LET271: Elektriska Mätsystem och Mätmetoder

LabVIEW

Jesper Pedersen

CHALMERS

Varför LabVIEW?

- **Obligatoriska** laborationer.
- LabVIEW-uppgifter kommer på **tentan**.
- Används av mängder av **företag**.

När LabVIEW?

- **30/1**, 10:15: Introduktion
- **30/1**, 13:15: Introduktion (forts.)
- **31/1**, 13:15: Introduktionsövningar (grupp A)
- **6/2**, 10:15: Repetition och exempel
- **7/2**, 13:15: Introduktionsövningar (grupp B)
- **8/2**, 8:15: Lösningar till övningar
- **16/2**, 8:15: LAB 3 (grupp A) (**obligatoriskt**)
- **23/2**, 8:15: LAB 3 (grupp B) (**obligatoriskt**)
- **1/3**, 10:15: Filhantering, Instrumentkommunikation, VISA, DAQ Assistant, Linear Fit.
- **1/3**, 13:15: LAB 4 (grupp A) (**obligatoriskt**)
- **2/3**, 8:15: LAB 4 (grupp B) (**obligatoriskt**)

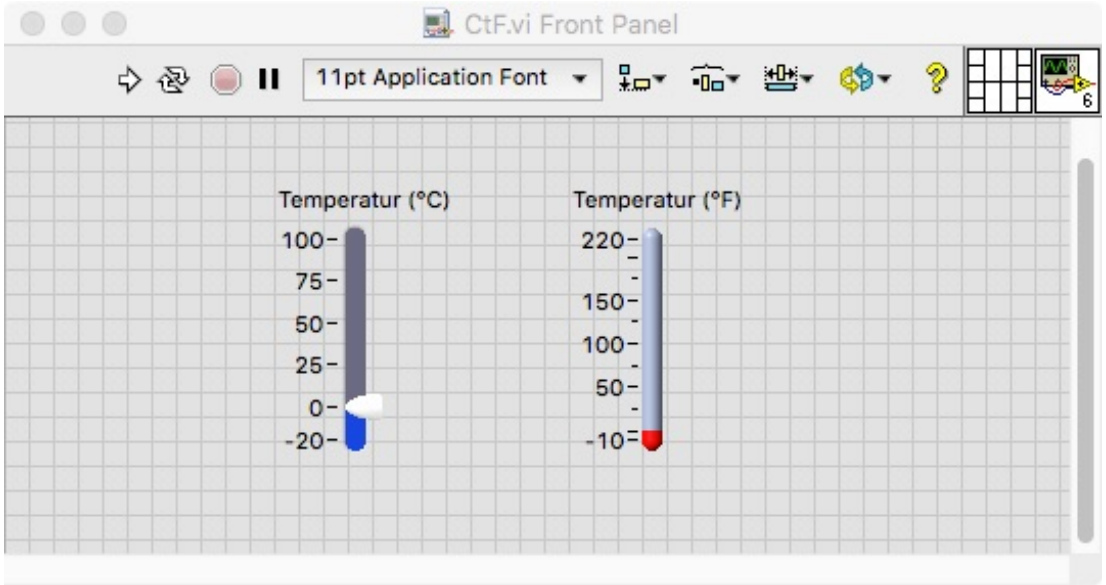
Översikt

LabVIEW

- **Lab**oratory **V**irtual **I**nstrument **E**ngineering **W**orkbench
- Grafisk programeringsmiljö
- Ett LabVIEW-program kallas **V**irtual **I**nstrument (VI)

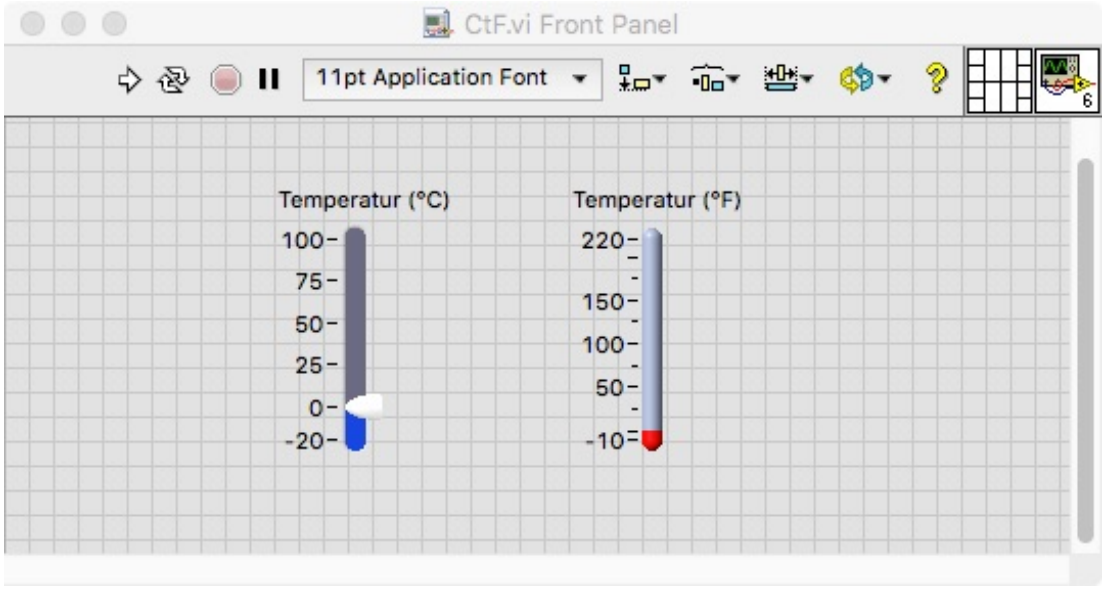
Virtual Instrument

- Frontpanel

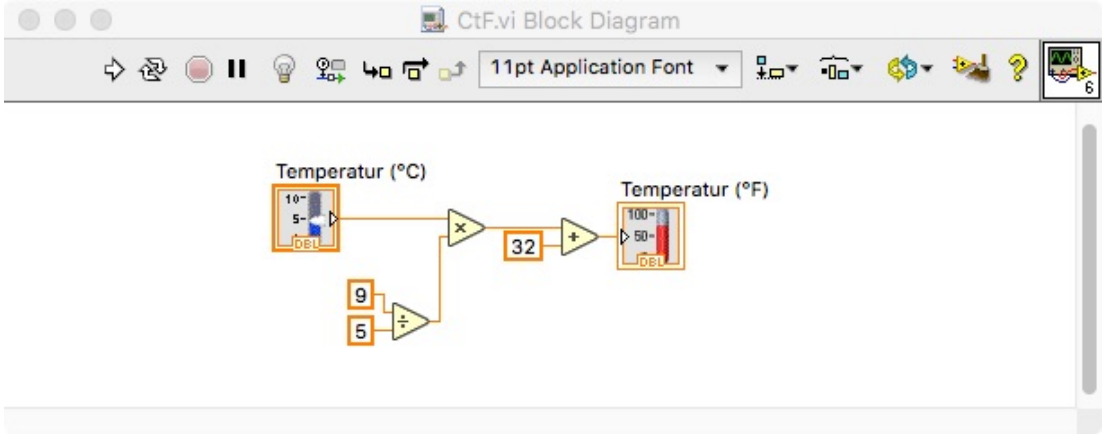


Virtual Instrument

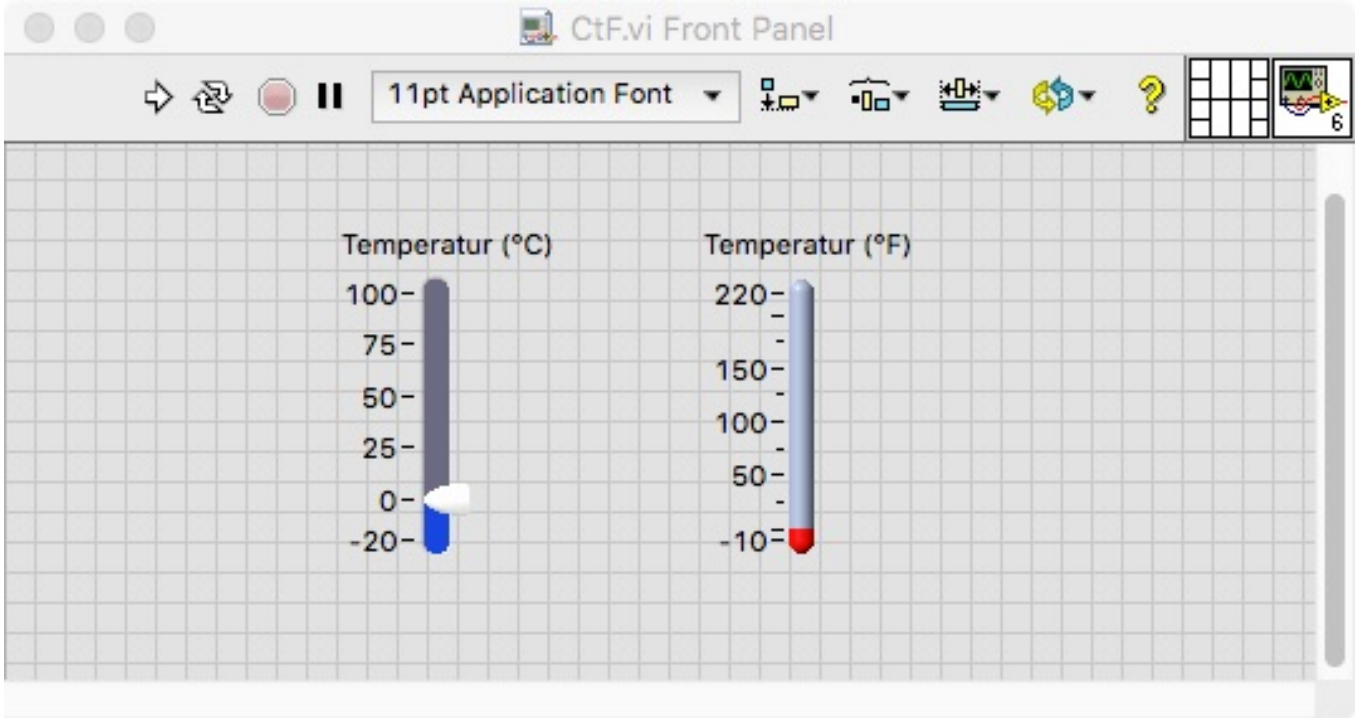
- Frontpanel



- Blockdiagram

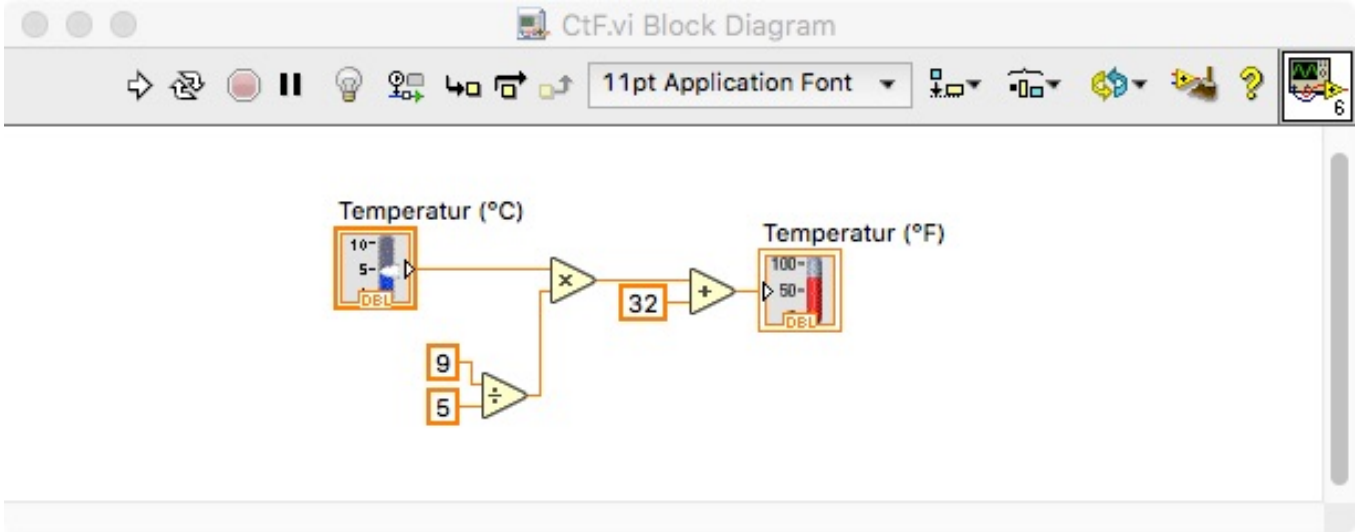


Frontpanel



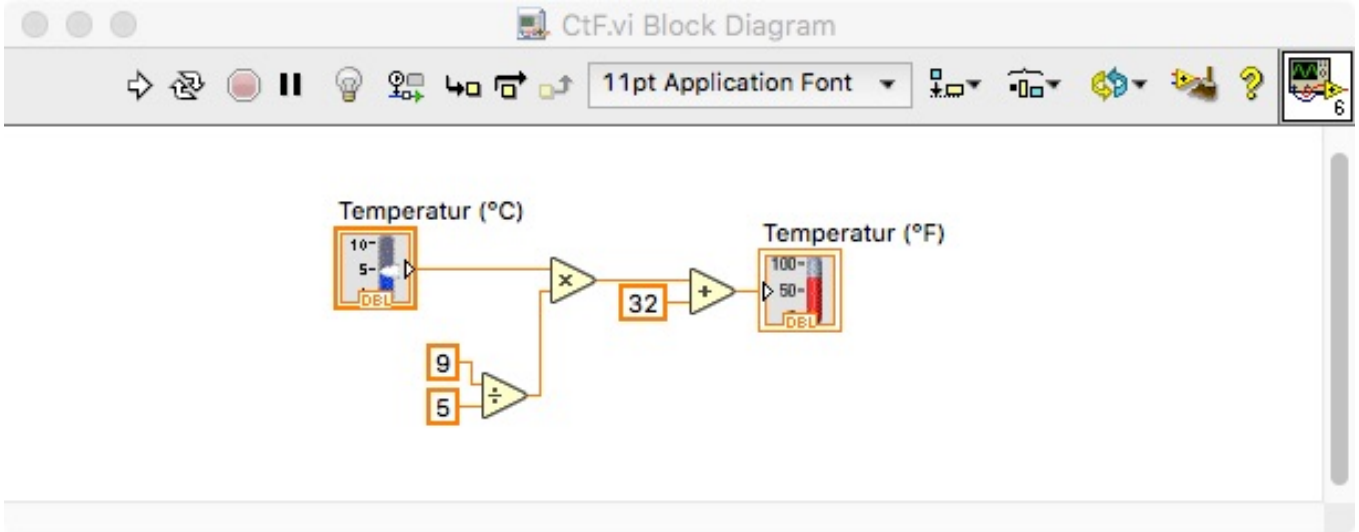
- Användargränssnitt.
- Uppbyggt av **kontroller** (inputs) och **indikatorer** (outputs).

Blockdiagram



- Innehåller den **grafiska koden**
- Består av:
  - Terminaler
  - Funktioner
  - Konstanter
  - Kopplingar

Blockdiagram



- Innehåller den **grafiska koden**
- Består av:
  - Terminaler
  - Funktioner
  - Konstanter
  - Kopplingar
  - Strukturer
  - Sub VI'n

Verktyg

View » Tools Palette

Verktyg

View » Tools Palette

Automatic Tool Selector

Verktyg

View » Tools Palette

Automatic Tool Selector

Operating Tool

Verktyg

View » Tools Palette

Automatic Tool Selector

Operating Tool

Positioning Tool

Verktyg

View » Tools Palette

Automatic Tool Selector

Operating Tool

Positioning Tool

Labeling Tool

Verktyg

View » Tools Palette

Automatic Tool Selector

Operating Tool

Positioning Tool

Labeling Tool

Wiring Tool

Verktyg

View » Tools Palette

Automatic Tool Selector

Operating Tool

Positioning Tool

Labeling Tool

Wiring Tool

Coloring Tool

Status

Status

Run

Status

Run

Run Continuously

Status

Run

Run Continuously

Abort Execution

Status

Run

Run Continuously

Abort Execution

Pause

Status

Run

Run Continuously

Abort Execution

Pause

Highlight Execution

Status

Run

Run Continuously

Abort Execution

Pause

Highlight Execution

Step

Status

Run

Run Continuously

Abort Execution

Pause

Highlight Execution

Step

Align, Distribute, Reorder

Översikt

Exempel

Exempel 1: Celsius till Fahrenheit

Skriv ett program som omvandlar grader Celsius (°C) till grader Fahrenheit (°F). Använd följande formel:

$$T_{\circ\mathsf{F}} = \frac{9}{5} \cdot T_{\circ\mathsf{C}} + 32$$

Kopplingar

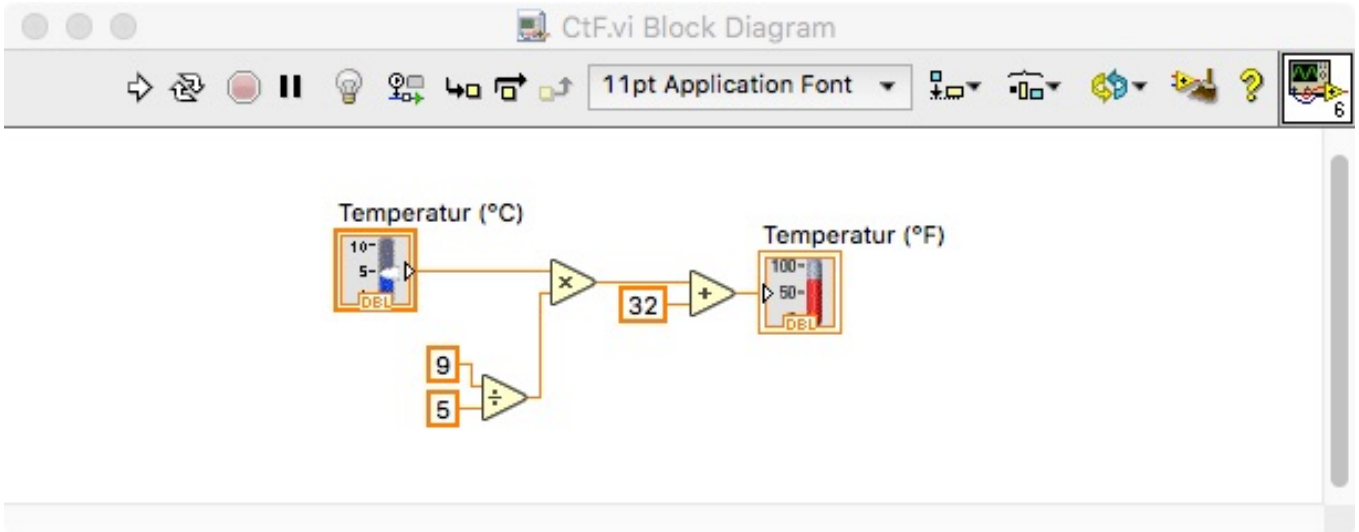
- Skicka data mellan objekt i blockdiagrammet genom **kopplingar**.

Kopplingar

- Skicka data mellan objekt i blockdiagrammet genom **kopplingar**.
- Färg, stil och tjocklek beror på **datatyp**.

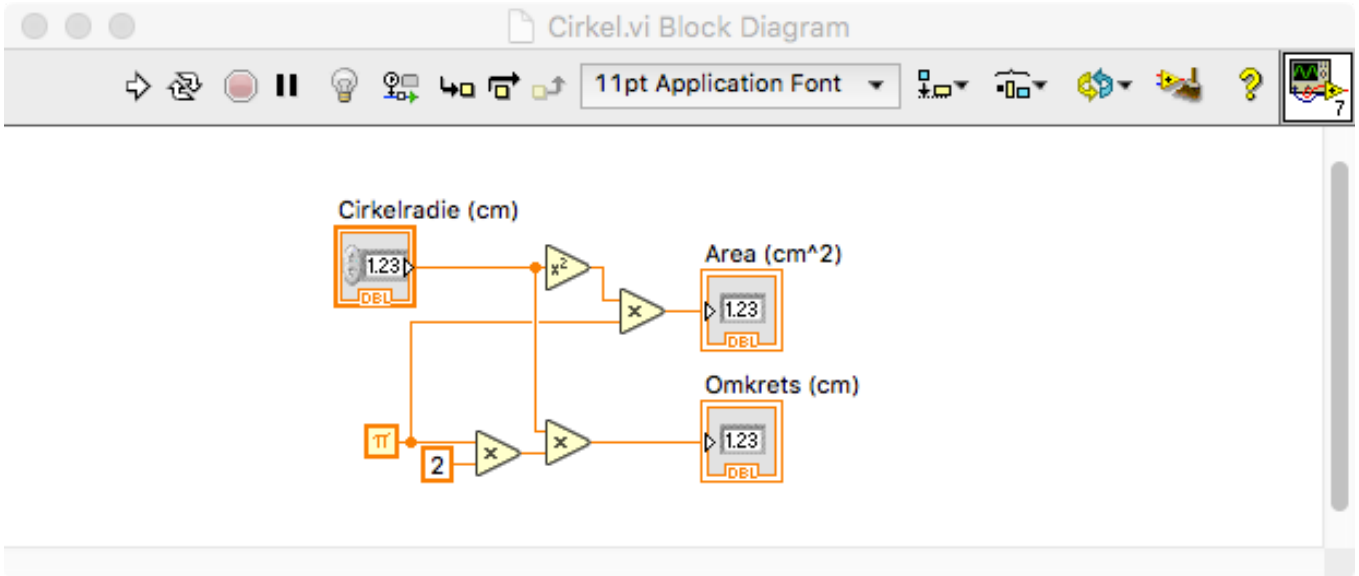
Typ	Skalär	1D Array	2D Array
Heltal			
Flyttal			
Boolskt			
Sträng			

Exekveringsordning



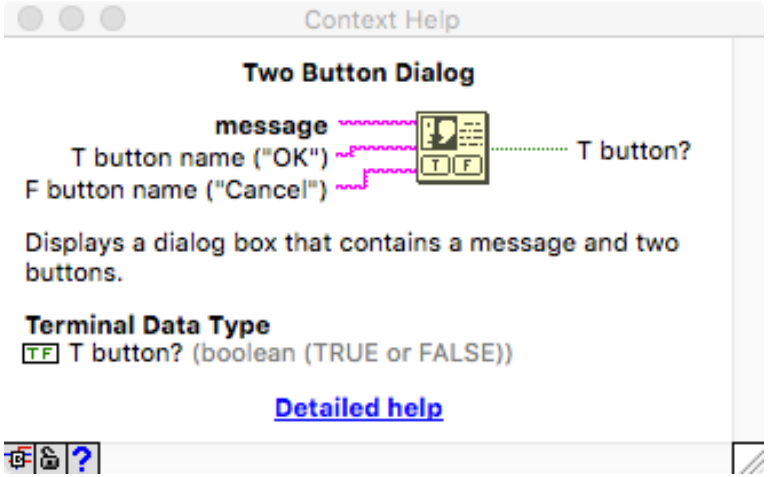
- En nod exekveras när den har **alla** nödvändiga inputs.
- När en nod exekveras producerar den utdata och skickar till nästa nod i dataflödet.

Exekveringsordning



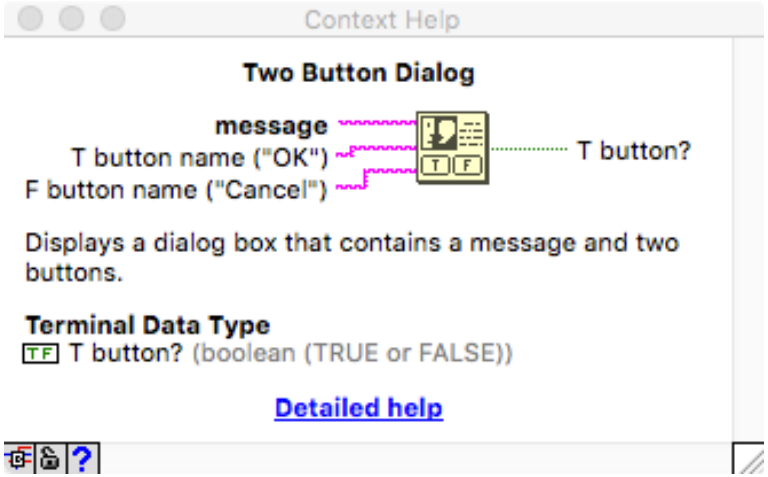
- Det kan vara **omöjligt** att säga vilken av oberoende noder som exekveras först.
- Om exekveringsordningen är viktig måste vi använda en **Sequence Structure**.

Hjälp



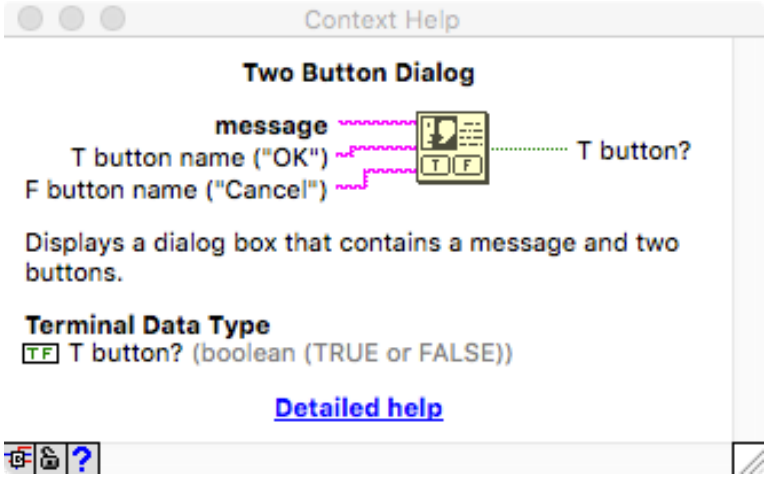
- **Context Help**
  - Visar grundläggande information om objekt.

Hjälp

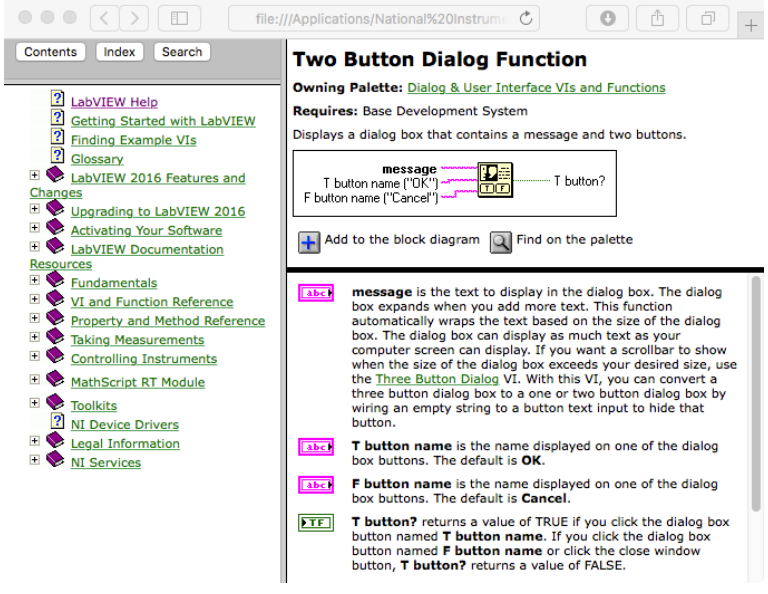


- **Context Help**
  - Visar grundläggande information om objekt.
  - Visa/dölj i **Help » Show Context Help** (⇧⌘H).

Hjälp

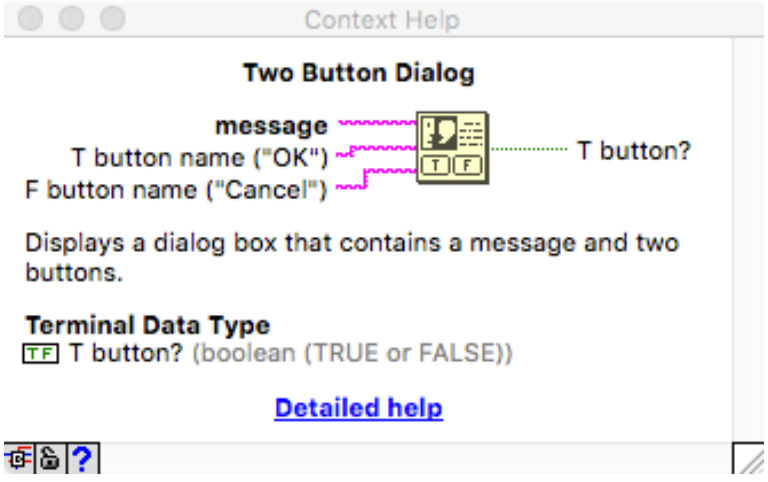


- **Context Help**
  - Visar grundläggande information om objekt.
  - Visa/dölj i **Help » Show Context Help** (⇧⌘H).

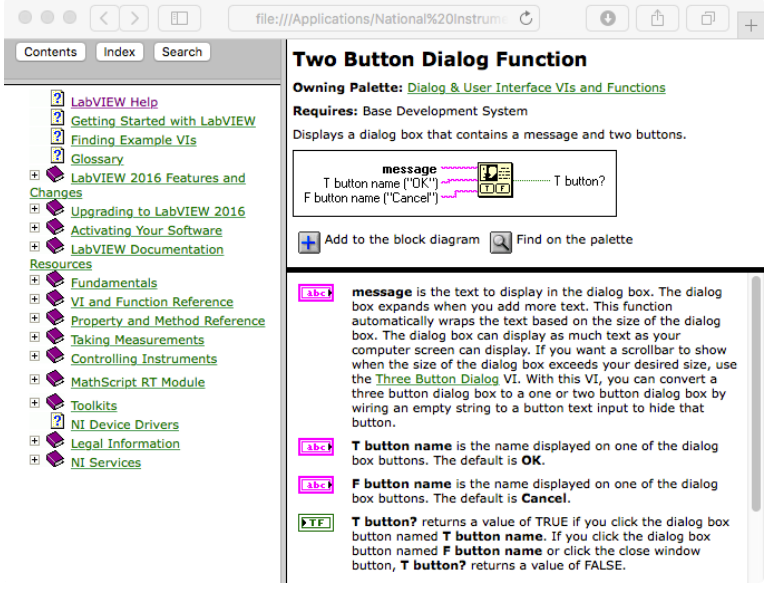


- **LabVIEW Help**
  - Detaljerad information om funktioner i LabVIEW.

Hjälp



- **Context Help**
  - Visar grundläggande information om objekt.
  - Visa/dölj i **Help » Show Context Help** (⇧⌘H).



- **LabVIEW Help**
  - Detaljerad information om funktioner i LabVIEW.
  - Öppna i **Help » LabVIEW Help** eller genom att klicka på **Detailed Help**-länken.

Felsökning

"Testing shows the presence, not the absence of bugs." Edsger Dijkstra

Felsökning

"Testing shows the presence, not the absence of bugs." Edsger Dijkstra

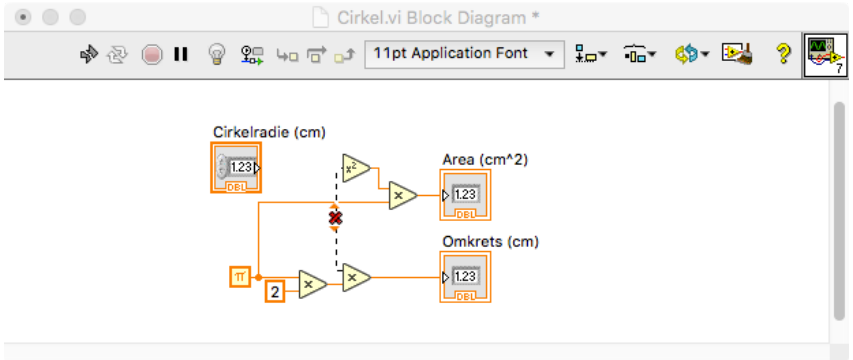
- Inom all programmering finns buggar som kan
  - **förhindra** att programmet körs.

Felsökning

"Testing shows the presence, not the absence of bugs." Edsger Dijkstra

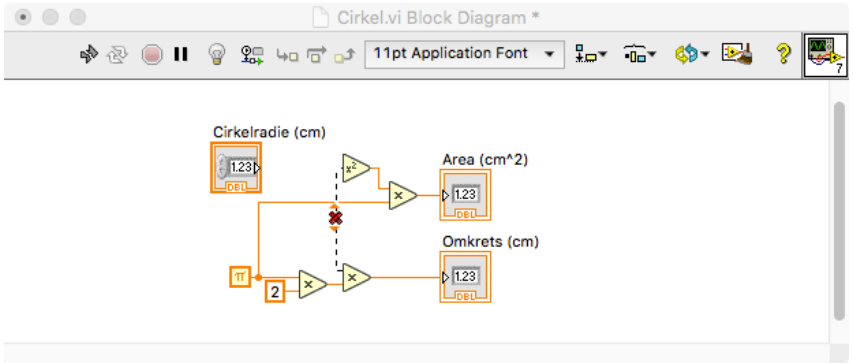
- Inom all programmering finns buggar som kan
  - **förhindra** att programmet körs.
  - generera **felaktiga** resultat.

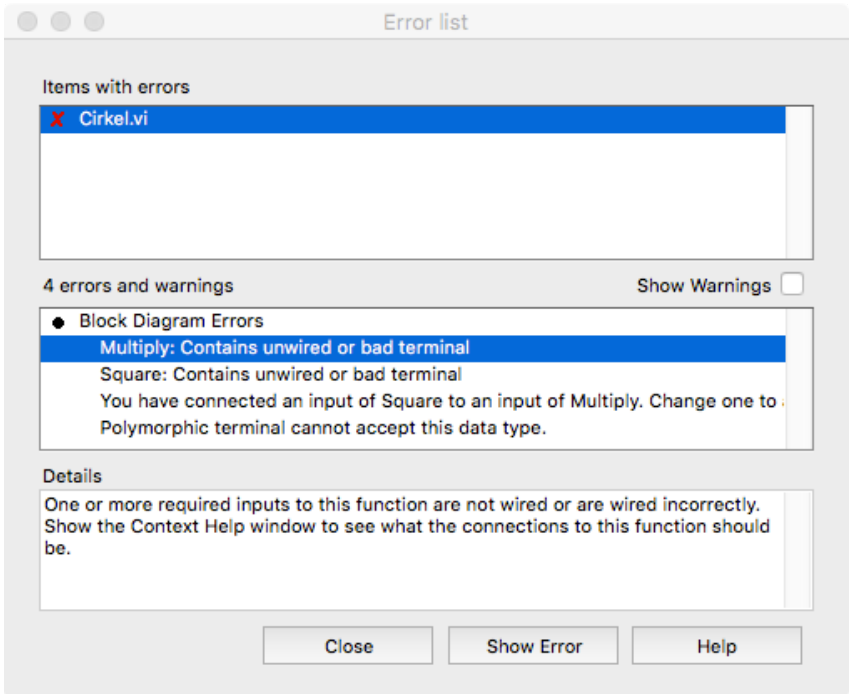
Felsökning



- Om LabVIEW-programmet inte kan köras indikeras detta genom en bruten **Run**-pil  .

Felsökning





- Om LabVIEW-programmet inte kan köras indikeras detta genom en bruten **Run**-pil  .

- **Vad göra?** Tryck på symbolen för att öppna **Error List Window**.

Felsökning

- Om programmet körs men producerar felaktiga resultat:

Felsökning

- Om programmet körs men producerar felaktiga resultat:
  - Använd **Highlight Execution**  .

Felsökning

- Om programmet körs men producerar felaktiga resultat:
  - Använd **Highlight Execution**  .
  - Använd **Step Into**, **Step Over** och **Step Out**  .

Felsökning

- Om programmet körs men producerar felaktiga resultat:
  - Använd **Highlight Execution**  .
  - Använd **Step Into**, **Step Over** och **Step Out**  .
  - Använd **Probe Tool**  .

Felsökning

- Om programmet körs men producerar felaktiga resultat:
  - Använd **Highlight Execution**  .
  - Använd **Step Into**, **Step Over** och **Step Out**  .
  - Använd **Probe Tool**  .

Exempel 2: Felsökning

Kör ett program med nämnda metoder.

Kortkommandon

Mac	Windows	Beskrivning
⌘E	Ctrl E	Visa frontpanelen
⇧⌘H	Ctrl H	Visa/dölj hjälp
⌘Z	Ctrl Z	Ångra
⌘R	Ctrl R	Run
⌘T	Ctrl T	Frontpanelen och blockdiagrammet sida vid sida
⌘B	Ctrl B	Ta bort trasiga kopplingar

Översikt

Strukturer

- Innehåller **sektioner** med grafisk kod.

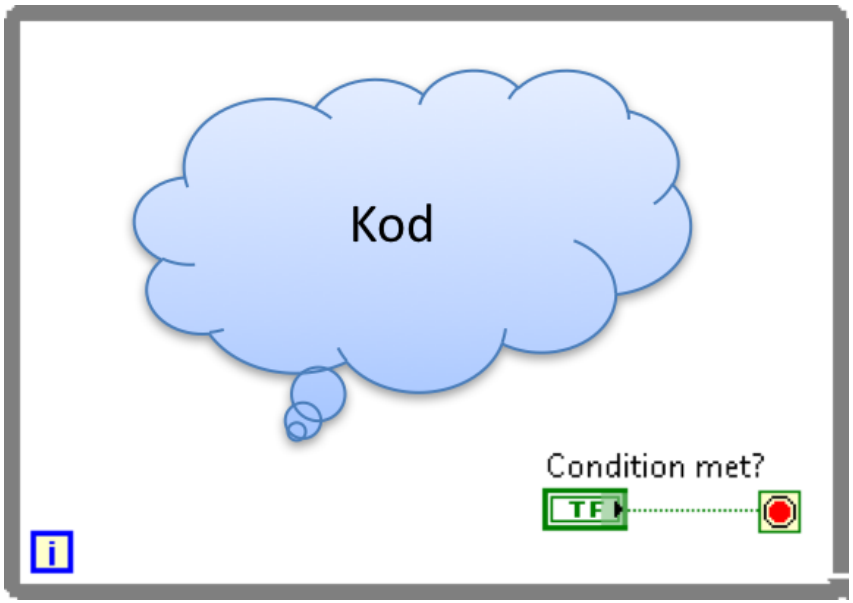
Strukturer

- Innehåller **sektioner** med grafisk kod.
- Kontrollerar **hur** och **när** koden exekveras.

Strukturer

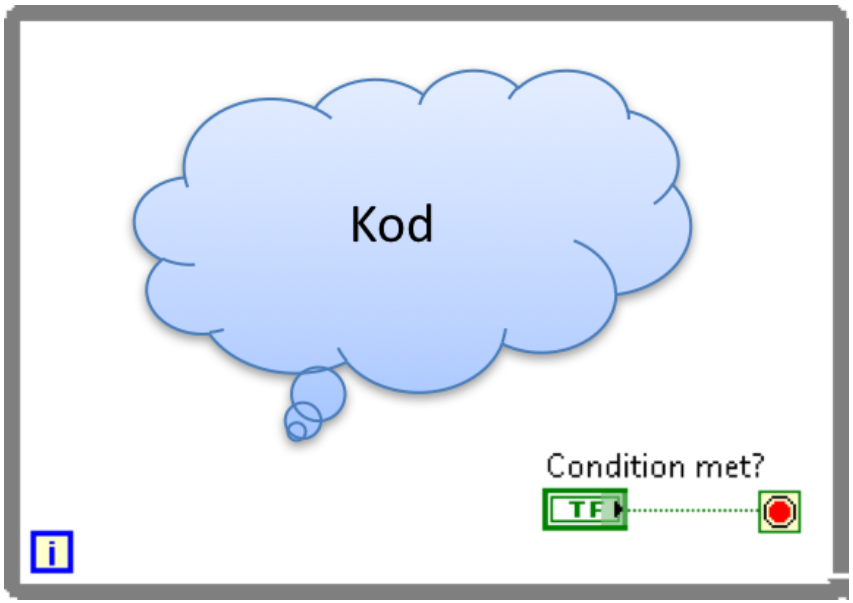
- Innehåller **sektioner** med grafisk kod.
- Kontrollerar **hur** och **när** koden exekveras.
  - While Loop
  - For Loop
  - Case Structure
  - Sequence Structure
  - Formula Node

While Loop



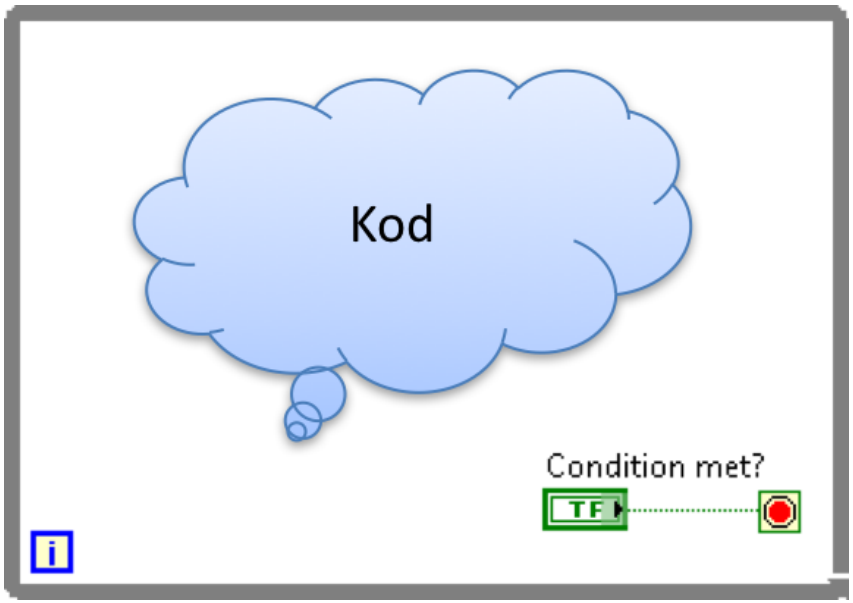
- Exekverar koden tills ett **villkor** är uppfyllt.

While Loop



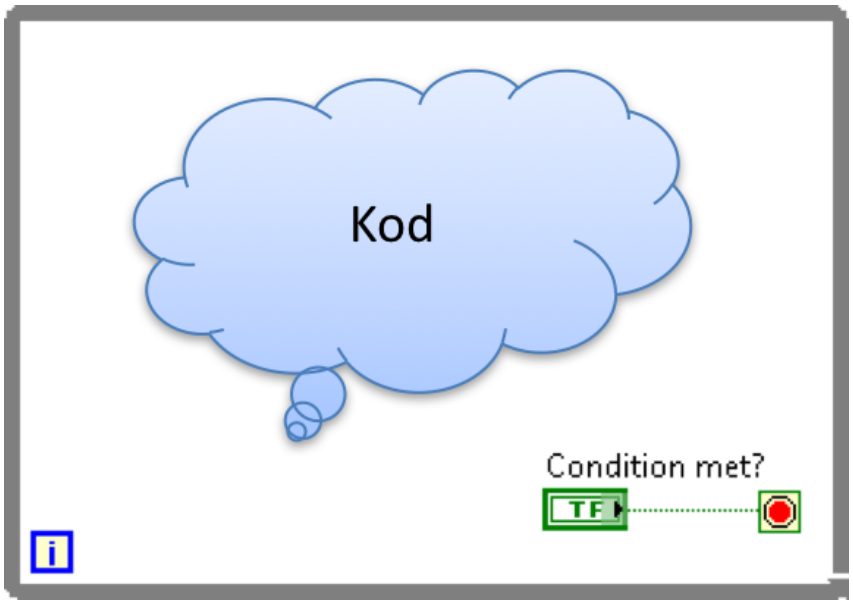
- Exekverar koden tills ett **villkor** är uppfyllt.
- Koden exekveras **minst en gång**.

While Loop



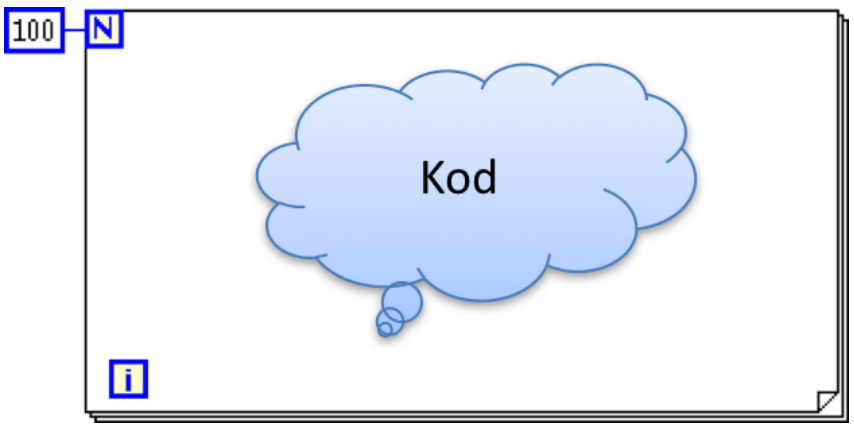
- Exekverar koden tills ett **villkor** är uppfyllt.
- Koden exekveras **minst en gång**.
- **Iterationsterminal**: Antal iterationer (räknar från **0**).

While Loop



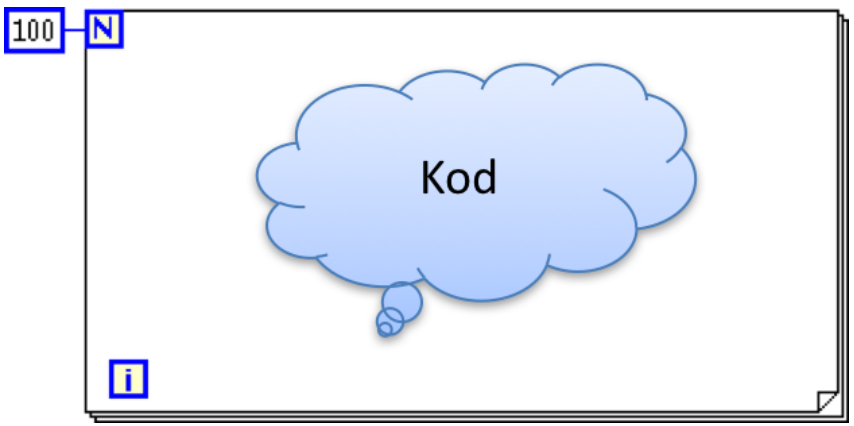
- Exekverar koden tills ett **villkor** är uppfyllt.
- Koden exekveras **minst en gång**.
- **Iterationsterminal**: Antal iterationer (räknar från **0**).
- **Villkorsterminal**:
  - **Stanna** om sant
  - **Fortsätt** om sant

For Loop



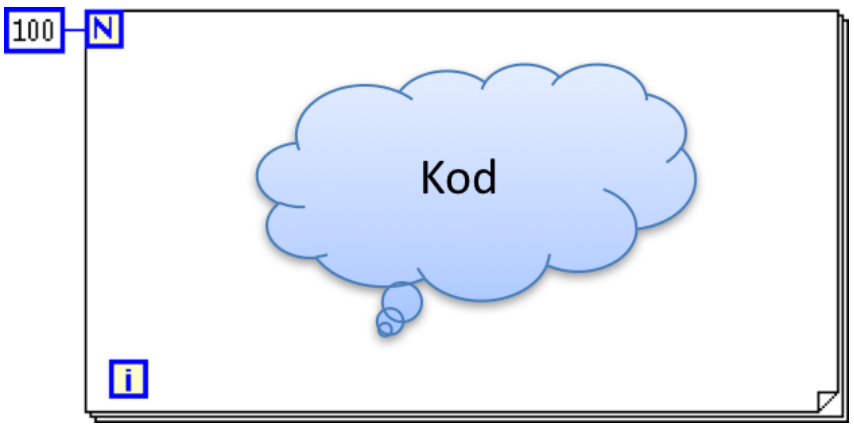
- Exekverar koden ett **förutbestämt** antal gånger.

For Loop



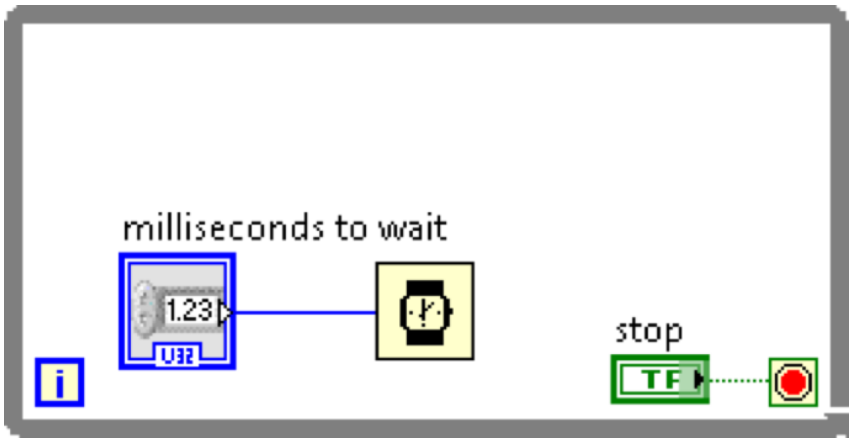
- Exekverar koden ett **förutbestämt** antal gånger.
- **Räknare**: Indikerar antalet iterationer.

For Loop



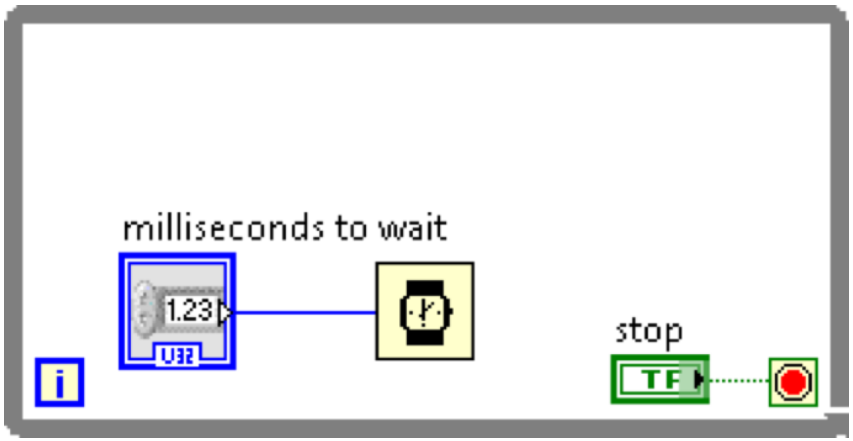
- Exekverar koden ett **förutbestämt** antal gånger.
- **Räknare**: Indikerar antalet iterationer.
- **Iterationsterminal**: Antal iterationer (räknar från **0** till **N-1**).

Loop Timing



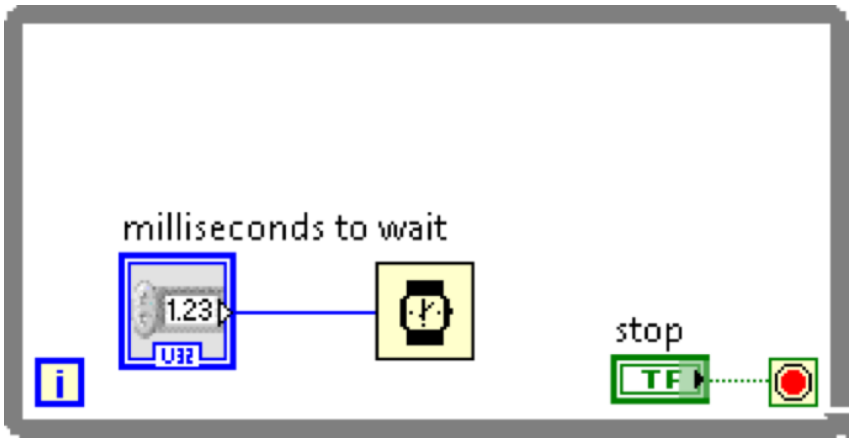
- När en iteration i loopen avslutas startar nästa iteration **omedelbart**.

Loop Timing



- När en iteration i loopen avslutas startar nästa iteration **omedelbart**.
- Använd **Wait**-funktionen för att ge processorn tid slutföra andra uppgifter.

Loop Timing

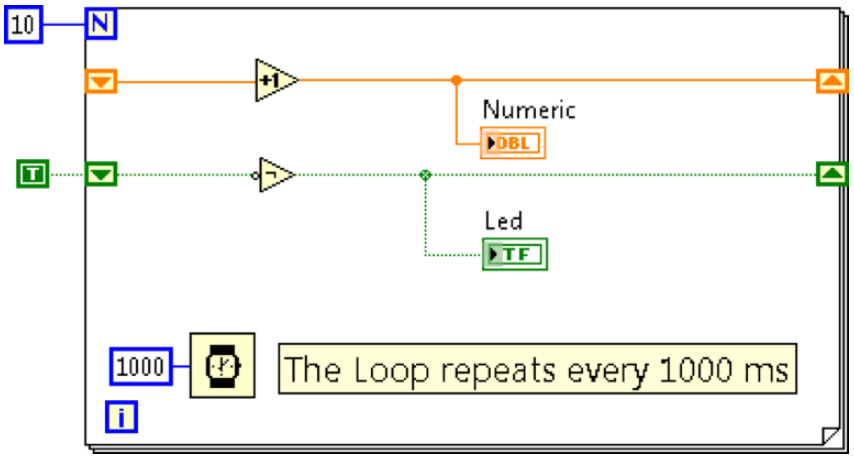


- När en iteration i loopen avslutas startar nästa iteration **omedelbart**.
- Använd **Wait**-funktionen för att ge processorn tid slutföra andra uppgifter.

Exempel 3: Loopar

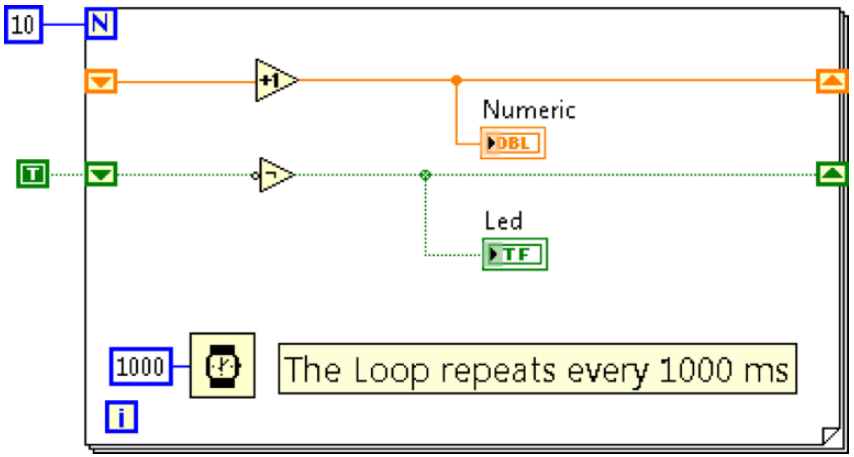
Skriv ett program som indikerar iterationerna i en for-loop för $N = 20$. Använd en wait-funktion. Gör detsamma med en while-loop som kan avslutas efter 20 iterationer eller när användaren avslutar programmet.

Shift Registers



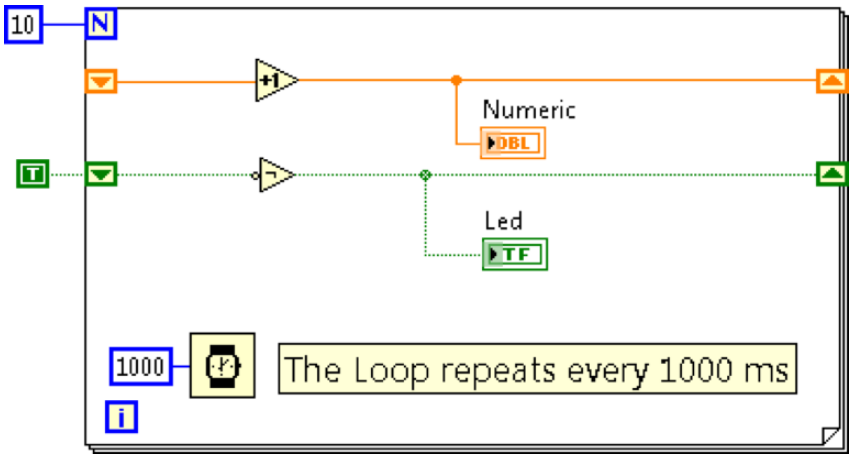
- Skicka data från föregående iteration med **Shift Register**.

Shift Registers



- Skicka data från föregående iteration med **Shift Register**.
- Skapa genom att högerklicka på höger eller vänsterkanten och välj **Add Shift Register**.

Shift Registers



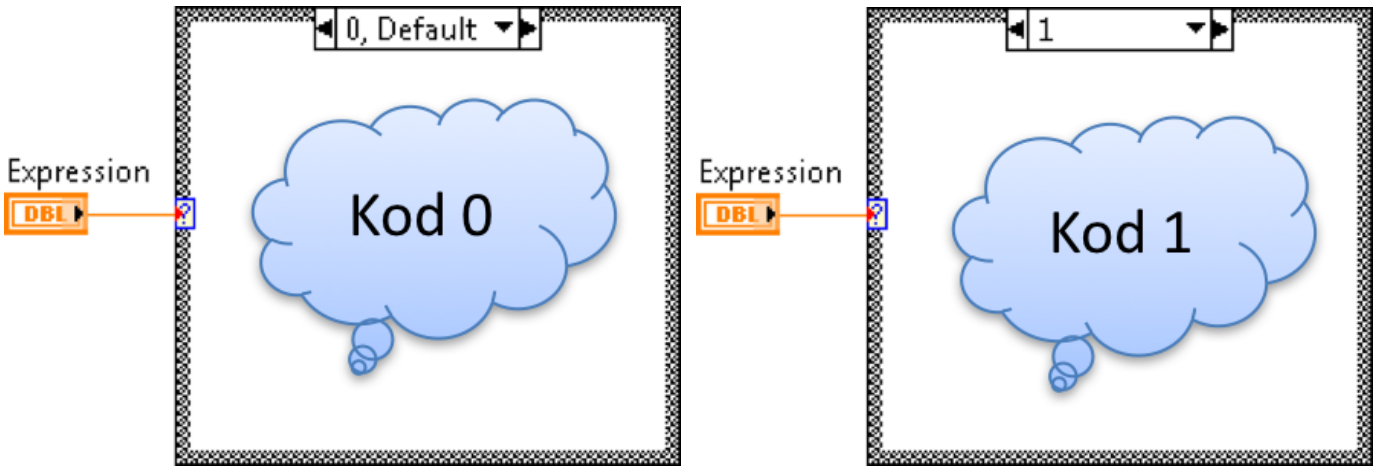
- Skicka data från föregående iteration med **Shift Register**.
- Skapa genom att högerklicka på höger eller vänsterkanten och välj **Add Shift Register**.

Exempel 4: Shift Register

Skriv ett program som slumpar fram ett heltal mellan 0 och 10 varje sekund tills programmet avslutas av användaren. Det nuvarande och föregående slumptalet ska visas på skärmen.

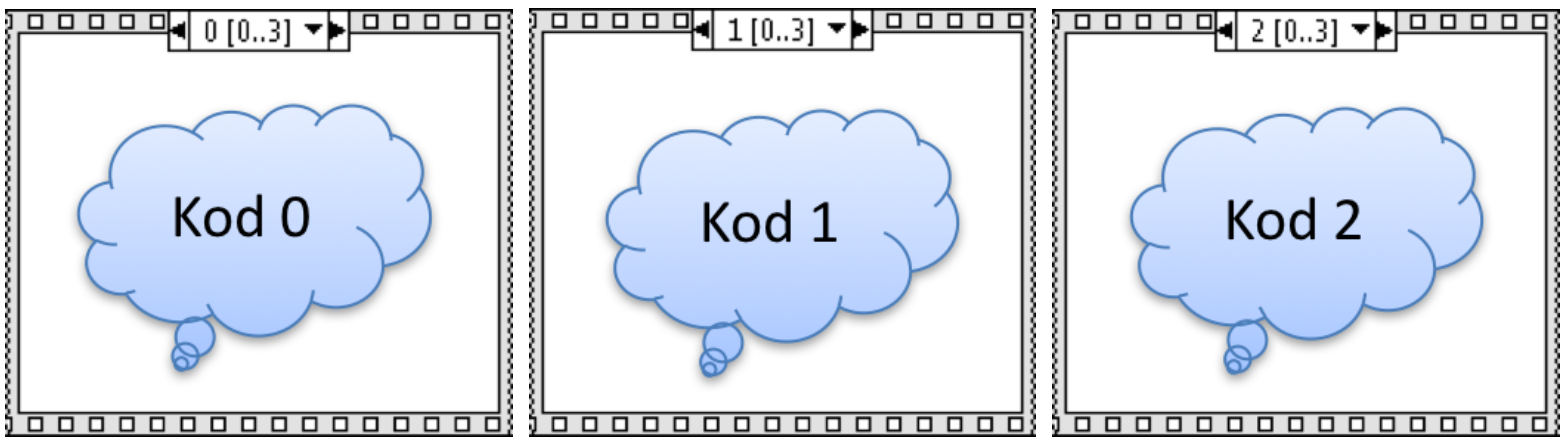
Case

- Har två eller fler **subdiagram**.
- Exekverar **endast ett** diagram.



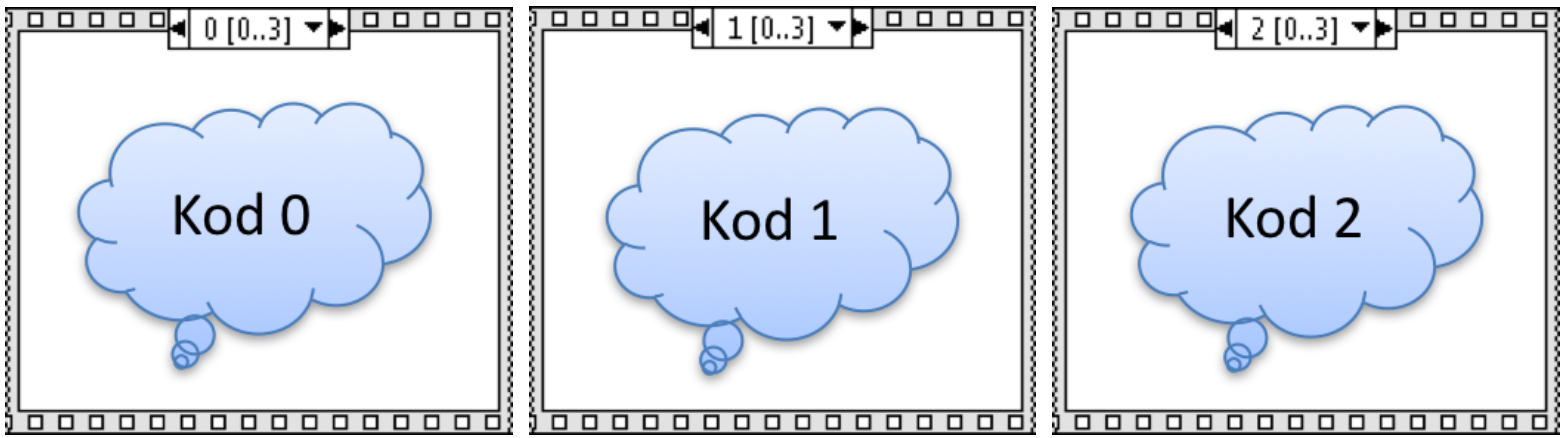
Stacked Sequences

- Ett eller flera subdiagram som **exekveras sekventiellt**.
- Garanterar att ett subdiagram körs **före** ett annat.



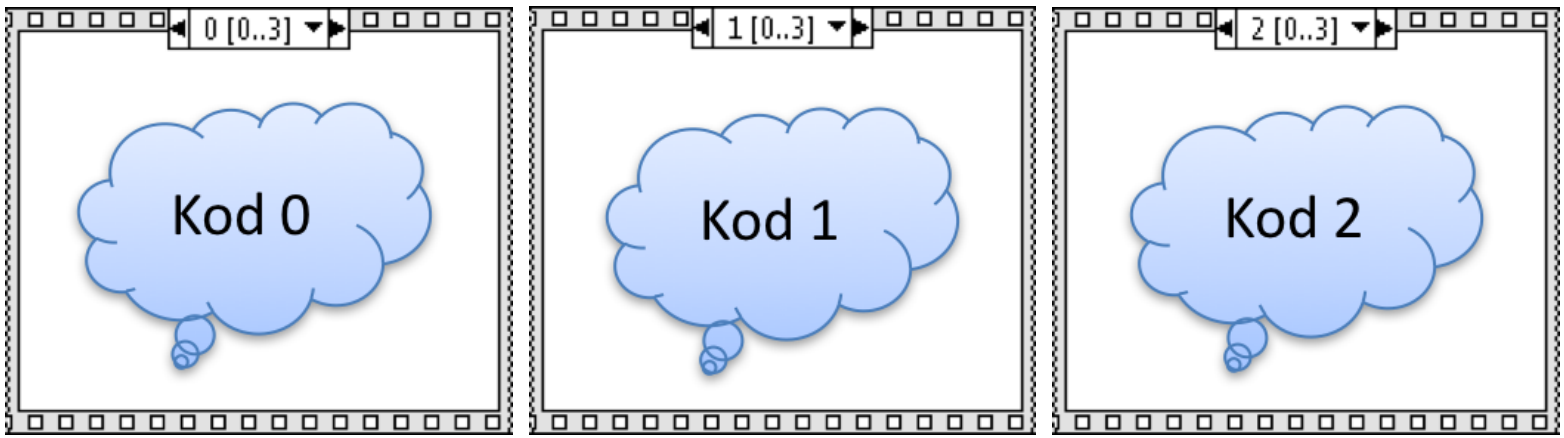
Stacked Sequences

- Ett eller flera subdiagram som **exekveras sekventiellt**.
- Garanterar att ett subdiagram körs **före** ett annat.
- Sekvensen startar när **all data** kopplad till sekvensen finns **tillgänglig**.



Stacked Sequences

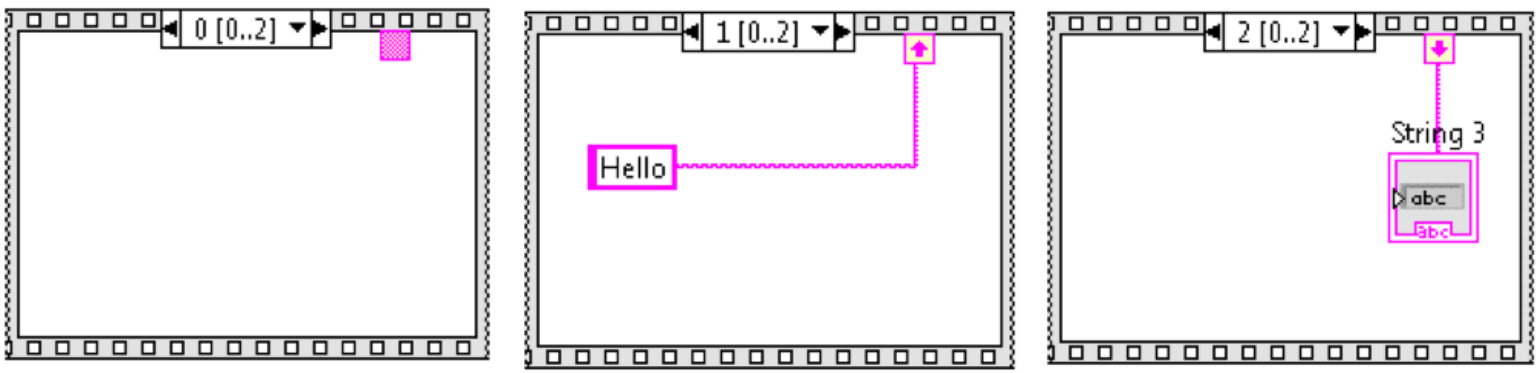
- Ett eller flera subdiagram som **exekveras sekventiellt**.
- Garanterar att ett subdiagram körs **före** ett annat.
- Sekvensen startar när **all data** kopplad till sekvensen finns **tillgänglig**.
- Data från varje subdiagram lämnar när **alla** subdiagram har körts.



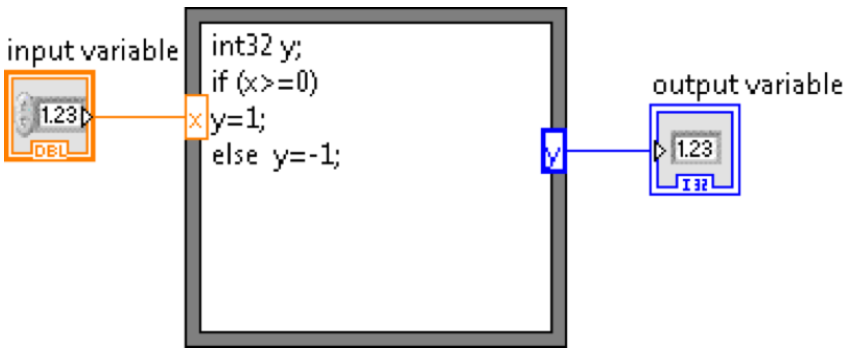
Stacked Sequences

Skicka data mellan subdiagram:

- Använd **Sequence Locals**
- Högerklicka på kanten och välj **Add Sequence Local**

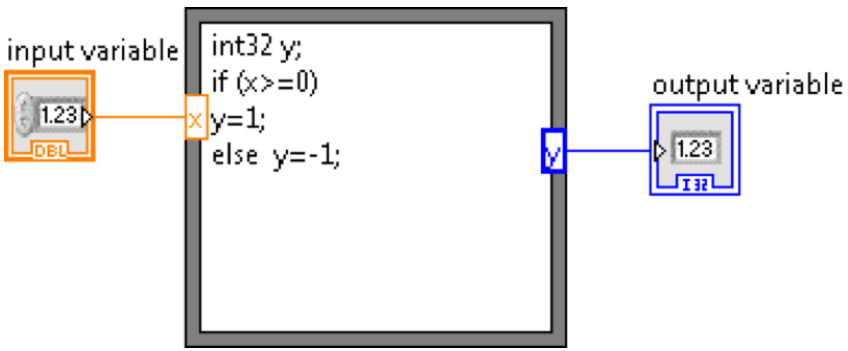


Formula Node



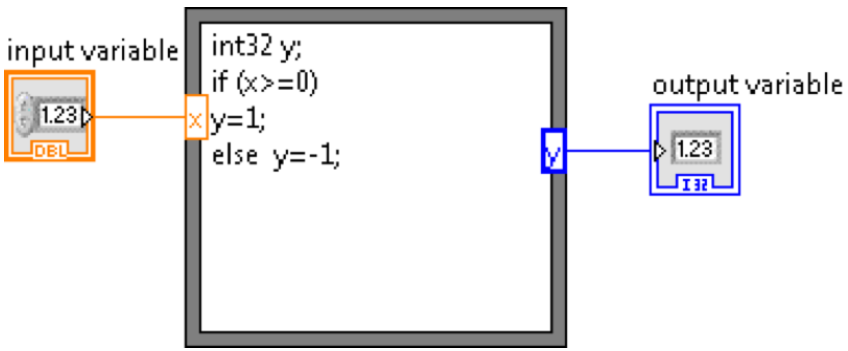
- Evaluerar **matematiska formler** i blockdiagrammet. Syntaxen liknar den i C.

Formula Node



- Evaluerar **matematiska formler** i blockdiagrammet. Syntaxen liknar den i C.
- Glöm inte att avsluta ett uttryck med **semikolon** (;).

Formula Node

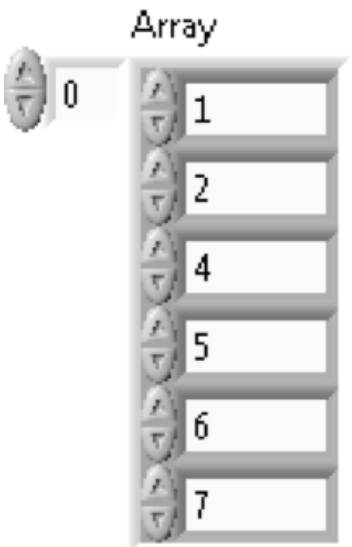


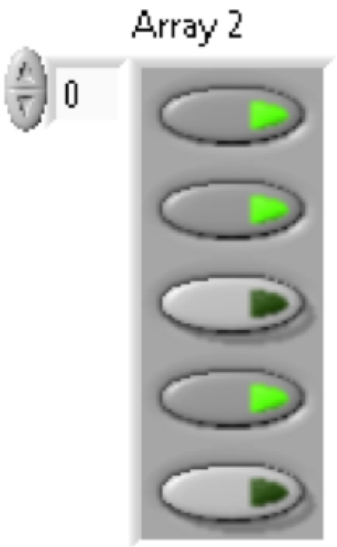
- Evaluerar **matematiska formler** i blockdiagrammet. Syntaxen liknar den i C.
- Glöm inte att avsluta ett uttryck med **semikolon** (;).
- Lägg till **inputs** och **outputs** genom att högerklicka på kanten och välja **Add Input/Output**.

Översikt

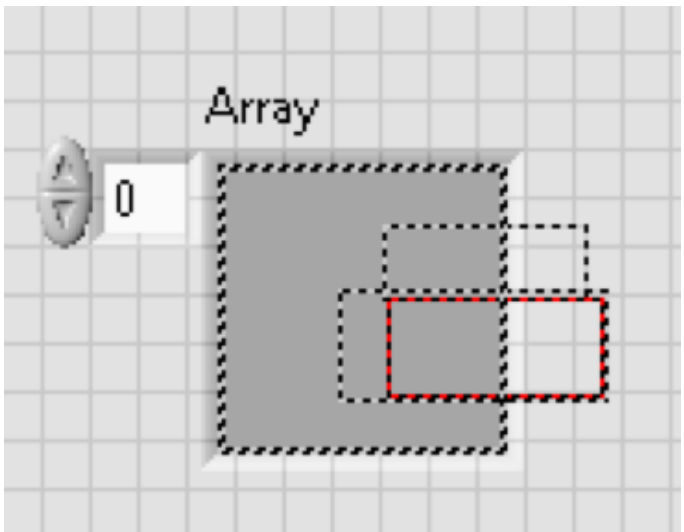
Arrays

- Kombinera data av samma typ i en **datastruktur**.
- Första **indexet** i en array är **0**.



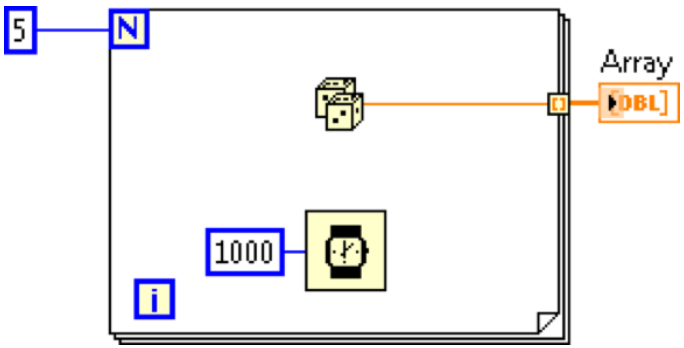


Skapa Arrays



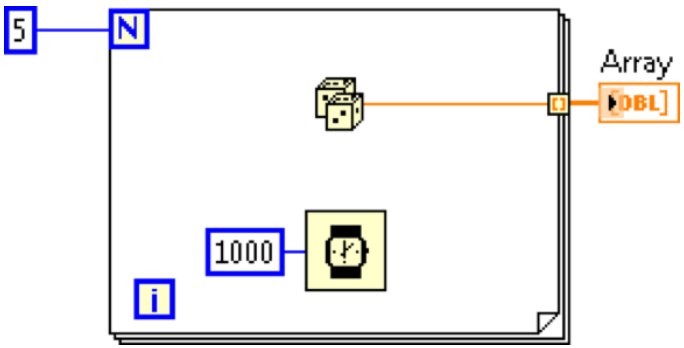
- Lägg till ett **Array Shell**.
- Lägg ett **dataobjekt** eller element i skalet.

Auto-Indexed Tunnels



- **Autoindexerade** tunnlar ger ett **nytt element** för varje loopiteration.

Auto-Indexed Tunnels



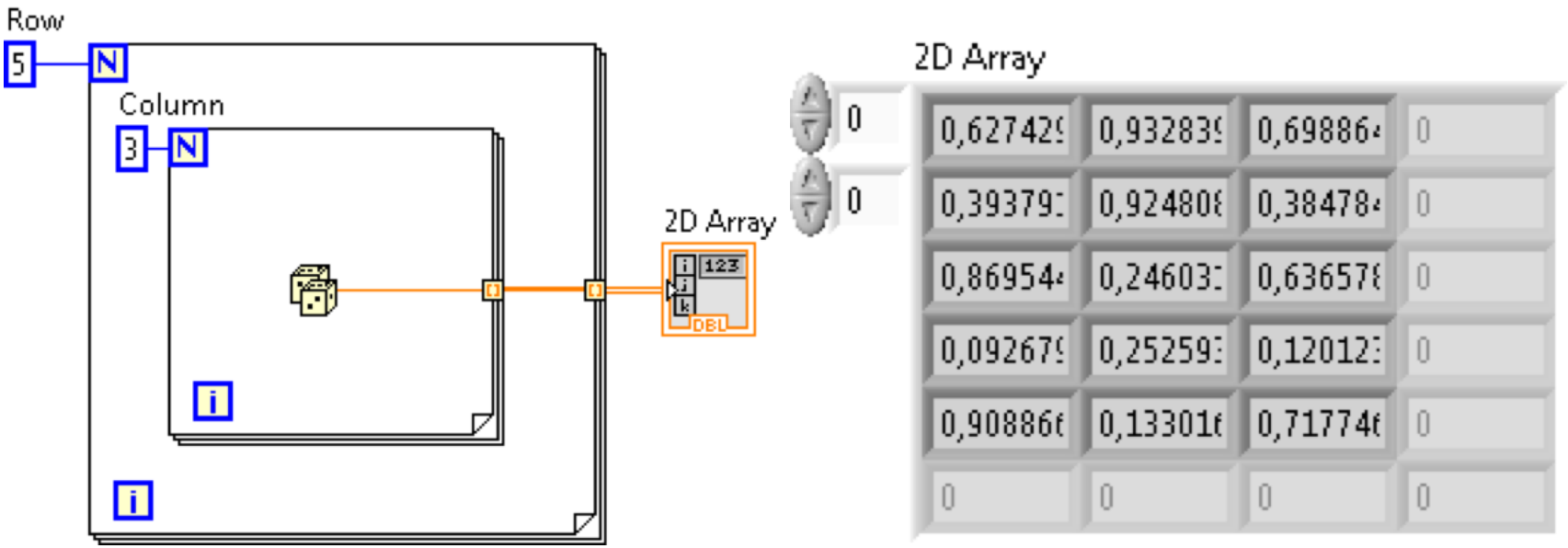
- **Autoindexerade** tunnlar ger ett **nytt element** för varje loopiteration.

Exempel 5: Autoindexerade tunnlar

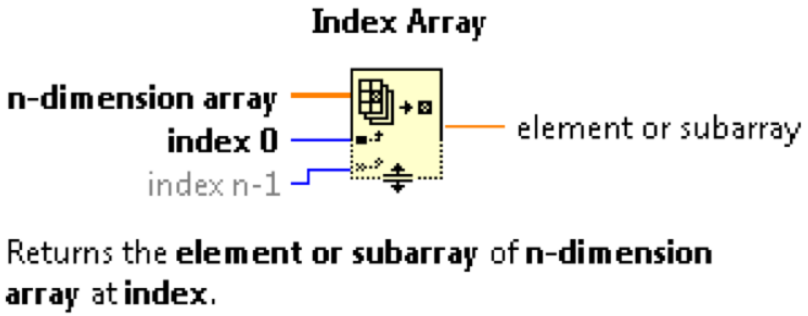
Skriv ett program som sparar 5 slumptal 0 eller 1 med lika stor sannolikhet i en array.

Skapa 2D Arrays

- Använd **nästlade** for-loopar.
- **Yttre** loopen ger **radindex** och **inre** loopen ger **kolumnindex**.

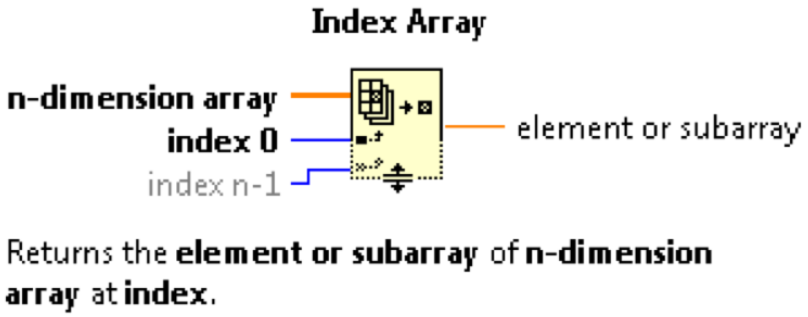


Exempel på Arrayfunktioner

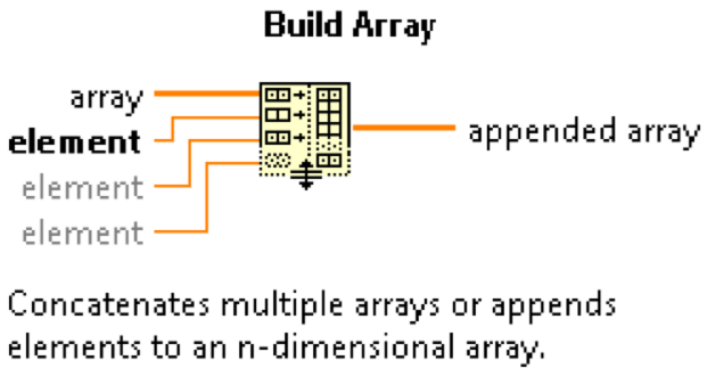


- Returnera värdet av ett **element** med specificerat **index**.

Exempel på Arrayfunktioner



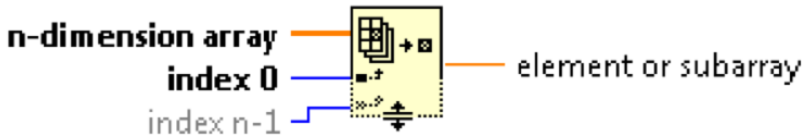
- Returnera värdet av ett **element** med specificerat **index**.



- **Lägg till** värden till en array.

Exempel på Arrayfunktioner

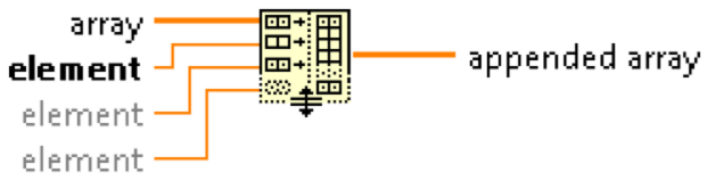
Index Array



Returns the **element or subarray** of **n-dimension array** at **index**.

- Returnera värdet av ett **element** med specificerat **index**.

Build Array



Concatenates multiple arrays or appends elements to an n-dimensional array.

- **Lägg till** värden till en array.

Transpose 2D Array

Rearranges the elements of **2D array** such that **2D array**[i,j] becomes **transposed array**[j,i].

- **Transponera** en array.

Översikt

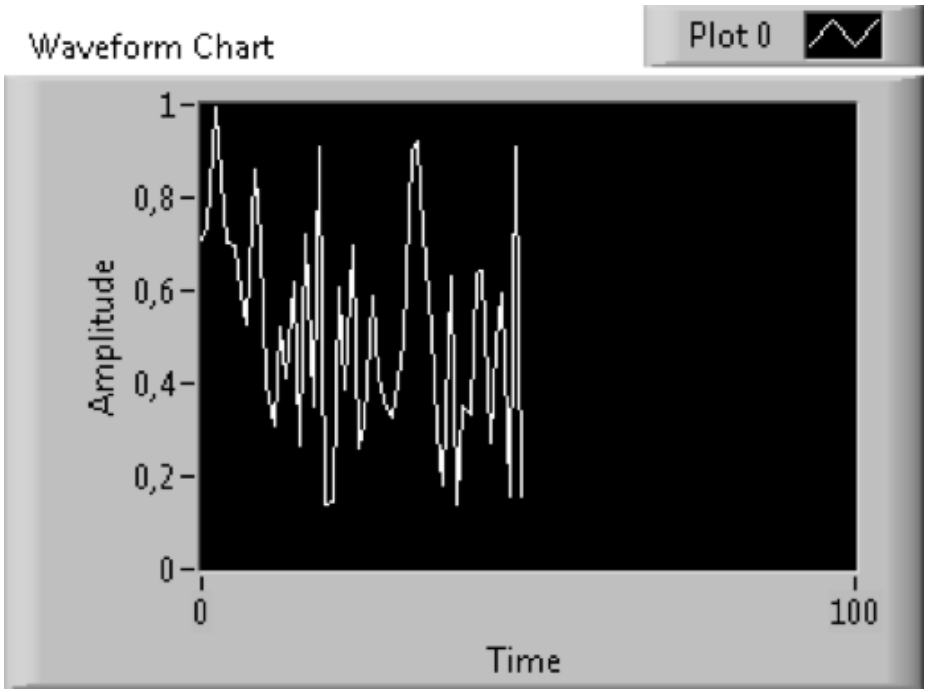
Plotta Grafer

Olika typer av **grafer** i LabVIEW:

- Waveform Chart
- Waveform Graph
- XY Graph

Waveform Chart

- Visar data **kontinuerligt** över tid.



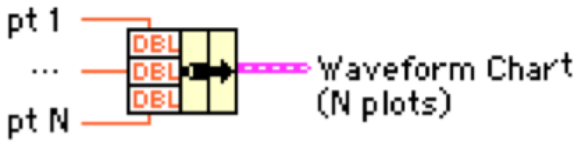
Waveform Charts:

Wire data directly to chart:

Data	Resulting Chart
Scalar	Single plot – 1pt
1D	Single Plot – 1 or more pts
WDT	Single Plot – 1 or more pts
2D	Multiplot – 1 or more pts

WDT (Waveform Data Type) includes timing info.

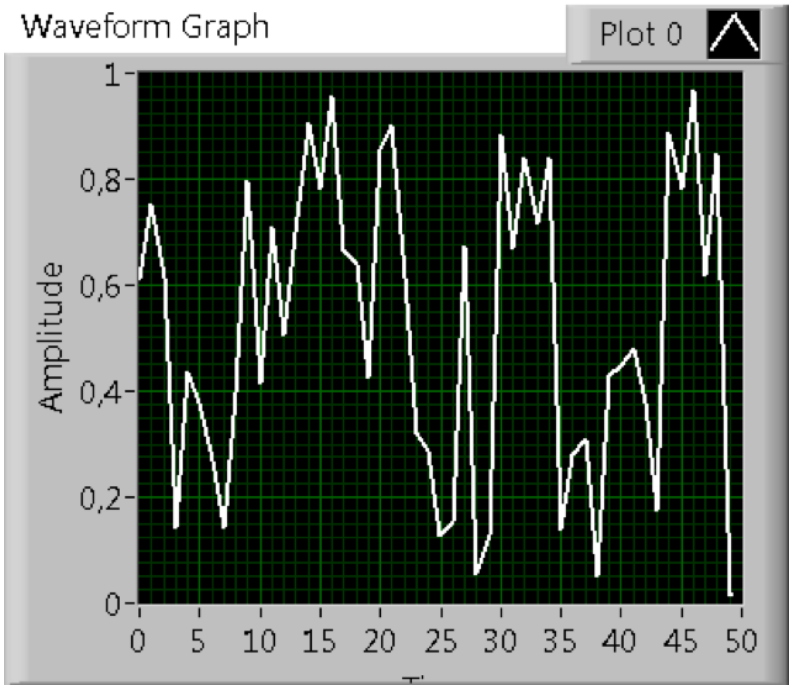
Or combine points with a bundle node:



Or use timing information in WDT.

Waveform Graph

- Plottar ett antal datapunkter.
- Möjlighet att specificera **starttid** och **steglängd**.

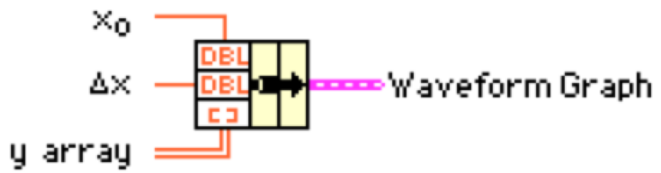


Waveform Graphs:

Wire data directly to waveform graph:

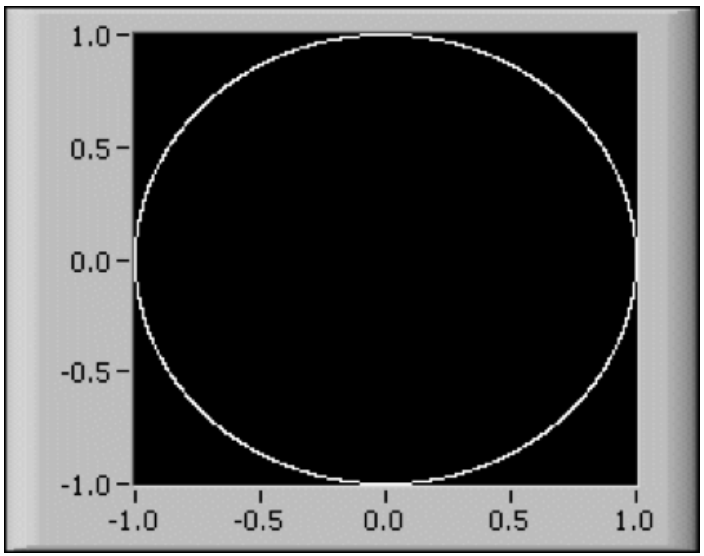
Y Array	Resulting Graph
1D	Single Plot
WDT	Single Plot
2D	Multiplot

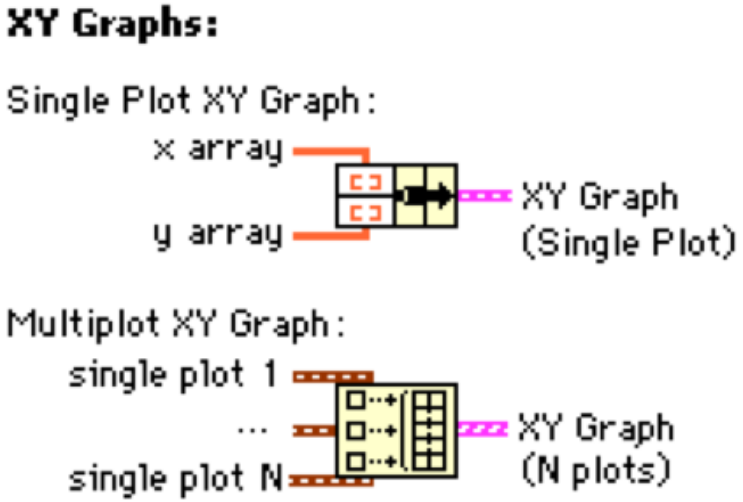
WDT (Waveform Data Type) includes timing info.
Others default to 0 for x_0 and 1 for Δx.
Combine timing information using a bundle node:



XY Graph

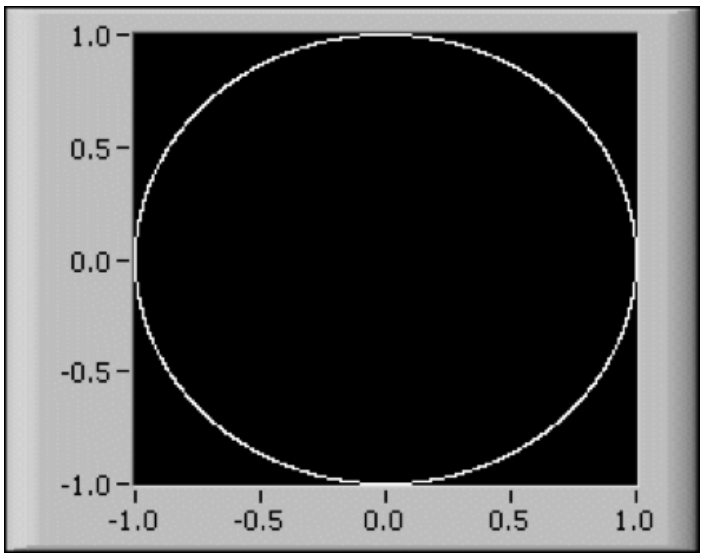
- Plottar x-värden mot y-värden.

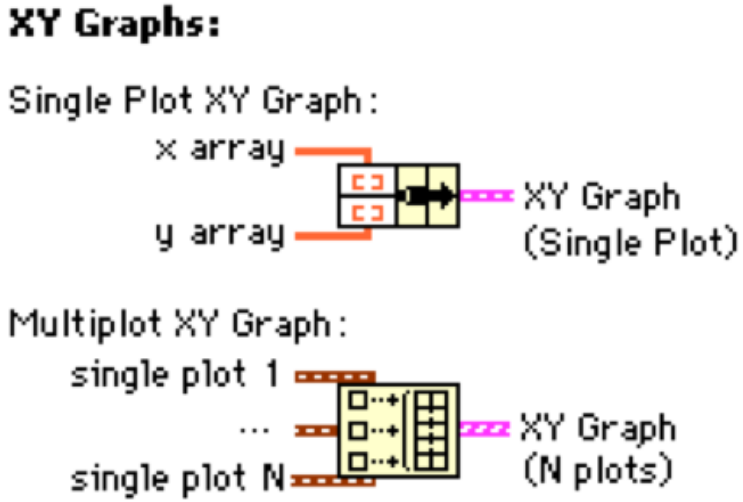




XY Graph

- Plottar x-värden mot y-värden.





Exempel 6: Plottning

Plotta funktionen

$$y = \sin(2\pi x), \; x = 0, 0.01, \ldots, 2$$

med Waveform Chart, Waveform Graph och XY Graph.

Sammanfattning

- Arbetsmiljö
- Programmeringsgrunder
- Strukturer (for-loop, while-loop, mm.)
- Arrays
- Grafer

Sammanfattning

- Arbetsmiljö
- Programmeringsgrunder
- Strukturer (for-loop, while-loop, mm.)
- Arrays
- Grafer

Mer information:

- Gratis LabVIEW-utbildning online för studenter och lärare (http://sweden.ni.com/webcasts/).
- Learn LabVIEW (http://www.ni.com/academic/students/learn-labview/).
- Introduction to LabVIEW (http://www.ni.com/getting-started/labview-basics/).

Sammanfattning

- Arbetsmiljö
- Programmeringsgrunder
- Strukturer (for-loop, while-loop, mm.)
- Arrays
- Grafer

Mer information:

- Gratis LabVIEW-utbildning online för studenter och lärare (http://sweden.ni.com/webcasts/).
- Learn LabVIEW (http://www.ni.com/academic/students/learn-labview/).
- Introduction to LabVIEW (http://www.ni.com/getting-started/labview-basics/).

LabVIEW är installerat på datorer i **JU-024**, **JU-025** och **JU-308-311**.