

Agilent VEE

Grafische Software-Entwicklungs-Umgebung
für Messtechnik und Test.

Agilent VEE • Grafische Software-Entwicklungs-Umgebung für Messtechnik und Test.
1. Auflage, für Agilent VEE ab Version 9.0

Die Informationen in diesem Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht.

Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise zu Fehlern ist der Herausgeber dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Die gewerbliche Nutzung der in diesem Buch gezeigten Modelle und Arbeiten ist nicht zulässig.

Microsoft Windows, Word und Excel sind eingetragene Warenzeichen der Microsoft Corp. ME/Meilhaus Electronic ist eineingetragenes Warenzeichen der Meilhaus Electronic GmbH. Weitere Produkt- und Firmennamen sind zum Teil eingetragene Warenzeichen der jeweiligen Firmen.

Redaktion Ernst-Michael Bratz

© 2009 by Meilhaus Electronic GmbH
Fischerstraße 2 • 82178 Puchheim, Germany
www.meilhaus.com
Alle Rechte vorbehalten
Printed in Germany

Inhalt

Einleitung	9
Grafisches vs. Quell-Code-orientiertes Programmieren	9
Compilerend vs. interpretierend	9
VEE Historie	10
Aufbau dieses Buches	10
Prinzipielle Arbeitsweise in Agilent VEE	12
1. Einführung in die Programmiertechnik	13
Grafisch programmieren	13
Die Agilent VEE Entwicklungs-Umgebung	14
Auswahl von Programmier-Objekten	14
Objekt-Eigenschaften	15
Verbinden von Objekten (Datenübergabe, Ablauffolge)	15
Löschen von Verbindungslinien	16
Daten- und Ablauf-Folge-Pins (Terminals)	16
Objekt- (Detail-) und Icon-View	16
■ Übung 1-1	17
Simulation eines verrauschten Sinus-Signales.	
2. Objekte, Funktionen, Benutzer-Oberflächen	21
Entwicklungs-Umgebung vs. Benutzer-Oberflächen	21
Benutzer-Objekte und Benutzer-Funktionen	22
Benutzer-Objekte	22
Benutzer-Funktionen	23
Einbinden externer Quell-Codes	24
■ Übung 2-1	24
Anpassung einer Kurve (Fit) an ein Mess-Signal (verrauschte Sinus-Schwingung), Fourier-Analyse.	
■ Übung 2-2	30
Aufbau einer Mathematik-Bibliothek.	
■ Übung 2-3	35
Zugriff auf eine (Mathematik)-Bibliothek (I).	

■ Übung 2-4	37
Zugriff auf eine (Mathematik)-Bibliothek (II).	
3. Schleifen, Verzweigungen, Records	39
Schleifen und Verzweigungen	39
Records und Arrays	41
■ Übung 3-1	42
Programmieren einer for-Schleife.	
■ Übung 3-2	44
Programmieren einer repeat/until-Schleife.	
■ Übung 3-3	50
Verzweigungen in einer Schleife (spezielle Elemente).	
4. Ein- und Ausgabe-Elemente	55
Die Eingabe-Elemente von Agilent VEE	55
Die Ausgabe-Elemente von Agilent VEE	56
■ Übung 4-1	57
Eingabe- und Abfrage-Struktur.	
■ Übung 4-2	61
Eigenschaften von I/O-Elementen verwenden.	
■ Übung 4-3	63
Kennenlernen und Ausprobieren der grafischen Displays von Agilent VEE.	
5. File-I/O	65
To/From File, DataSet, String, To Printer	65
■ Übung 5-1	66
Aufbau eines Spreadsheet-Files (Tabellenform).	
■ Übung 5-2	71
Lesen von einem Spreadsheet-File.	
■ Übung 5-3	73
Schreiben und Lesen von Records (DataSets).	

6. Debugging	77
Verschiedene Möglichkeiten der Fehlersuche	77
■ Übung 6	78
Programme Verwalten und Debuggen in Agilent VEE.	
7. Spezielle Elemente von Agilent VEE	83
Sequencer	83
Comparator	83
Bus IO Monitor	84
Globale Variablen	84
■ Übung 7-1	84
Testablauf mit Sequencer, Verwendung von globalen Variablen.	
■ Übung 7-2	90
Verwendung des Comparators.	
8. Offenheit von Agilent VEE	93
Datenaustausch	93
Einbinden von Programmteilen, Erweitern der VEE-Funktionalität	93
■ Übung 8-1	96
Ausführung von Programmen - Execute Program.	
■ Übung 8-2	98
Aufrufen von DLLs.	
■ Übung 8-3	99
Einfaches Beispiel mit ActiveX Controls.	
■ Übung 8-4	101
Einfaches Beispiel mit .NET Controls.	
■ Übung 8-5	103
ActiveX Automation: Fernsteuern von Microsoft Excel aus Agilent VEE heraus mit ActiveX.	
9. Schnittstellen bedienen, Geräte steuern	109

Externe Messdaten erfassen	109
Mess-Instrumente über Schnittstellen steuern	109
VISA/SICL	110
■ Übung 9-1	112
Ansteuerung eines GPIB-Gerätes.	
■ Übung 9-1, Teil A - Arbeiten mit fertigen Gerätetreibern und Gerätepanel	113
■ Übung 9-1, Teil B - Arbeiten mit einzelnen Treiberfunktionen (Component Driver)	115
■ Übung 9-1, Teil C - Arbeiten mit GPIB-Geräten über Direct I/O	116
■ Übung 9-2	118
Ansteuerung eines Agilent Oszilloskops aus der DSO/MSO 7000A Serie über USB und Ethernet/LXI.	
■ Übung 9-2, Teil A: Per USB	118
■ Übung 9-2, Teil B, Ergänzung: Per LXI/Ethernet	124
10. Einbinden von Datenerfassungskarten	127
Einbinden über Treiber/DLL	127
Das Meilhaus Electronic Treiber-System ME-iDS	128
Funktionsgruppen/Devices und Subdevices	129
Prinzipieller Aufbau eines Programmes mit ME-iDS	130
■ Übung 10-1	131
Ansteuern der Digital-Kanäle einer Meilhaus Electronic Multifunktions-Karte.	
■ Übung 10-2	137
Demo-Programm für Analog-Eingabe verstehen.	
11. Applikations-Beispiele, Ausblicke	141
Multithreading in Agilent VEE Pro 9.0	141
VEE Pro steuert Prüfplätze für den Test von PC-Karten	146
Einfach und schnell Agilent Mess-Geräte bedienen lernen	147
Einbinden der RedLab Serie in Agilent VEE	149
Installation	149
Einbinden in VEE	150
■ Übung 11-1	152
Temperatur messen mit RedLab TC und einem Thermoelement.	

Anhang	154
System-Voraussetzungen für VEE Pro und VEE Express	154
Vergleich VEE Pro, VEE Express und VEE Student	155
Interessante Web-Links für VEE-User	156
Funktions-Umfang von VEE Pro	157
Die wichtigsten VEE Pro Menüs	158
Die wichtigsten Grund-Daten-Typen von Agilent VEE	161
Index	162

Einleitung

„VEE-sionen werden Wirklichkeit-
So schnell setzen Sie Ihre Vorstellungen in die Software-Realität um.“

Haben Sie schon einmal mit einer der „klassischen“ Programmiersprachen gearbeitet - zum Beispiel einer C-Variante, Visual Basic, Delphi oder dem früheren Turbo Pascal? Dann sind Ihnen Begriffe wie Syntax, Datentypen (Ganzzahlen, Gleitkommazahlen, String etc.), Daten-Strukturen (Felder, Records, Arrays etc.) und Kontroll-Strukturen wie Schleifen und Verzweigungen (if/then, while etc.) geläufig. Aber auch wenn Sie noch nie mit einer Programmiersprache zu tun hatten, werden Sie VEE leicht verstehen und erlernen können. Denn Agilent VEE Pro ist eine visuelle oder **grafische Programmier-Umgebung** speziell für Messtechnik, Instrumentierung, zum Steuern, Testen, Kommunizieren und vieles mehr.

Grafisches vs. Quell-Code-orientiertes Programmieren

Grafisch programmieren bedeutet, dass Sie sehr viel weniger Programmcode von Hand eintippen müssen, als bei den klassischen Programmiersprachen. Dementsprechend müssen Sie sehr viel weniger über Syntax und Kontrollstrukturen auswendig wissen - wichtig ist, dass Sie deren Bedeutung verstanden haben. Denn in VEE programmieren Sie sehr intuitiv durch „Verdrahten“ von Funktions-Blöcken auf der Arbeitsfläche: Sie verwenden einfach die Maus und verbinden Icons. Dabei entsteht eine Struktur, die einem **Fluss- oder Block-Diagramm** ähnlich ist. Dies hat zwei entscheidende Vorteile: Zum einen kommt es der Denkweise des Technikers und Ingenieurs sehr viel näher als die oft kryptischen Anweisungen der codeorientierten Sprachen. Zum anderen sind die entstandenen Programme auch nach längerer Zeit sehr viel einfacher nachzuvollziehen, als reiner Quell-Code.

Warum ist VEE Pro eine Programmier-Umgebung speziell für **Messtechnik, Instrumentierung und zum Steuern, Testen, Kommunizieren**? Diese Frage ist sehr einfach zu beantworten: VEE Pro umfasst umfangreiche Bibliotheken, die auf Anwendungen genau in diesen Bereichen zugeschnitten sind. Dazu gehört zum Beispiel auch MATLAB Script, das ein integraler Bestandteil der VEE-Umgebung ist. Sie können mit VEE Pro **Messdaten erfassen**, indem Sie PC-Einsteckkarten (PCI-Express, PCI, CompactPCI/PXI und andere), Mess-Module oder Messinstrumente über Ethernet, GPIB, RS232 oder USB steuern. Auch für **Analyse, Verarbeitung und Export** der Daten bieten die VEE-Bibliotheken einen riesigen Fundus an Möglichkeiten. Außerdem gestalten Sie mit VEE sehr attraktive und an der technischen Praxis orientierte **Visualisierungen und Nutzerinterfaces** mit Displays, Drehknöpfen, Schaltern etc. Jedes VEE-Programm können Sie als „Runtime-Version“ (d. h. ein ohne die komplette Entwicklungsumgebung lauffähiges VEE-Programm) verwenden oder weitergeben.

Compilierend vs. interpretierend

Die Übersetzung des vom Menschen lesbaren Programmcodes in den Maschinencode kann vor oder während der Ausführung des Programmes erfolgen. Dementsprechend unterscheidet man in compilierende und interpretierende Sprachen: Wird der Quelltext des Programmes als Ganzes vor der Ausführung übersetzt, spricht man von einem Compiler. Kommt hingegen ein Programm zum Einsatz, das einen Quellcode zur Laufzeit einliest, analysiert und ausführt (also nicht in eine auf dem

System direkt ausführbare Datei umwandelt), spricht man von einem Interpreter oder „Laufzeit-Compiler“. VEE hat keinen Compiler, der abgeschlossenen Maschinencode erzeugt und ist daher eher den interpretierenden Sprachen zuzuordnen.

VEE Historie

VEE Pro wurde ursprünglich als **HP-VEE** von **Hewlett-Packard** entwickelt. In den Jahren 1999 - 2000 entstand die Firma **Agilent Technologies** als „Ausgliederung“ von fünf Abteilungen der 1939 gegründeten Firma Hewlett-Packard und firmierte zu einem selbstständigen Unternehmen. Dabei wurde aus HP-VEE (Version 5) **Agilent VEE** (Version 6). Liefen die ersten Versionen von HP-VEE noch unter Windows 3.1/3.11, 95 und NT, so ist VEE Pro (Version 9 und höher) heute auf den aktuellen Windows-Versionen XP, 2000, Vista und folgende zuhause. Zudem gab es ursprünglich auch Versionen für HP-UX und Solaris

Von Version zu Version wurden Funktionen hinzugefügt oder verbessert und neue Standards in die Entwicklungsumgebung aufgenommen. Es sollen an dieser Stelle nur einige der wichtigsten genannt werden: Die **Web-Server- und ActiveX-Funktionalität** wurde in der Version 5 ergänzt. Ab Version 6 ist es möglich, LabWindows/CVI-Gerätetreiber in VEE einzubinden, wodurch insgesamt **weit über 900 Messgeräte unterstützt** werden. Zudem wurde ab dieser Version **MATLAB Skript ein fester Bestandteil von VEE** und neben **GPIO** und **RS232** können auch die Schnittstellen **USB** und **LAN** zur Gerätesteuerung genutzt werden. In der Version 7 ist erstmals die Einbindung des **Microsoft .NET-Framework** realisiert und in der Version 9 sind Funktionen zur Nutzung der **Parallelprozessor-Technologien Multithreading und Multi-Core** hinzugefügt worden.

Einige Zeit lang gab es neben der Vollversion von VEE auch eine Lowcost-Variante HP-VEE/VEE OneLab, die jedoch mit der VEE Pro Version 6.2 eingestellt wurde. Ab der Version 8 wurde eine spezielle Variante von VEE für Schüler/Studenten (VEE Student) sowie für den ausschließlichen Einsatz mit den Agilent USB-Mess-Modulen (VEE Express) eingeführt.

Weitere Informationen zu den Systemvoraussetzungen und VEE Varianten finden Sie im Anhang.

Aufbau dieses Buches

Das vorliegende Buch ist nicht als Handbuch oder Nachschlagewerk zu Agilent VEE gedacht. Im Vordergrund stehen vielmehr Übungen, mit denen sich der VEE-Einsteiger schnell in Programmier-Techniken und Vorgehensweisen einarbeiten kann.

Die Übungen orientieren sich an der Messtechnik-Praxis. Für einige der Aufgaben werden bestimmte Messgeräte, Module oder Messkarten eingesetzt. Falls Sie diese nicht zur Verfügung haben sollten, können Sie viele Schritte auch ohne angeschlossenes Instrument bzw. Hardware durchführen, um die prinzipielle Arbeitsweise kennen zu lernen und auf andere Geräte-Typen zu übertagen. Anhand der Abläufe können Sie das Gelesene gleich direkt in VEE ausprobieren und erhalten dabei noch zusätzliche Hintergrund-Infos. Die Übungen haben immer diesen Aufbau:

- **Aufgabenstellung:** Das Thema der Übung in einem Satz.
- **Inhalte dieser Übung:** Die Lerninhalte der Aufgabe.
- **Schritt für Schritt:** Die Übung als „Kochrezept“ zum Nachmachen.

Die Funktionalität von VEE ist so umfangreich, dass wir Ihnen in diesem Buch natürlich nicht alle Möglichkeiten und jedes Detail der Entwicklungsumgebung vorstellen können. Im Anhang finden Sie einige Internet-Adressen, unter denen Sie als erfahrener VEE-Anwender weitere Infos finden.

Das Buch ist in die folgenden Abschnitte unterteilt:

1. **Einführung in die Programmieretechnik**
Arbeiten mit den Agilent VEE Menüs, Verbinden der Objekte, Ziehen und Löschen von Fäden, Detail- und Icon-View, Gestalten von Anzeigen, Generieren von virtuellen Mess-Daten, Darstellung mehrerer Datensätze.
2. **Objekte, Funktionen, Benutzer-Oberflächen**
Bediener-Oberflächen, Erstellen von Objekten, Arbeiten mit mehreren Oberflächen, Mathematik-Bibliothek, Arbeiten mit Funktionen, freie Formeleingabe, Zugriff auf externe Agilent VEE Funktionen.
3. **Schleifen, Verzweigungen, Records**
For- und andere Schleifen, Bedeutung der START-Taste, Anwendung des Collectors, dynamische I/O-Elemente, Zugriff auf Record-Variablen, Repeat Until Break, Aktionsverzweigungen, Bedeutung des Confirm-Knopfes.
4. **Ein- und Ausgabeelemente**
Dialog-Boxen, Alarm-Elemente, Eingabelisten, Sichern eines Programmes, Stoppen eines Programm-Ablaufes, Einbindung von Grafiken, grafische Displays.
5. **File I/O**
Aufbau eines Files in Tabellenform, Speichern unterschiedlicher Datentypen, Lesen von Files, Schreiben und Lesen von Records und Data-Sets.
6. **Debugging**
Arbeiten mit Show Execution Flow und Show Data Flow, Arbeiten mit Breakpoints und Step-Ausführung.
7. **Spezielle Elemente von Agilent VEE**
Sequencer, Comparator, Bus-I/O-Monitor, Globale Variablen.
8. **Offenheit von Agilent VEE**
Externe Programme starten, Einbindung von DLLs, ActiveX, .NET etc.
9. **Schnittstellen bedienen, Geräte steuern**
GPIB (IEEE488/HP-IB/IEC-Bus), USB, LXI, Anmelden von Geräten, Instrument Finder, Arbeiten mit fertigen Gerätetreibern, Ansteuern von Geräten ohne Treiber (Direct I/O).
10. **Einbinden von Datenerfassungskarten**
Struktur des Meilhaus Electronic Treiber-Systems ME-iDS, ME-Board - das „ME-iDS-Menü“ in VEE Pro, Programmieren von Messabläufen auf Basis fertiger Demo-Programme.
11. **Applikations-Beispiele, Ausblicke**
Interessante Fachartikel zu Agilent VEE, Applikationsberichte, Ausblicke.

Prinzipielle Arbeitsweise in Agilent VEE

Vereinfacht dargestellt läuft das Programmieren mit Agilent VEE aus der Sicht Ihrer praktischen Anwendung in diesen 4 Schritten ab:

- **Messen:** Mithilfe der VEE-Bibliotheken programmieren Sie das Erfassen Ihrer Messdaten, zum Beispiel in dem Sie eine PC-Einsteckkarte einbinden oder Messgeräte über die Schnittstellen GPIB, Ethernet oder USB ansteuern.
- **Analysieren:** Auch hier nutzen Sie die VEE-Bibliotheken, zum Beispiel das integrierte MATLAB Script, um Ihre erfassten Daten zu analysieren oder für die weitere Verarbeitung zu exportieren.
- **Visualisieren:** Mit den VEE-Display-Funktionen stellen Sie die erfassten Daten am Bildschirm visuell dar. Bei Bedarf generieren Sie komfortable Benutzer-Oberflächen, um Ihr Programm als selbstständige Applikation zu verwenden und auch dritten zugänglich zu machen.
- **Debugging + Start:** In diesem Schritt sorgen Sie dafür, dass Ihr Programm fehlerfrei läuft. Falls Sie es unabhängig von der Entwicklungsumgebung zum Einsatz bringen möchten, generieren Sie eine Runtime-Version.

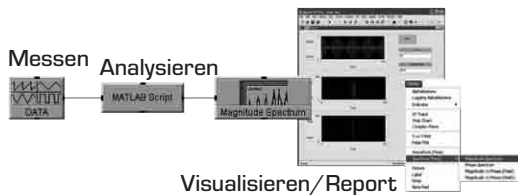
① **Schritt 1: Messen** - Konfigurieren Sie ein „Measure“ Objekt für beliebige Signale, z. B. RF, Temperatur, Vibration, Puls, Spannung.



② **Schritt 2: Analysieren** - Wählen Sie hierzu aus über 500 Analyse-Funktionen inkl. MATLAB Script, um Ihre Daten zu verarbeiten



③ **Schritt 3: Visualisieren** - Wählen Sie eine der vielen möglichen Darstellungs-Arten für Ihre Daten...



...oder exportieren Sie die Werte, z. B. in Excel

④ **Schritt 4: Debugging und ausführen.** Erzeugen Sie unbegrenzt Runtime-Versionen Ihrer Applikation.

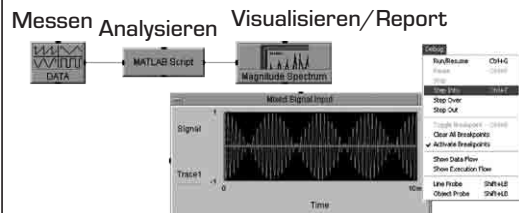


Bild 1-1: In 4 Schritten zum Mess-Programm in Agilent VEE

1. Einführung in die Programmierertechnik

Grafisch programmieren

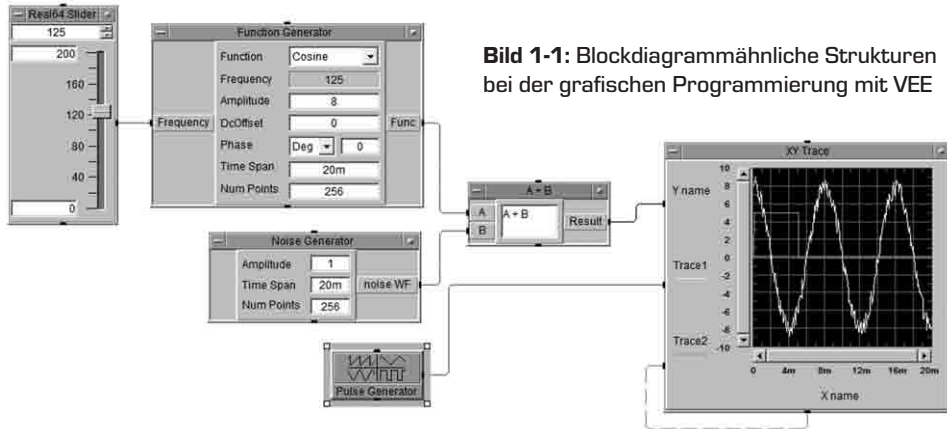


Bild 1-1: Blockdiagrammähnliche Strukturen bei der grafischen Programmierung mit VEE

In der Einleitung haben Sie bereits gelesen: Agilent VEE ist eine **grafische Programmier-Umgebung**. Statt herkömmlichen Quell-Code einzugeben, verbinden Sie grafische Objekte zu einer Art Blockdiagramm. Agilent VEE erzeugt keinen Quell-Code - die grafischen Objekten repräsentieren bestehenden Code.

Agilent VEE umfasst alle für Messprogramme erforderlichen Funktionalitäten:

- **Ansteuern von Messelektronik**
Ethernet/LXI, USB, GPIB- (IEEE488-/HP-IB-/IEC-) Schnittstelle und GPIB-Geräte, RS232, VXI-Systeme, PC-Einsteckkarten.
- **Mathematik- und Analyse-Bibliothek**
Arithmetische und trigonometrische Funktionen, digitale Filter und Fenster, FFT, freie Eingabe von Formeln u. v. m.
- **Ein-/Ausgabe von Daten**
Alphanumerische und grafische Darstellungsarten, File-I/O.
- **Strukturiertes Programmieren**
Programmieren von (Test-)Abläufen, Schleifenstrukturen einer Hochsprache, Bedingungen etc., Test-Sequencer
- **Gestalten von Benutzer-Oberflächen**
Ansprechende, intuitiv zu bedienende Oberflächen mit Displays, Knöpfen, Reglern etc.

Die Agile VEE Entwicklungs-Umgebung

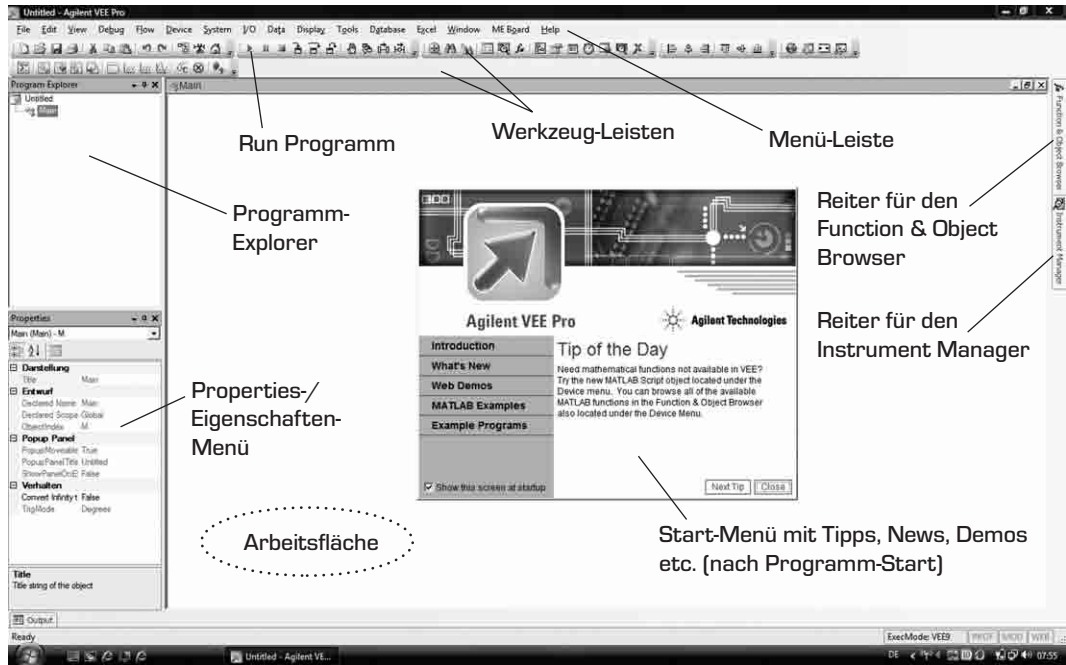


Bild 1-2: Die VEE Entwicklungs-Umgebung

- Die **Menüleiste** enthält alle Parameter-Eingaben, Struktur-Elemente und Funktionen zur Erstellung eines Programms (siehe auch Anhang: **Die wichtigsten VEE Pro Menüs**).
- Die **Werkzeuggeste** enthält Knöpfe zum schnelleren Zugriff auf bestimmte Funktionen, die Sie natürlich alle auch in den Menüs finden, zum Beispiel die Debugging-Optionen.
- In der **Arbeitsfläche** wird das Programm in Form eines Blockdiagramms aufgebaut.

Das Erstellen und Ausführen eines Programms erfolgt in einem Fenster. Optional kann eine sogenannte Bedieneroberfläche angelegt werden, welche nur Ein-/Ausgabeelemente enthält (wird im Folgenden noch genauer beschrieben).

Auswahl von Programmier-Objekten

Die **Menü- und Werkzeuggeste** enthalten alle Objekte, die Sie benötigen, um ein Programm zu erstellen. Wählen Sie das gewünschte Objekt aus den Menüs und Untermenüs aus. Das gewählte Objekt kann dann mit der Maus im Arbeitsbereich platziert werden.

Im Weiteren wird diese Vorgehensweise mit folgender Syntax beschrieben (Beispiel: Auswahl des Funktionsgenerators, **Bild 1-3**, rechts):

Device ➔ **Virtual Source** ➔ **Function Generator**

Die (Unter-)Menüs der Menüleiste lernen Sie im Lauf des Buches kennen. Hilfe zu verschiedenen Themen erhalten Sie über das VEE **Help**-Menü, zum Beispiel mit

Help ➔ **Contents and Index...**

(siehe **Bild 1-4**, unten).

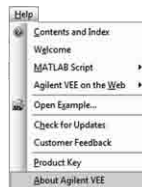


Bild 1-4 (oben): Das Help-Menü



Bild 1-3 (rechts): Auswahl des virtuellen Funktionsgenerators aus dem Device-Menü

Objekt-Eigenschaften

VEE-Objekte haben verschiedene Eigenschaften (englisch „Properties“), die unter anderem zum Beispiel das Erscheinungsbild prägen (Schrift, Farbe etc.) oder die Funktion selbst und die Möglichkeiten der Einbindung in ein Ablaufdiagramm betreffen. Diese Objekt-Eigenschaften können Sie über das **Properties**-Menü am linken unteren Rand der Entwicklungsumgebung verändern. Falls das **Properties**-Menü nicht eingeblendet sein sollte, klicken Sie

View ➔ **Properties**

oder klicken Sie mit der rechten Maustaste auf ein Objekt und wählen Sie **Properties** aus dem aufgeklappten Menü.

Verbinden von Objekten (Datenübergabe, Ablauffolge)

Zwischen Objekten müssen meistens Daten übergeben werden. Hierzu befinden sich links und rechts an den Objekten Datenübergabe-Pins (**Bild 1-5**). Die Verknüpfung erzeugen Sie, indem Sie mit der linken Maustaste in die Nähe des Pins klicken. Der Mauszeiger zieht dann einen Faden hinter sich her. Führen Sie die Maus zum gewünschten Pin des zweiten Objektes und klicken Sie nochmals in dessen Nähe. Einen unerwünschten Faden am Mauszeiger schalten Sie wieder ab, indem Sie die linke Maustaste doppelklicken (oder Escape/esc-Taste).

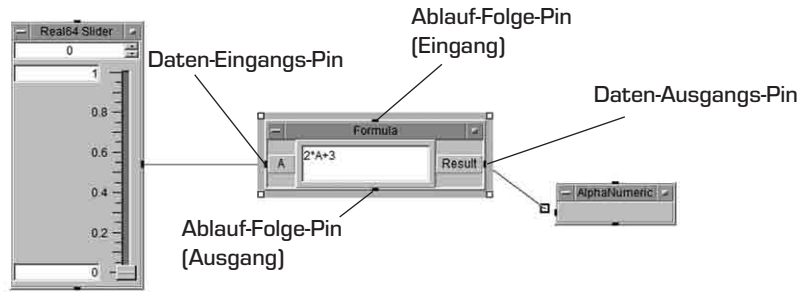


Bild 1-5: Verknüpfen von Objekten zur Datenübergabe

Löschen von Verbindungslinien

Eine bestehende Verbindung löschen Sie mit der Wahl des Punktes

Edit ➔ Delete Line

und Anklicken der zu löschenden Linie oder kürzer: Ctrl(Strg) und Shift-Tasten drücken und mit der Maus die Linie anklicken, oder Rechts-Klick auf die Linie und **Delete Line** auswählen.

Daten- und Ablauf-Folge-Pins (Terminals)

Die Daten-Pins haben wir bereits kennen gelernt: Sie dienen zur Übergabe von Daten. An der Ober- bzw. Unterseite von Objekten befinden sich Ablauf-Folge-Pins (auch Kontroll-Pins genannt). Diese werden bei vielen (einfachen) Programmen nicht gebraucht. Sie definieren im Bedarfsfall die Reihenfolge, in der Agilent VEE Objekte abarbeitet. Mehr dazu im weiteren Verlauf des Buches.

Ein wichtiger Punkt unter den Objekt-Eigenschaften (rechter Mausklick auf Objekte) ist die Einführung zusätzlicher Terminals an ein Objekt. Dies können sowohl Daten-Terminals (zum Beispiel mehrere Datensätze in einer Grafik darstellen) oder Ablauf-Folge-Terminals sein (zum Beispiel Definition bestimmter Eigenschaften aus dem Programmablauf).

Objekt- (Detail-) und Icon-View

Jedes VEE-Objekt bietet zweierlei Ansichten: Eine Objekt- oder Detail-Ansicht (großes Bild) und eine Icon-Ansicht (Symbolbild). Sie schalten zwischen diesen Ansichten hin und her durch Anklicken des Feldes rechts oben in der Menüleiste der Funktion (➔ Icon-Ansicht) bzw. Doppelklick auf das Symbol (➔ Detail-Ansicht).

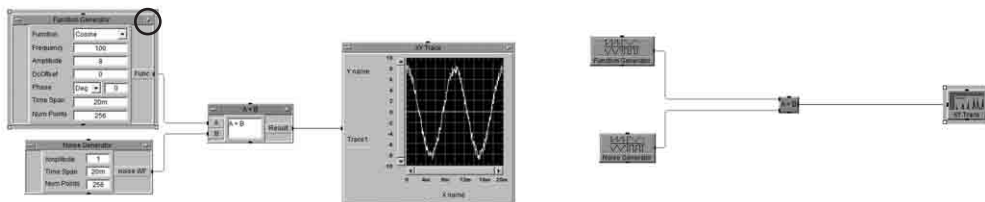


Bild 1-6: Detail-Ansicht (oder Object-View) und Icon-View des gleichen Programms

■ Übung 1-1

■ Aufgabenstellung:

Simulation eines verrauschten Sinus-Signales.

■ Inhalte dieser Übung:

- Verknüpfen von Objekten (Arbeiten mit Verbindungslinien).
- Erstellen und Starten einer Applikation.
- Editieren von Objekten und Ändern von Objekt-Eigenschaften.
- Objekt- und Icon-View.

■ Schritt für Schritt:

- 1) Als ersten Schritt holen Sie einen (virtuellen) Funktionsgenerator unter **Device** ➔ **Virtual Source** ➔ **Function Generator**.
- 2) Positionieren Sie den Funktionsgenerator in der linken oberen Ecke des Monitors durch Anklicken und Ziehen an der Menüleiste (linke Maustaste gedrückt halten). Ändern Sie den Wert bei **Frequency** auf 100 Hz ab.
- 3) Holen Sie unter **Device** ➔ **Virtual Source** ➔ **Noise Generator** simuliertes Rauschen. Positionieren Sie dieses Objekt unter dem Funktionsgenerator.
- 4) Holen Sie **(Function & Object Browser)** ➔ **Operators** ➔ **Arithmetic** ➔ **f_x +**

Typ-Anpassung

Viele Agilent VEE Objekte besitzen ein hohes Maß an Eigenintelligenz. Beispielsweise kann die Funktion a+b zwei (oder mehrere) Zahlen addieren, (mehrere) Strings zusammenhängen oder die Elemente von Feldern addieren.

Die Typanpassung erfolgt normalerweise automatisch. Sie kann aber auch nach Doppelklick auf ein Eingangs-Terminal gezielt definiert werden.

- 5) Positionieren Sie **a+b** rechts vor den beiden Objekten.
- 6) Holen Sie eine graphische Anzeige, für diesen Fall eine **y(n)**-Darstellung (n bedeutet die laufende Nummer der Werte) mit **Display** ➔ **XY Trace**.
- 7) Verknüpfen Sie die Datenausgangspins des Funktions- und Noise-Generators mit den Pins von **a+b**, den Ausgang von **a+b** mit dem Eingang des Displays.
- 8) Drücken Sie in der Werkzeugleiste den Knopf **Run**. Ihr Monitorbild sollte wie folgt aussehen (**Bild 1-7**):

1. Einführung in die Programmierertechnik

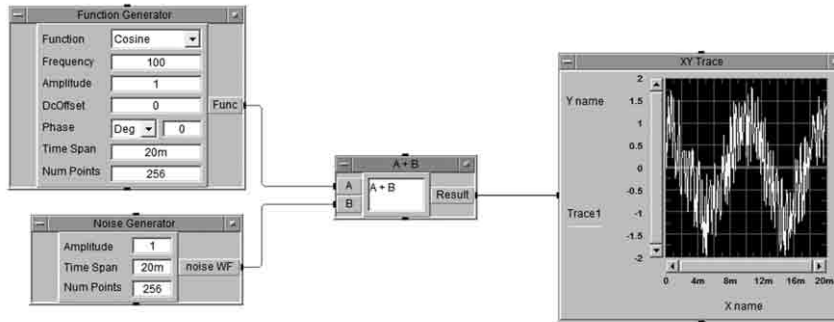


Bild 1-7: Ihr erstes VEE-Programm!

- 9) Ein deutliches Signal ist nicht erkennbar, da die Signal-Amplitude und die Rausch-Amplitude gleich groß sind. Sehen Sie sich die Parameter des Funktions- und des Rausch-Generators an. Ändern Sie die Signal-Amplitude (zum Beispiel auf 8), um ein akzeptables Signal-zu-Untergrund-Verhältnis zu simulieren.

- 10) Probieren Sie die Umschaltung zwischen Detail- und Icon-Ansicht (Sie erinnern sich: Umschalten durch Anklicken des Feldes rechts oben in der Menüleiste des Objekts bzw. Doppelklick auf das Icon).

- 11) Machen Sie sich mit den einzelnen Menüpunkten des Eigenschaften-/Properties-Menüs links unten in der Entwicklungsumgebung vertraut. Probieren Sie auch den rechten Mausklick auf ein VEE-Objekt.

Ändern Sie einige Eigenschaften wie Schrift, Darstellung der Kurve, Cursor.

- 13) Wählen Sie aus dem Eigenschaften-Menü der Grafik

Add Terminal ➔ **Control Input...**

In dem erscheinenden Menü klicken Sie auf

Auto Scale ➔ **OK**

Es erscheint an der Grafik ein neues Terminal. Ziehen Sie eine Verbindung zwischen dem Ablauffolge-Pin der Grafik und dem neuen Terminal (Kontroll-Verbindungen werden von Agilent VEE gestrichelt gezeichnet). Die Funktion ist folgende: Nach dem ersten Zeichnen des Datensatzes wird der Ablauffolge-Pin der Grafik aktiviert. Dieser setzt den Eingang Auto Scale, so dass sich bei dieser Verschaltung jede Grafik selbst skaliert.

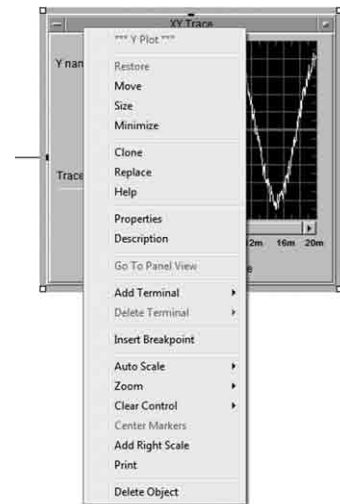


Bild 1-8: Objekt-Eigenschaften

- 14) Bewegen Sie als nächstes den Mauszeiger auf den linken Rand des Funktionsgenerators und drücken Sie Ctrl-a (Strg-a) um ein Terminal anzufügen. Wählen Sie

Frequency ➔ **OK**

Doppelklicken Sie auf das Terminal und ändern den Namen in `freq`. Alternativ könnte ein Eingangs-Terminal auch über das Eigenschaften-Menü erzeugt werden.

- 16) Holen Sie **Data** ➔ **Continuous** ➔ **Real64 Slider** und positionieren Sie das Element links neben den Funktionsgenerator. Setzen Sie die maximale Frequenz auf 200. Verbinden Sie den Ausgang des Reglers mit dem Frequenzeingang des Generators und stellen Sie einen beliebigen Wert ab mindestens 1 am Regler ein.
- 17) Holen Sie mit **Device** ➔ **Virtual Source** ➔ **Pulse Generator** einen simulierten Rechteckpuls. Öffnen Sie das Eigenschaften-Menü der XY-Trace Grafik und erzeugen Sie mit **Add Terminal** ➔ **Data Input** einen zweiten Eingang für die Daten des Pulsgenerators. Übergeben Sie die Daten an die Grafik. Setzen Sie die Höhe des Pulses (High) auf 5.
- 18) Auch die Gesamt-Erscheinung der VEE Entwicklungsumgebung können Sie verändern. Wählen Sie dazu **File** ➔ **Default Preferences** ➔ **Colors & Fonts** ➔ **Color** ➔ **<Farbauswahl>** und ändern Sie zum Beispiel die Farbe des Display-Hintergrundes.
- 19) Drücken Sie in der Werkzeugleiste **Run**. Das Schirmbild sollte in etwa wie folgt aussehen (ändern Sie gegebenenfalls die Werte des Pulsgenerators wie Amplitude, Anstiegszeit etc.):

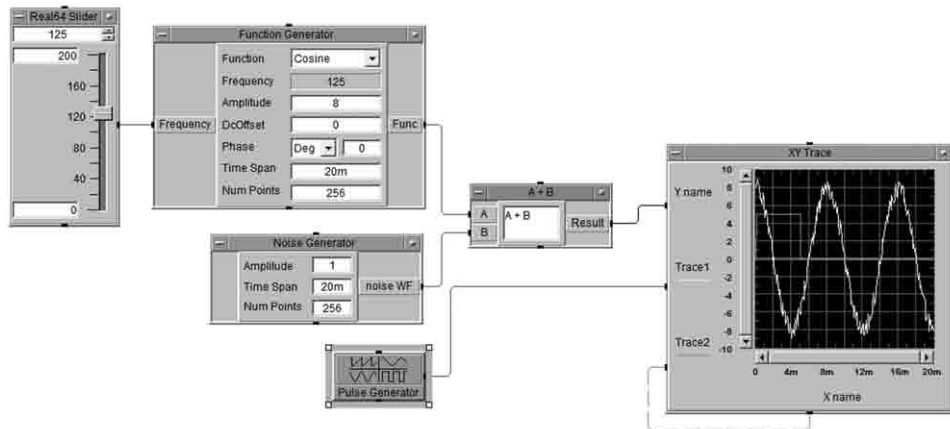


Bild 1-9: Etwa so sollte Ihr Programm am Ende der ersten Übung aussehen

- 20) Speichern Sie Ihr Programm mit **File** ➔ **Save as...** zum Beispiel unter `...\ex1_1.vee`