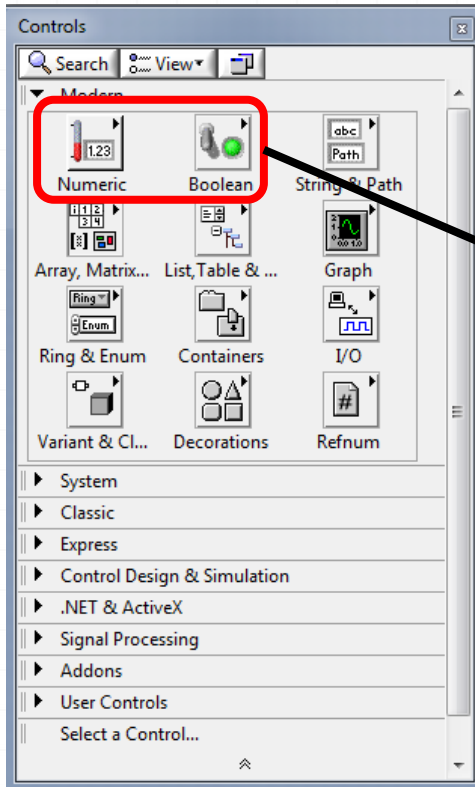
- » Akış çizelgesi ön pencerenin altında yatan program yapısını içerir ve dört ana bileşenden oluşur:
  - » **Uçbirim Simgeleri:** Ön penceredeki yönlendirme ve göstergelere doğrudan bağlıdır. Ön pencere ile akış çizelgesi arasındaki veri alışverişini sağlarlar.
  - » **Düğümler:** Belirli bir işlevi yapmak üzere girdi ve çıktı bağlantı noktaları olan nesnelerdir.
  - » **Yapılar:** Koşullar ve döngüler ile VI'nın akışını belirlerler.
  - » **Teller:** Akış çizelgesinde verinin akmasını sağlarlar.



# Tahtalar

## - Yönlendirme Tahtası -

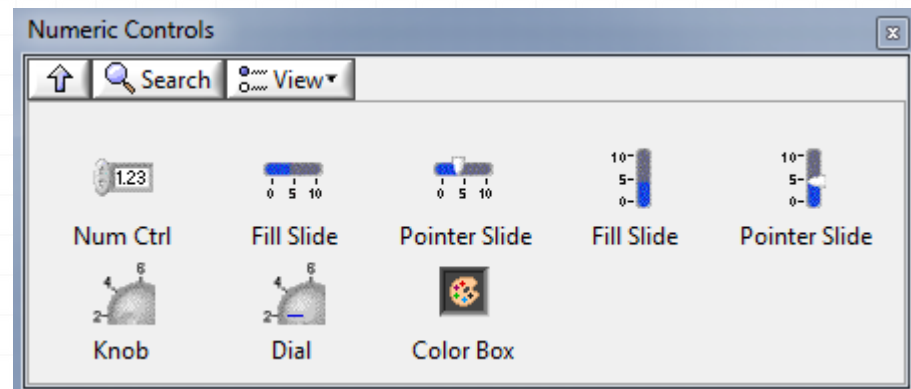
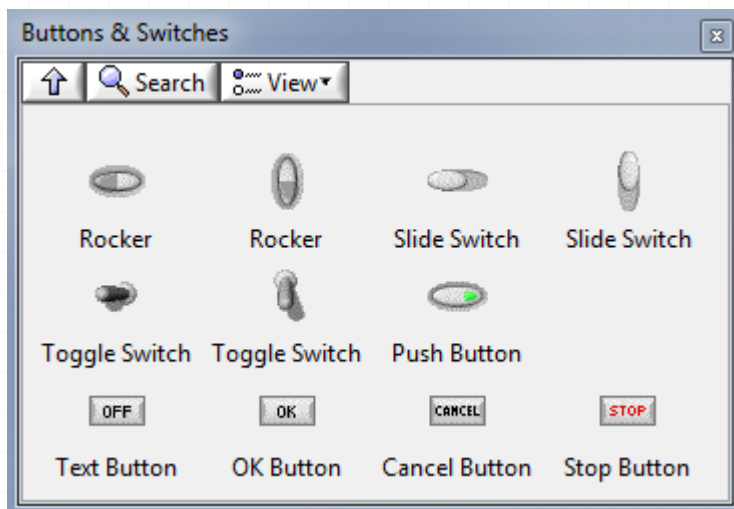
- » VI'da kullanılacak yönlendirme ve göstergeleri barındırır.
- » Ön pencere üzerinde sağ tıklanarak ulaşılabilir.





## - Yönlendirme Alt Tahtası -

» VI'nın girdilerini ayarlamak için kullanılacak yönlendirmeler:

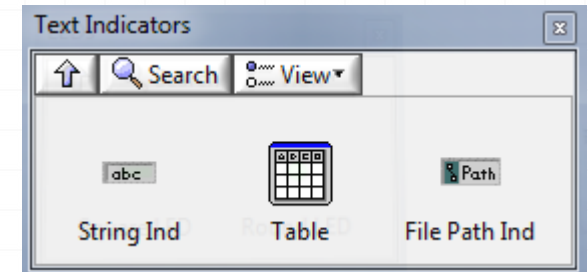
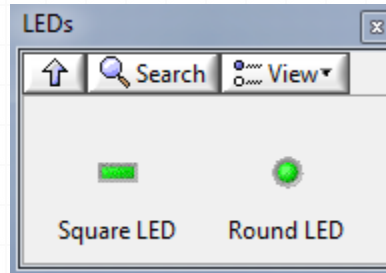
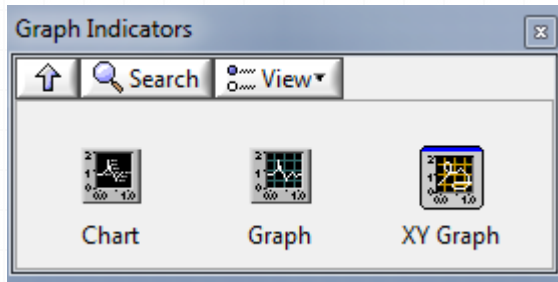
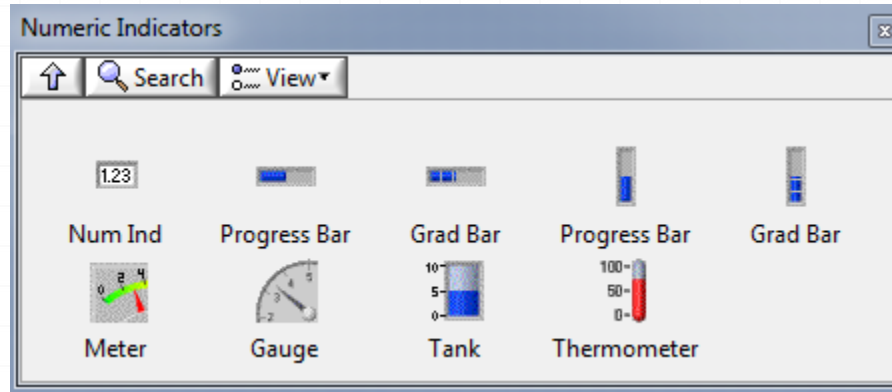




# Tahtalar

## - Göstergeler Alt Tahtası -

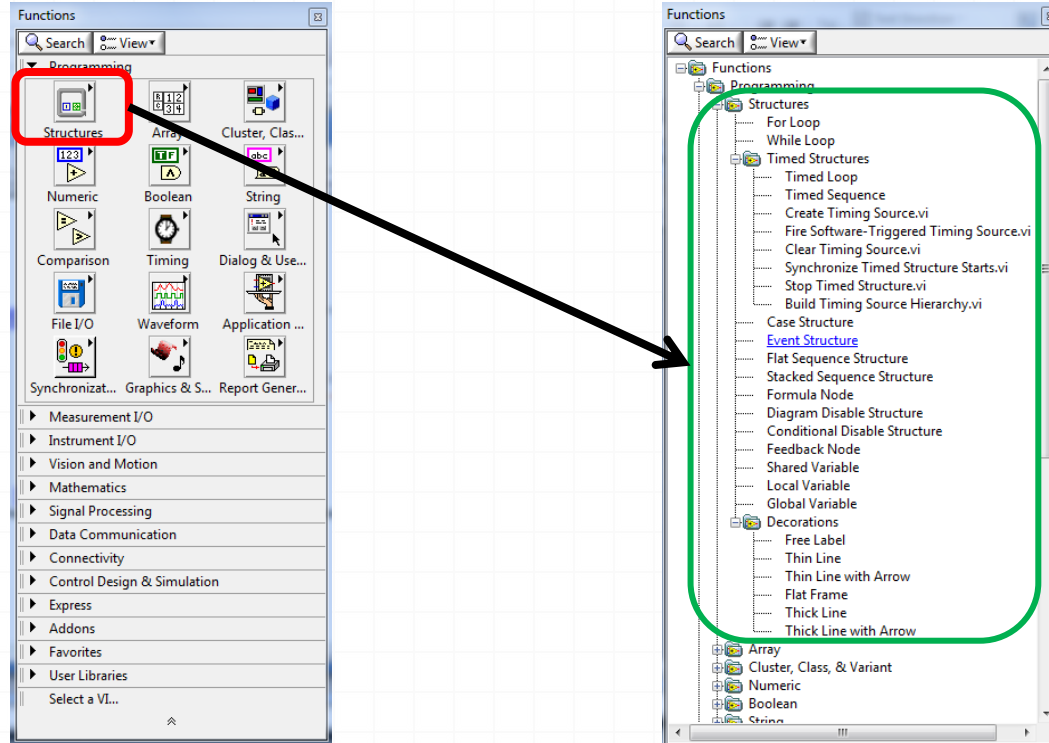
» VI'nın çıktılarını ayarlamak için kullanılacak göstergeler:



# Tahtalar

## - İşlevler Tahtası -











- » VI yapısını oluştururken gerekli olan programlama araçlarını barındırır.
- » Akış çizelgesi üzerinde sağ tıklanarak ulaşılabilir.

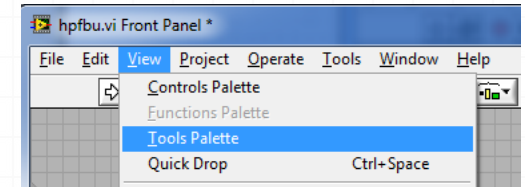
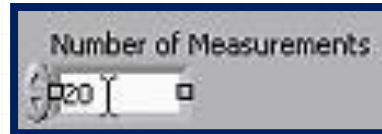
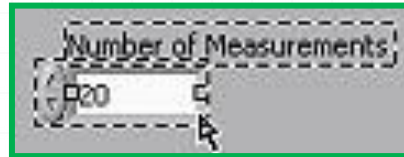
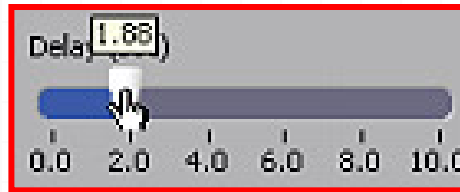


# Tahtalar

## - Araçlar Tahtası -

>> VI oluştururken fare imlecinin hangi görev için kullanılacağını ayarlamak için kullanılır.

-  İşlem aracı
-  Konumlandırma aracı
-  Etiketleme aracı
-  Tel aracı
-  Kısayol menüsü aracı
-  Kaydırma aracı
-  Durma noktası aracı
-  Sonda aracı
-  Renk kopyalama aracı
-  Renklendirme aracı








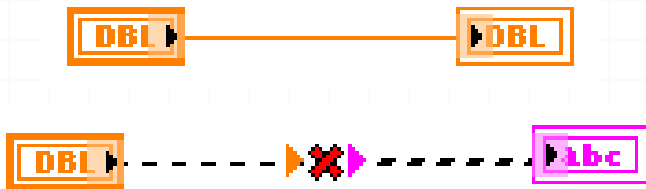
Kendiliğinden seçim



# Programlama Ortamı

## -Veri Türleri-

	Ondalıklı sayı	0.04, 1.008, vb. - 64 bit (~15 hane)
	Tam sayı	2, 6, 11, 321, vb. - 32 bit (-2147483648 - 2147483647)
	Dizgi	doğru, hayır, A, vb.
	Devinimsel	gerekli olan veri türüne kendini dönüştürebilir (Çabuk VI).
	Mantıksal	Doğru, Yanlış







You have connected two terminals of different types.  
The type of the source is double [64-bit real (~15 digit precision)].  
The type of the sink is string.

# Programlama Ortamı

## -Uçbirim Simgeleri-

- » Ön penceredeki bileşenler akış çizelgesinde uçbirim simgeleri olarak gözükürler.
- » Girdi ve çıktıları için bağlantı noktalarını ve veri türünü gösterirler.










	Yönlendirme	Gösterge
Uçbirim simgeleri		
Veri türü uçbirim simgeleri		



# Programlama Ortamı

## -Teller-

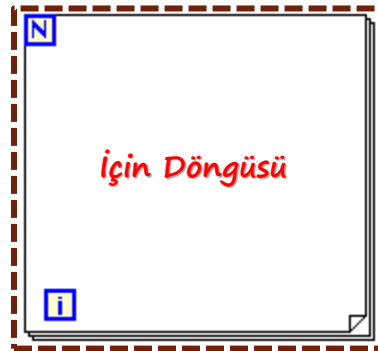
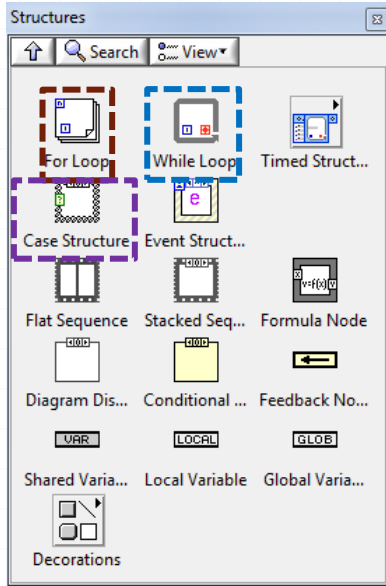
- » Akış çizelgesindeki iki bileşen arasında veri alış-verişi sağlar.
- » Tellerin rengi ve kalınlığı veri türleri ile ilgili bilgi verir.

Tel Türü	Sayı	1B Dizi	2B Dizi	Renk
Sayısal				Turuncu (Ondalıkli) Mavi (Tamsayı)
Mantıksal				Yeşil
Dizgi				Pembe

# Programlama Ortamı

## - Yapılar -

- » Metin tabanlı programlama dillerindeki yapılara oldukça benzerdir.
- » Akış çizelgesinde yapılar görsel olarak boyutu ayarlanabilir çerçeveler ile gösterilirler.
  - » Çerçevenin içine aldığı bileşenler yapı türüne göre tekrarlanabilir yada koşula bağlanabilir.

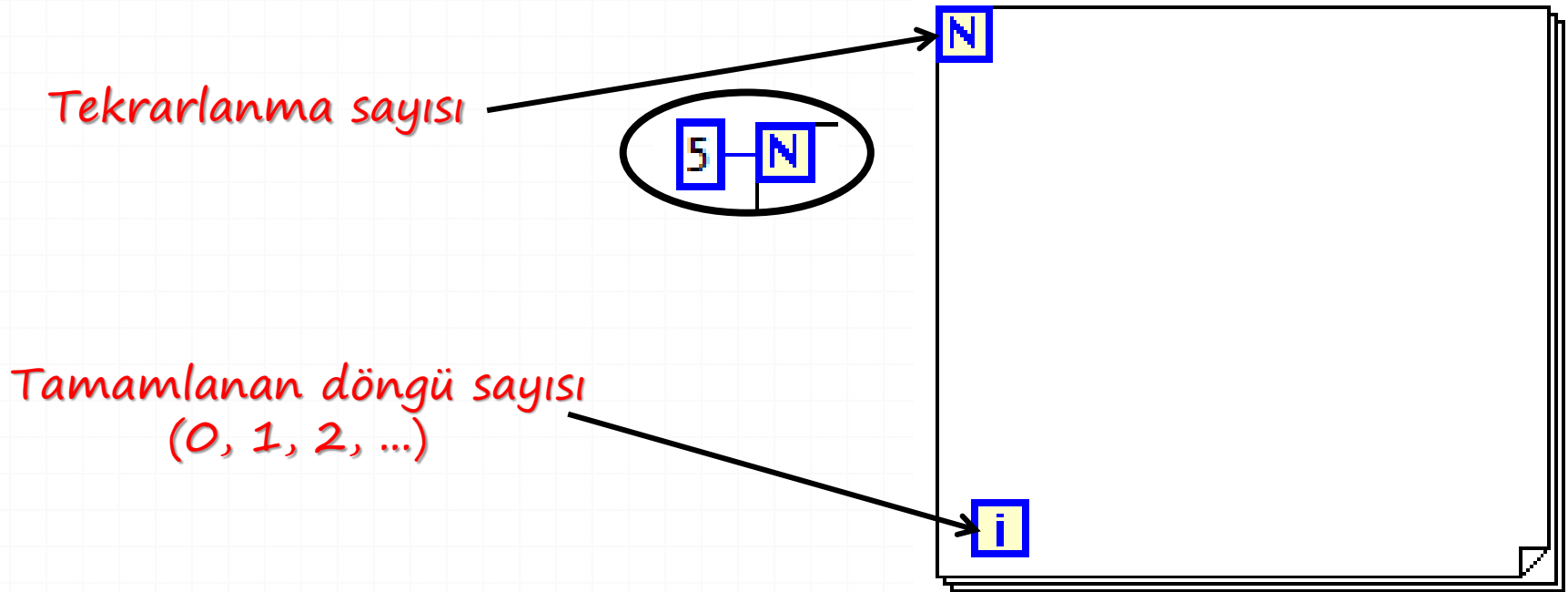




# Yapılar

## -İçin Döngüsü-

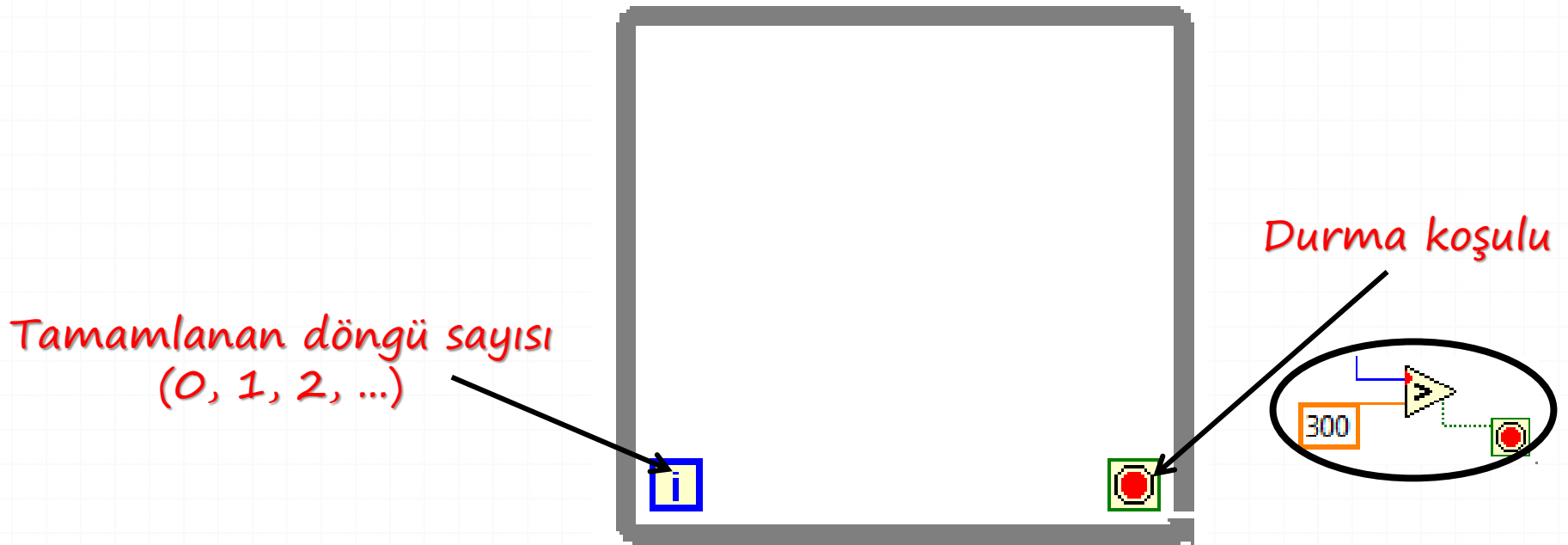
» Bu döngü VI içindeki bir veya birden fazla işlevin belirtilen sayıda tekrarlanması için kullanılır.



# Yapılar

## - İken Döngüsü -

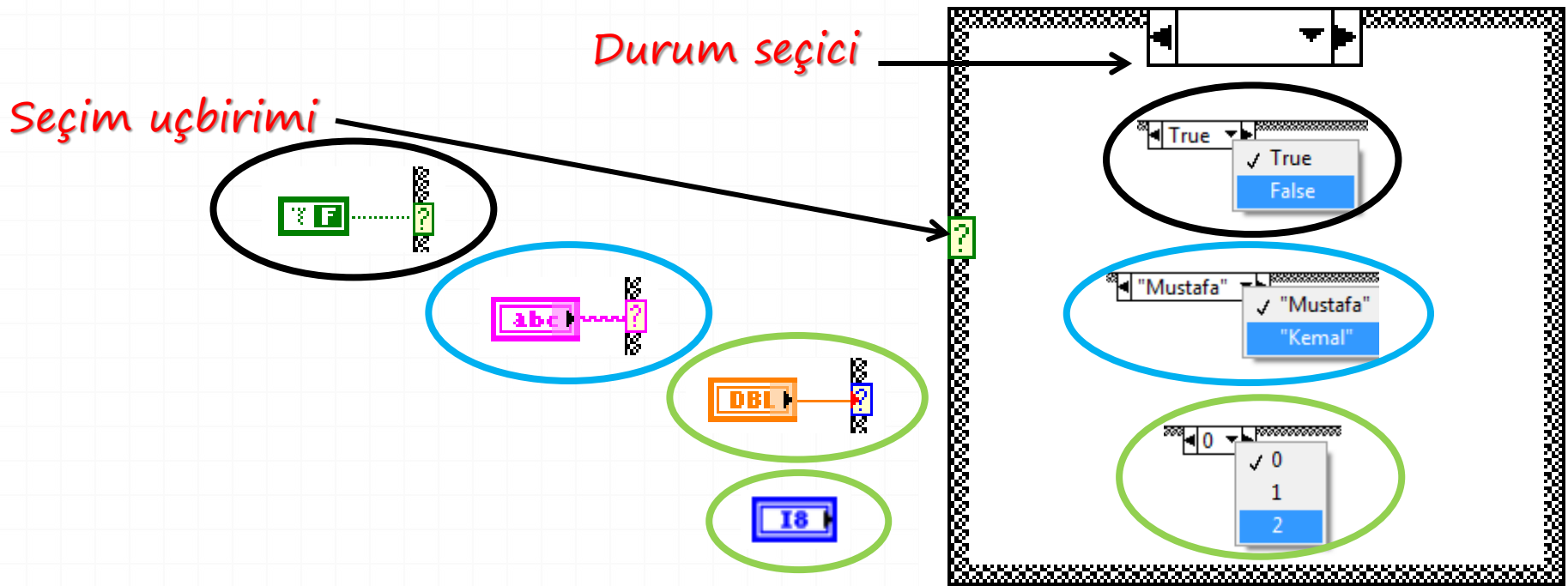
» Bu döngü VI içindeki bir veya birden fazla işlevin belirli bir koşul altında tekrarlanması için kullanılır.



# Yapılar

## -Durum Yapısı-

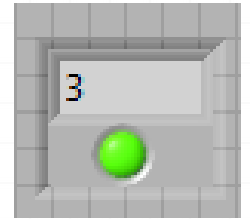
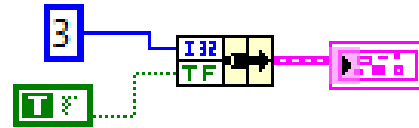
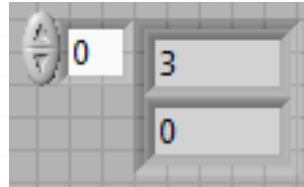
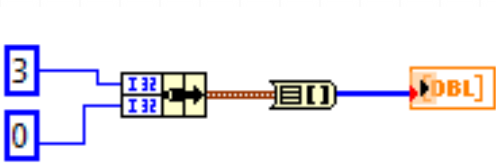
>> Bu yapı VI içindeki bir veya birden fazla işlevin farklı durumlar altında nasıl çalışacağını belirlemek için kullanılır.



# Programlama Ortamı

## -Dizi ve Kümeler-

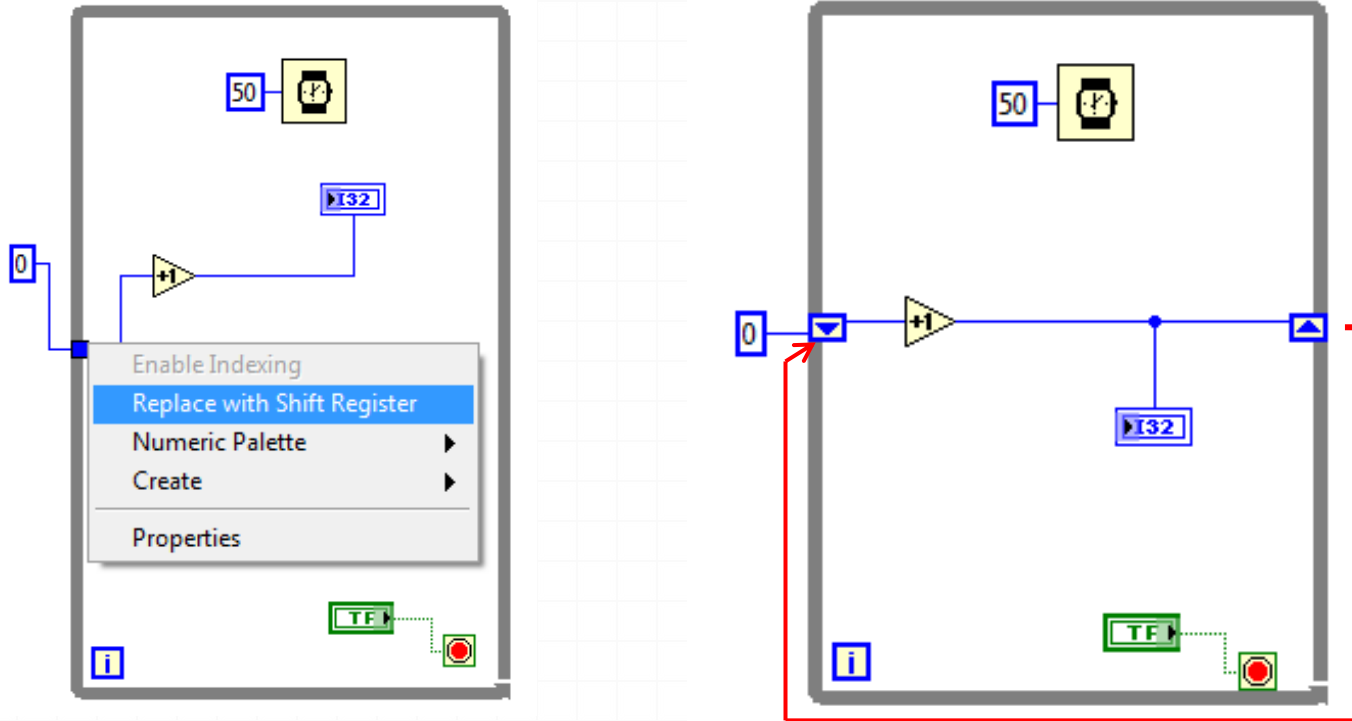
- » Metin tabanlı programlama dillerinde olduğu gibi diziler aynı veri türündeki bir öbek verinin toplandığı bileşenlerdir.
  - » Ör: Bir rastgele sayı üreticinden gelen verilerin çıktısını aynı veri türünden oluştukları için bir dizi olarak depolamak mantıklıdır.
- » Kümeler ise (Metin tabanlı programlama dillerindeki sınıflar gibi) farklı veri türündeki verilerin toplanabildiği bileşenlerdir.
  - » Akış çizelgesindeki tel sayısını azaltmasının yanı sıra bir çok farklı veri türündeki verinin birleştirilip tek uçbirime girdi olarak verilmesini sağlar.



# Programlama Ortamı

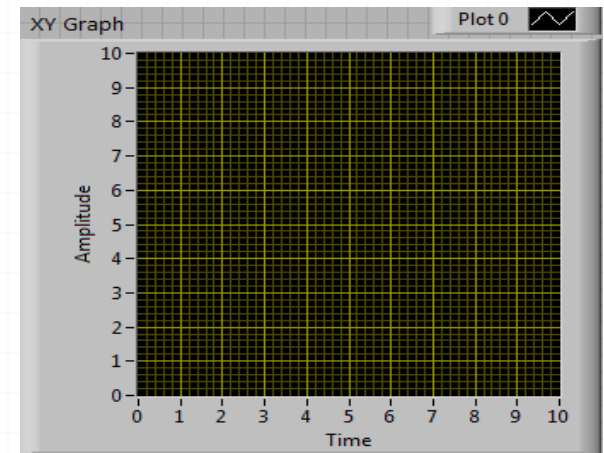
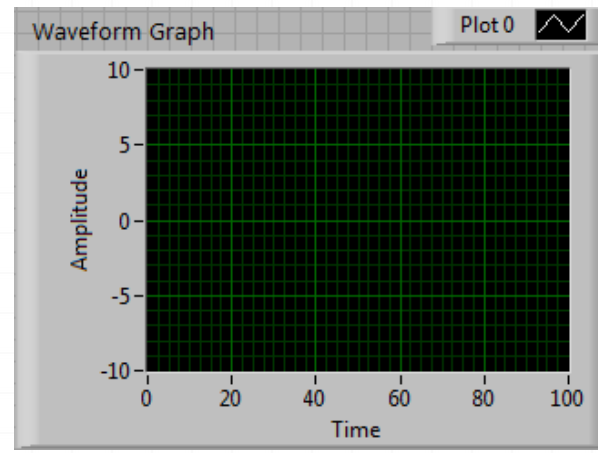
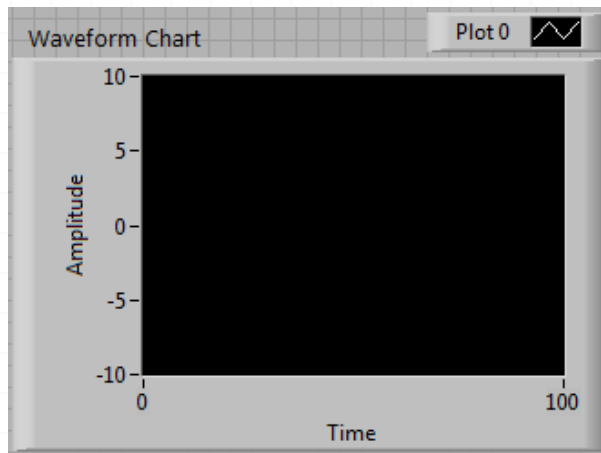
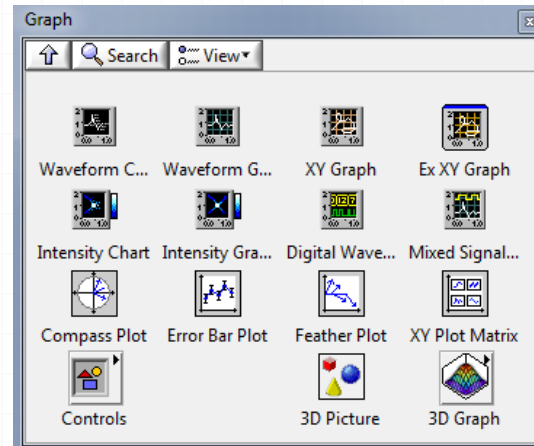
## - Kaydırma Yazmacı -

» Döngüler ile çalışırken bir önceki çevrimdeki verileri kaydederek yeni çevrim için kullanılabilir duruma getirir.



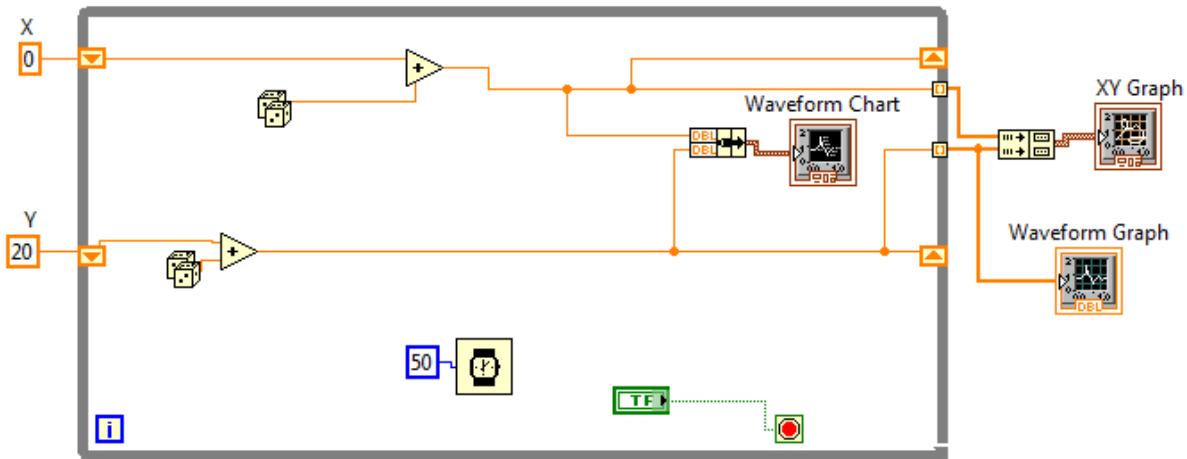
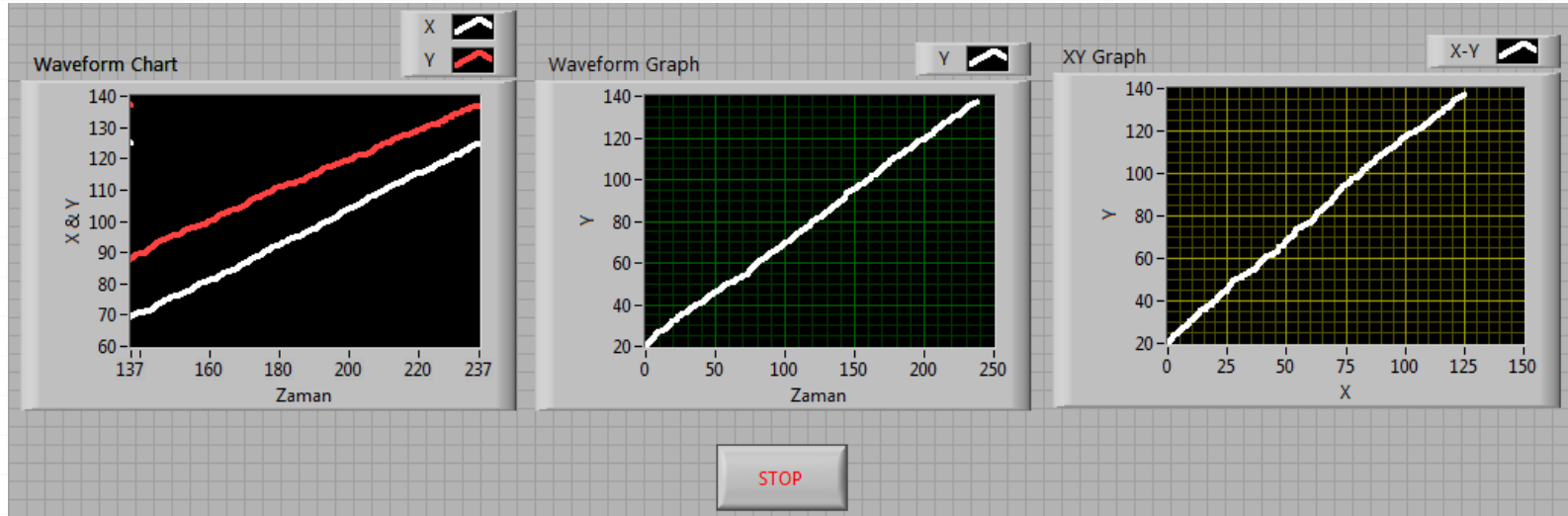
# Programlama Ortamı

## -Çizimler-



# Programlama Ortamı

## -Çizimler (Örnek)-

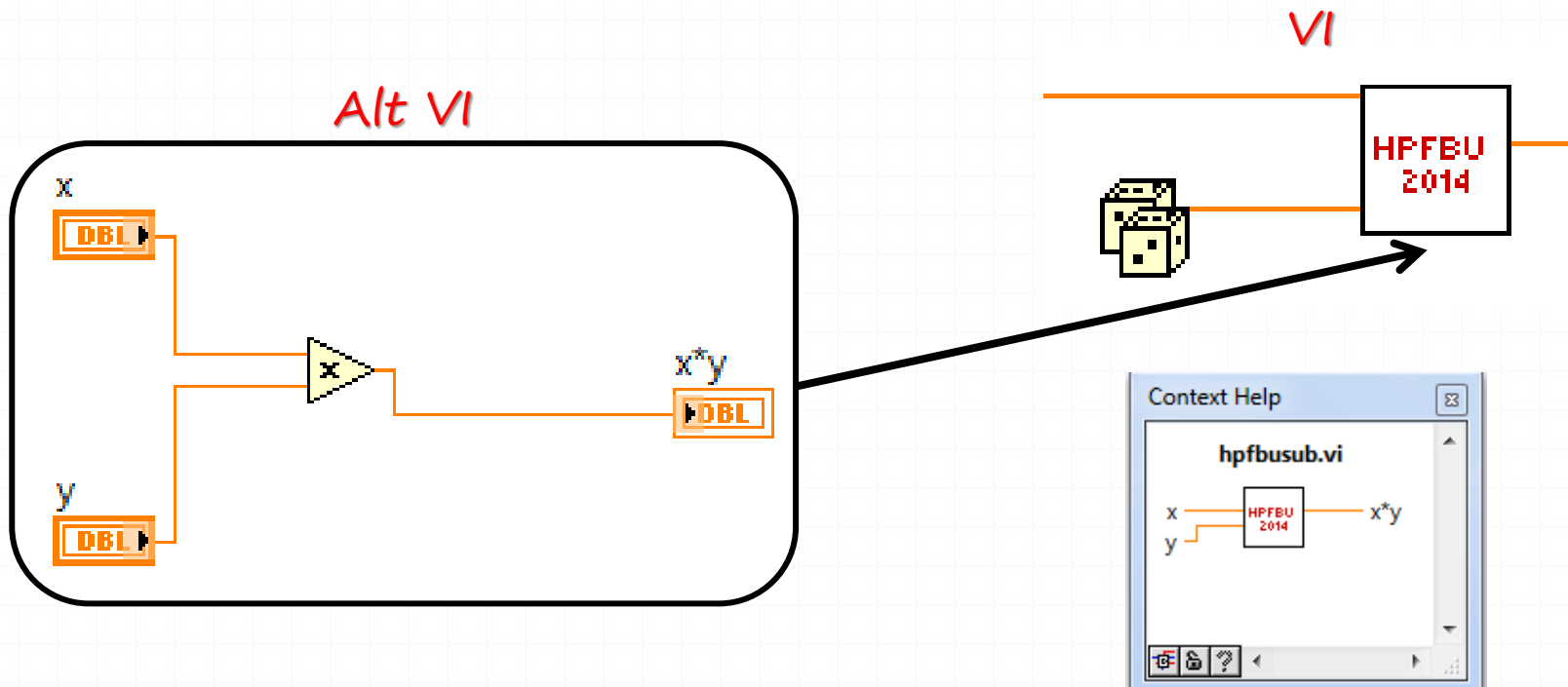




# Programlama Ortamı

-Alt VI-

>> Eğer bir VI bir başka VI'in içinde kullanılacak olursa "alt VI" olarak adlandırılır.



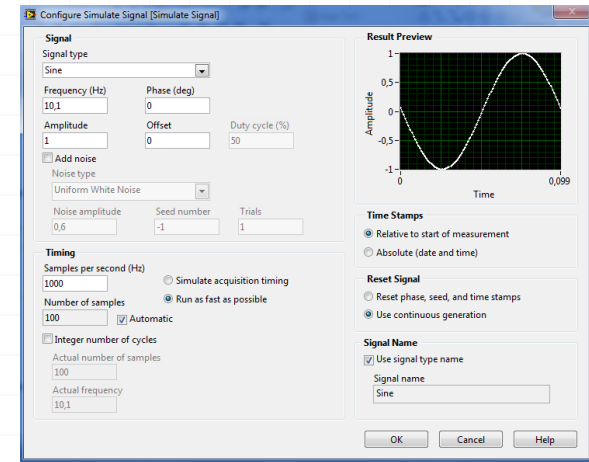
# Programlama Ortamı

## -Çabuk VI-

- » Program ile hazır gelen ve kullanıcıya hızlı çözümler sağlayan özel VI'lardır.
- » Akış çizelgesine eklendiğinde bir yapılandırma penceresi açılarak VI ile ilgili değişkenlerin ayarlanmasına imkan sağlar.
  - » Böylece aynı işlevi gerçekleştirebilmek için kullanmanız gereken bileşen sayısı ve dolayısıyla tel sayısı azalır.



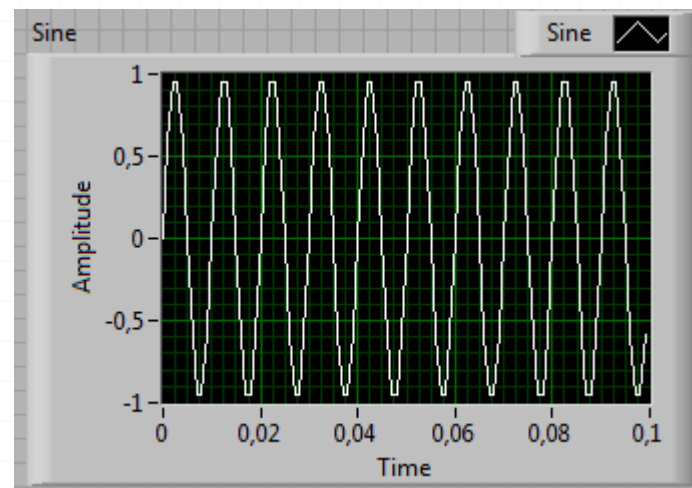
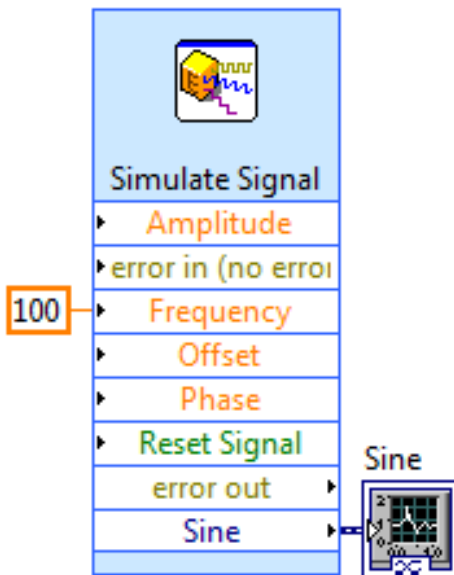
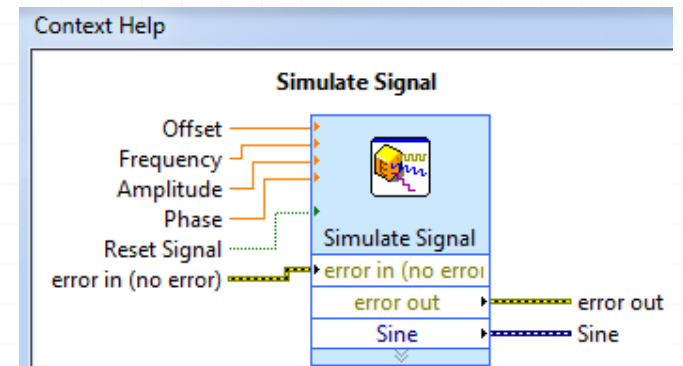
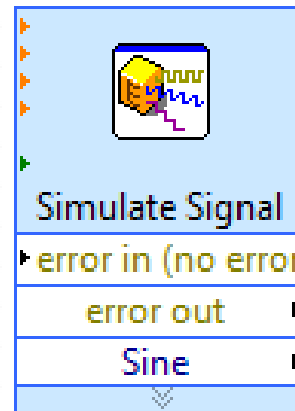
Çabuk VI



# Programlama Ortamı

## -Çabuk VI (Örnek)-

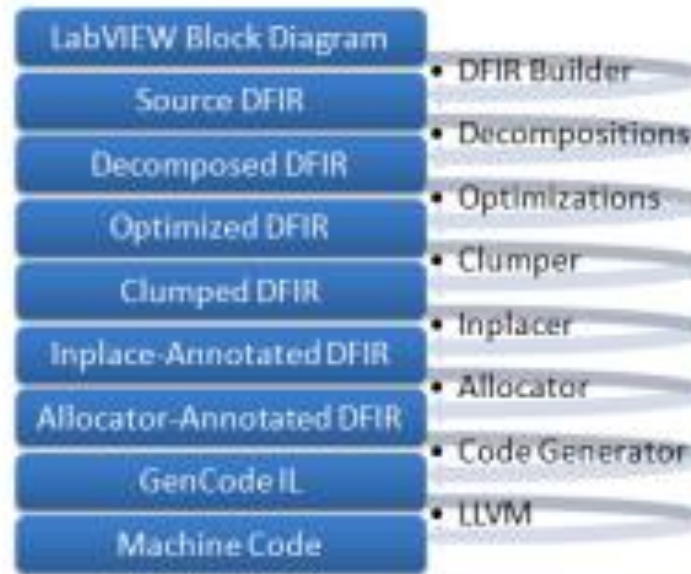
Simulate Signal



# Derlemek

## - VI Nasıl Derlenir? -

- » Oluşturulan programın derlenmesi, VI'yı çalıştırmak için "▶" simgesine tıklandığında kendiğiliğinden gerçekleştirilir.
- » G derleyicisi akış çizelgesini, çalışılan sistemin işlemcisinde çalıştırılabilecek bir makine kodu yığınınına çevirir.



# Örnek VI

## -Ön Pencere-

**SPP**

Akım ve gerilim artis yonu (T: artis, F: azalis)

**Gerilim Ayarla (V)** **Akım Ayarla (I)**

Akım isareti (T: -, F: +)

**Ac/Kapat** **DUR**

**Guc Kaynagi Yolu** VISA

**Gaussmetre Yolu** %GPIB0::12::INSTR

**Gerilim Koruma Siniri (V)** **Akım Koruma Siniri (I)**

Voltaj artma miktarı Akım artma miktarı

**Calisma Kipi** **Kontrol Secimi** **Alan Secimi** **Alan Birimi**

**Baslama Kipi** **Kip Secimi** **Hassasiyet** **Olcum Kipi**

**Akım** **Manyetik Alan** **Gerilim** **B-I**

20,202=

17,5=

15=

12,5=

10=

7,5=

5=

2,5=

0=

0 100 200 300 400 500 600 713

t

200=

150=

100=

50=

0=

-50=

-100=

-150=

-200=

0 100 200 300 400 500 600 713

t

4=

3,5=

3=

2,5=

2=

1,5=

1=

0,5=

0=

0 100 200 300 400 500 600 713

t

200=

150=

100=

50=

0=

-50=

-100=

-150=

-200=

-22 -10 0 10 22

I (A)

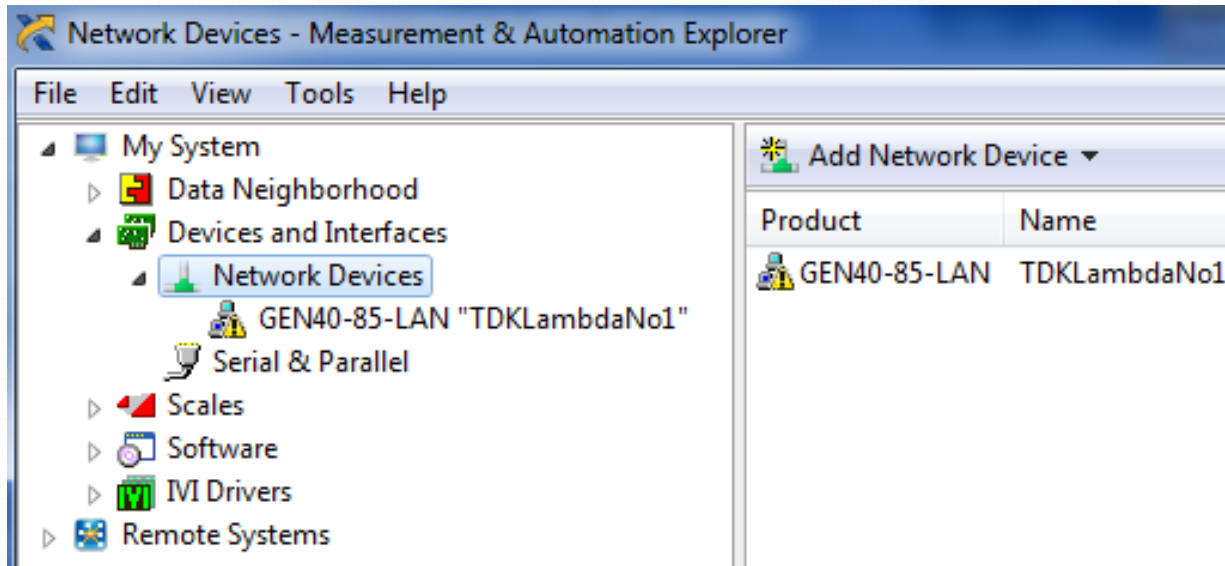
# Bilgisayar ile Bağlantı Yolları

-MAX-

NATIONAL INSTRUMENTS

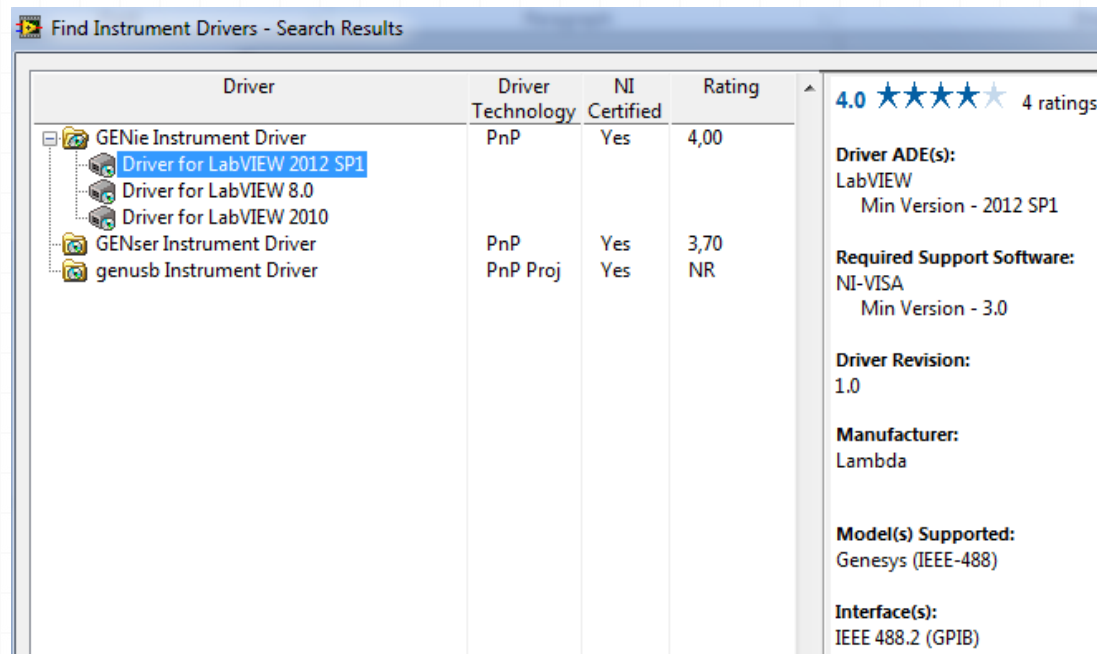
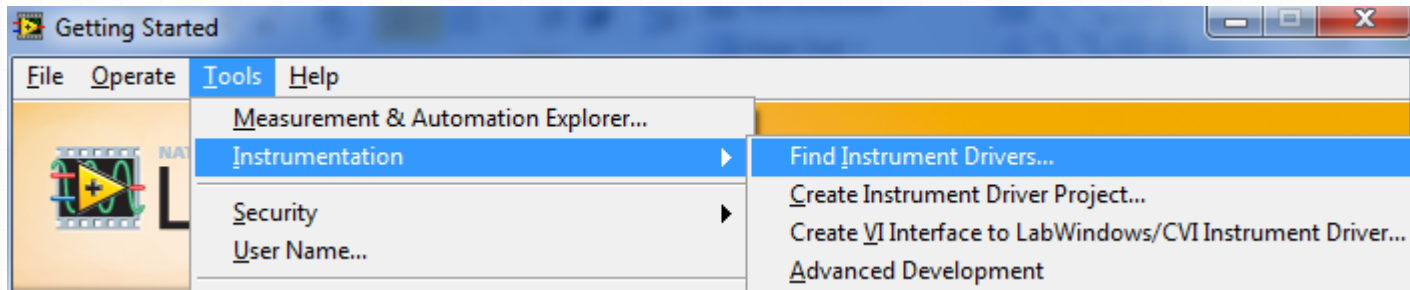
Measurement & Automation Explorer

» NI DAQ, GPIB, IMAQ, IVI, PCI, Motion, VISA ve VXI bağlantı yollarına sahip donanımları LabVIEW'da test edebilme, kullanabilme ve düzenleyebilme imkanı sağlar.



# Donanım Sürücülerini

-NI-



Driver	Driver Technology	NI Certified	Rating
GENie Instrument Driver	PnP	Yes	4,00
Driver for LabVIEW 2012 SP1			
Driver for LabVIEW 8.0			
Driver for LabVIEW 2010			
GENser Instrument Driver	PnP	Yes	3,70
genusb Instrument Driver	PnP Proj	Yes	NR

**4.0** ★★★★★ 4 ratings

**Driver ADE(s):**  
LabVIEW  
Min Version - 2012 SP1

**Required Support Software:**  
NI-VISA  
Min Version - 3.0

**Driver Revision:**  
1.0

**Manufacturer:**  
Lambda

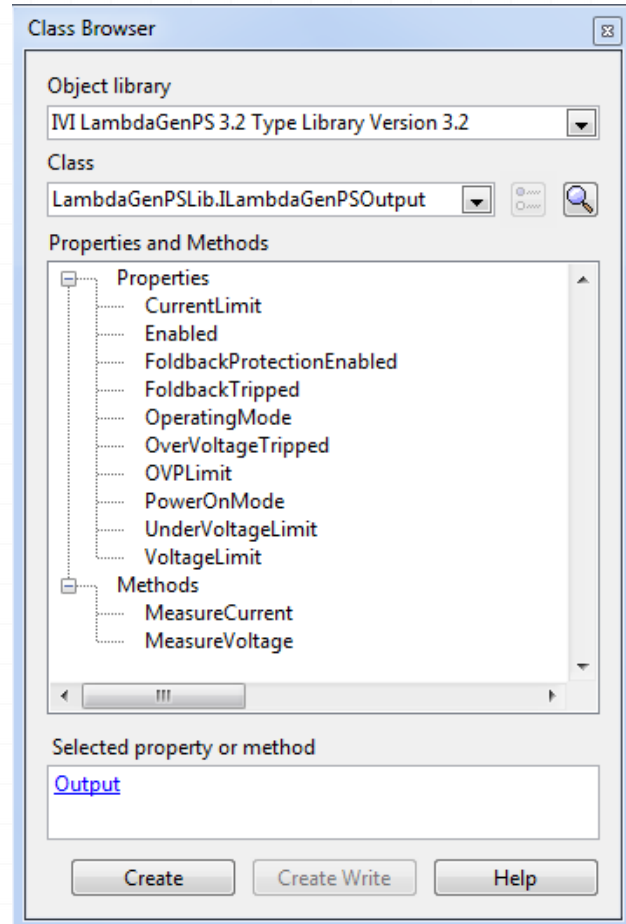
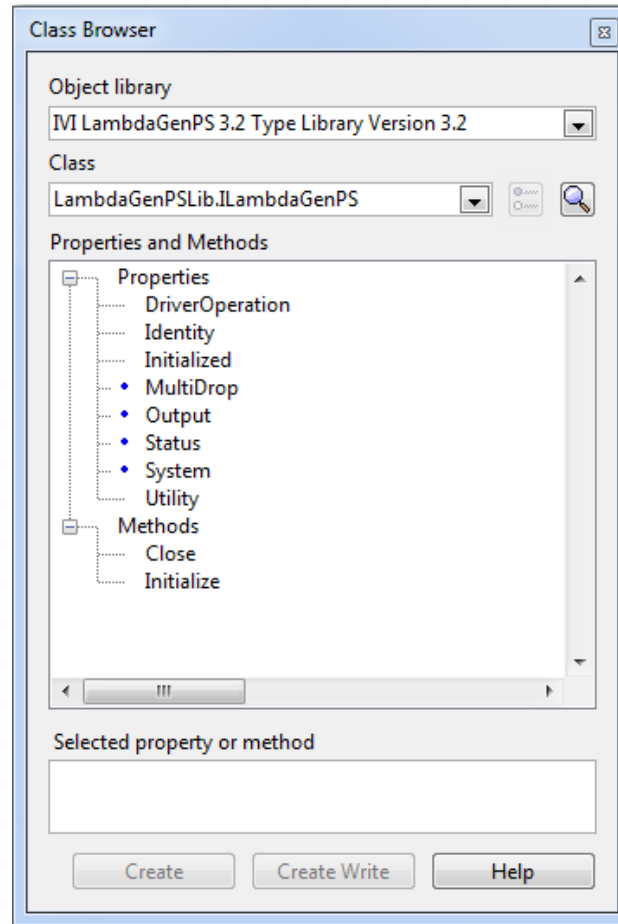
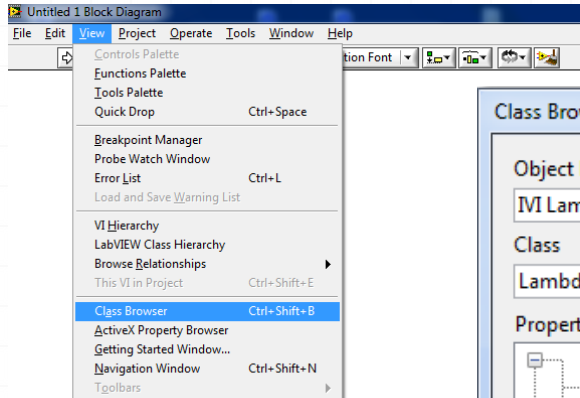
**Model(s) Supported:**  
Genesys (IEEE-488)

**Interface(s):**  
IEEE 488.2 (GPIB)



# Donanım Sürücüleri

## -IVI-COM-





*Dinlediğiniz için teşekkürler...*

0

# Kaynaklar

- » University of Oslo, LabVIEW programming I, FYS3240, 2013.
- » Hans-Petter Halvorsen, Introduction to LabVIEW, 2012.
- » Korolov Ihor, How to use LabVIEW, 2011.
- » Todd Murphey, LabVIEW for Control Analysis and Design.