

Eerste kennismaking met LabVIEW

Tekst en illustraties gebaseerd op "Introduction to LabVIEW - an interactive tutorial"

Vrije bewerking door Herbert De Smet.

februari-maart 2000

Het meetinstrument

Eén van de meest populaire toepassingen van LabVIEW is het meten van een fysische grootte, zoals bijvoorbeeld een temperatuur, met behulp van de computer. In het (simplistische) voorbeeld hiernaast is iemand de temperatuur van een vloeistof aan het meten. Door LabVIEW te gebruiken, wordt de computer het meetinstrument.



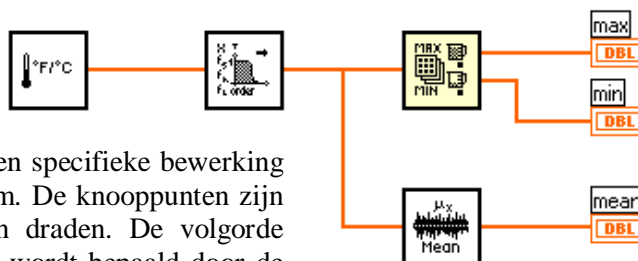
Virtuele Instrumenten (VI's)

Omdat de computer als meetinstrument dienst doet, ligt het voor de hand om het uitzicht van een 'normaal' meetinstrument zo goed mogelijk na te bootsen op het scherm van de computer. De bedieningsknoppen kan men uiteraard niet fysisch kopiëren, maar enkel simuleren met een pictogram. Het bedienen van die virtuele knoppen gebeurt met de muis, het toetsenbord of eventueel een 'touch screen'. Men noemt LabVIEW instrumenten daarom ook wel "Virtuele Instrumenten" of kortweg "VI's"

De bedieningsknoppen kan men uiteraard niet fysisch kopiëren, maar enkel simuleren met een pictogram. Het bedienen van die virtuele knoppen gebeurt met de muis, het toetsenbord of eventueel een 'touch screen'. Men noemt LabVIEW instrumenten daarom ook wel "Virtuele Instrumenten" of kortweg "VI's"

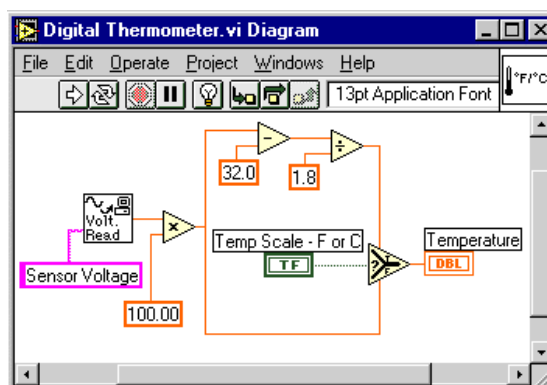
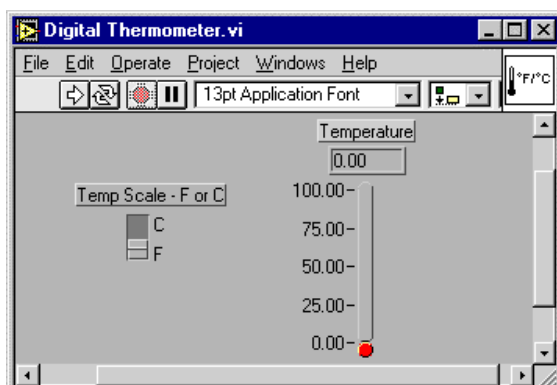
Grafische programmeertaal

LabVIEW omvat de grafische programmeertaal 'G'. De broncode ziet eruit als een flowchart in plaats van een lijst met instructies. Elk knooppunt in de flowchart vertegenwoordigt een specifieke bewerking die wordt voorgesteld door een pictogram. De knooppunten zijn met elkaar verbonden door middel van draden. De volgorde waarin het programma wordt uitgevoerd wordt bepaald door de data die via de draden van één knooppunt naar een ander lopen. Dit zal verder duidelijker worden.



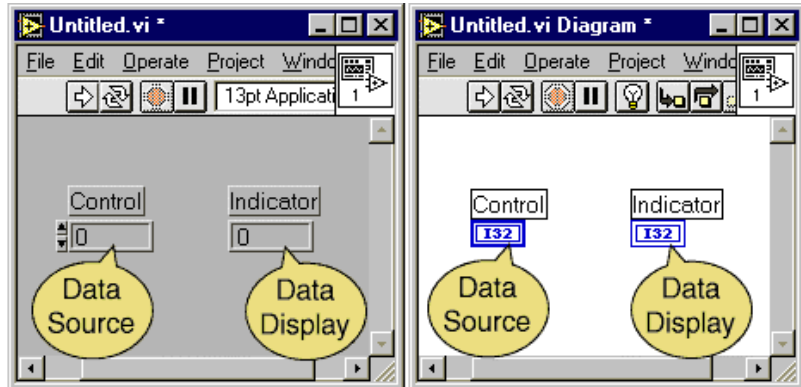
Frontpaneel en blokdiagram

De 2 voornaamste onderdelen van een VI zijn het frontpaneel en het blokdiagram. Het frontpaneel (links getoond in de figuur) is de user interface en het blokdiagram (rechts) bevat het programma (broncode).



'Controls' en 'Indicators'

Wanneer het frontpaneel actief is (op de voorgrond staat), heeft men de beschikking over het zogenaamde 'Controls' palet. Dit palet bevat een uitgebreide verzameling voorgeprogrammeerde bedieningsknoppen en indicatoren, maar kan ook uitgebreid worden met zelf gedefinieerde 'controls' of 'indicators'. Een 'control' komt overeen met een databron ('data source') in het blokdiagram; een indicator komt overeen met een data weergave ('data display') in het blokdiagram. Wanneer een 'control' of een 'indicator' wordt geplaatst op een frontpaneel, dan wordt automatisch de corresponderende 'data source' resp. 'data display' in het blokdiagram geplaatst (en vice versa). Een 'data source' of een 'data display' noemt men meer algemeen ook een 'terminal' in het jargon van de grafische programmeertaal. Merk op dat een 'control' een dik en een 'indicator' een dun kader heeft in het blokdiagram. Voor de rest zijn ze ononderscheidbaar.



Functies

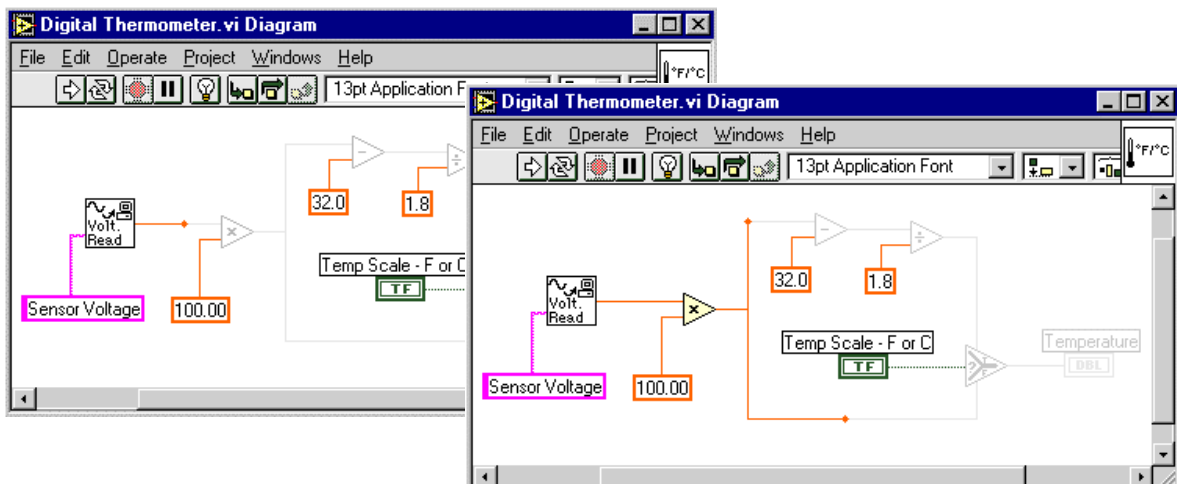
Het functie-palet is beschikbaar telkens wanneer het blokdiagram actief is (op de voorgrond staat). Dit palet bevat een indrukwekkend aantal ingebouwde functies en complete VI's en kan desgewenst zelfs nog worden aangevuld met zelfgemaakte VI's. Iedere functie of VI heeft een aantal inputs en een aantal outputs. Door het verbinden (met een draad) van alle nodige inputs met een control of met de output van een andere functie of VI, en van iedere gewenste output met een indicator of met de input van een andere functie of VI, kan men een compleet programma maken.



Dataflow

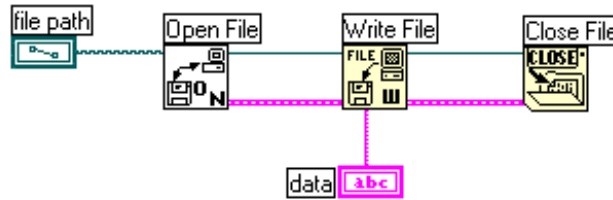
De volgorde waarin functies en VI's uitgevoerd worden, wordt bepaald door de manier waarop ze met elkaar en met de terminals verbonden zijn via de draden. Een functie of een VI wordt uitgevoerd zodra en slechts wanneer al zijn inputs een waarde hebben gekregen, m.a.w. als alle nodige data beschikbaar zijn. Pas nadat die functie of VI is uitgevoerd, zullen haar of zijn outputs op hun beurt een waarde gekregen hebben. Men zegt dat de grafische programmeertaal G een dataflow-gestuurde taal is. Dit in tegenstelling met de klassieke programmeertalen waarbij de volgorde van uitvoering bepaald wordt door de volgorde van de instructies in de broncode.

Ter illustratie wordt in de volgende 2 figuren getoond hoe in "Digital Thermometer.vi" de



vermenigvuldigingsfunctie slechts wordt uitgevoerd zodra zowel de output van de VI "voltage read" als het getal waarmee vermenigvuldigd moet worden (100) beschikbaar zijn. Pas daarna kan het resultaat gebruikt worden door de volgende functies.

In sommige gevallen is de volgorde van uitvoering zeer belangrijk, bijvoorbeeld bij output naar een file. Een file moet eerst geopend worden. Pas dan mag men erin schrijven en vervolgens moet ze gesloten worden. Elke andere volgorde geeft aanleiding tot fouten.



Datatypes

Indien u een kleurenversie van dit document heeft (bv. de pdf-versie), dan zal het al opgevallen zijn dat de draadjes en de terminals verschillende kleuren kunnen hebben. Deze kleur hangt samen met het type van data dat door de draad "loopt". Bijvoorbeeld: een oranje draad komt overeen met een "floating point" waarde (getal met decimale punt) en een groene draad komt overeen met een Boolese waarde ("True" of "False"). De meeste inputs van functies en VI's verwachten data van een welbepaald type. Indien data van een verkeerd type wordt aangeboden, dan zal het draadje automatisch veranderen in een zwarte stippellijn, hetgeen erop wijst dat er een conflict is opgetreden. Er bestaan evenwel functies die data van één type in een ander kunnen omzetten (bv. van integer naar floating point). In triviale gevallen doen de functies of VI's zelf deze "type casting". Bij wijze van waarschuwing wordt dan wel een klein vierkantje getoond aan de ingang waar de type casting nodig was.

Sub-VI's en connector-diagram

Een zelfgemaakte VI kan gebruikt worden als een sub-VI in een andere VI. Daartoe kan een VI worden voorgesteld als een icoontje, net als de bij LabVIEW meegeleverde VI's. Het icoontje wordt geplaatst op het blokdiagram van de aanroepende VI. Iedere "control" in de sub-VI kan voorgesteld worden als een input-pin en iedere "indicator" kan voorgesteld worden als een output-pin. De posities van de input- en de output-pinnen langs de omtrek van het icoontje worden bepaald door het connector-diagram. Het connector-diagram is net zo groot als het icoontje dat bij de VI hoort, en toont de posities van alle pinnen. Het legt ook de relatie vast tussen iedere pin en de corresponderende control of indicator. Door met de rechtermuisknop te klikken op het icoontje rechtsboven in het frontpaneel van een VI verkrijgt men een popup-menu. Met dit menu kan men ofwel het icoontje, ofwel het connector-diagram van deze VI zichtbaar maken en eventueel bewerken.

Quick Start guide

Na deze eerste kennismaking met LabVIEW is het tijd geworden om de "Quick Start guide" te raadplegen, waarmee men in minder dan 2 uur tijd een redelijke basiskennis van LabVIEW zou moeten kunnen verwerven. De Quick Start guide is net als dit document beschikbaar als een pdf-file en kan bekeken of afgedrukt worden met behulp van Acrobat Reader.