

## EDTest - Electronic Device Testsystem

Testsequenzer für die Testautomatisierung

Universelle-Testsequenzer-Software zur automatischen Testablauf-Steuerung von Test-Protokollierung/ Dokumentation, Datenbank, Statistik, Seriennummernverwaltung, Prüferverwaltung, Instrumentenverwaltung/ Wartung, usw.

Funktionsprüfung, Design-Verifizierung, Kalibrierung/Justage, Serien- und Rückläufer-Test für Bauelemente, Baugruppen und Geräte.

The screenshot displays the EDTest V2.9.0.25 software interface. It features a main window with a menu bar (Projekt, Protokoll, Statistik, Extras, Ansicht, Werkzeuge, Info) and a toolbar. The central area is divided into several panels:
 

- EDTest-Demo:** Shows test parameters like 'LosBez', 'Artikel-Nr.', and 'Serien-ID' (WT-0149). A table lists test results with columns for 'Pass/FAIL', 'Name', and 'Info'.
- Ablauf-Steuerung:** Includes a 'Testablauf-Mode' dropdown set to 'Standard-Test' and a 'Start' button.
- Aktueller Test:** A table showing the current test steps with columns for 'Nr.', 'Name', 'Kommando', 'Grenzwerte', 'Ergebnis', 'Einheit', 'P/F', 'Ergebnis-Kommentar', and 'Infozettel'.
- Testprogramm bearbeiten:** A detailed view of the test program with tabs for 'Testablauf', 'Makro', 'Sequenzen', and 'Makros (Sequenz)'. It includes a table for test steps with columns for 'Status', 'Aktiv', 'Nr.', 'Name', 'Kommando', 'Par-Block-1', and 'Par-Block-2'.
- Distribution TS-11:** A graph showing a normal distribution curve for 'Values [A] (Div.50)' with a 'Count' on the y-axis.

EDTest-SPC Statistic-Process-Control v3.0.0.5

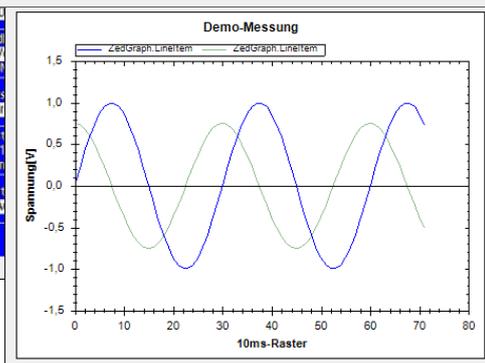
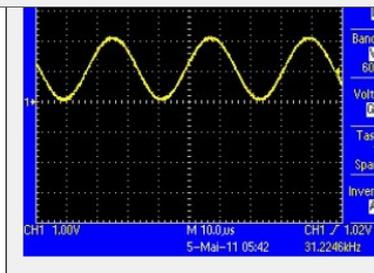
PROZESS FILTER SYSTEM

EDTest statistische Prozess-Kontrolle

Status Online 13:07  
Waiting 5 Interval [min]

Fertigungs-Prozess	Datum (von)	Datum (bis)	Los	Variante	ControllerID	Prüfinge	Pass-Anzahl	FAIL-Prüfungen	Ausfallrate 9[%]	FPY 65[%]	Handlingzeit
AFV2					1011-00	489	481	8	1.64	98.36	5.3s
Thermolüfter_DC_Modul		22/2016				201	200	33	0.50	97.01	0.0s
CPV2						368	345	23	6.25	93.75	0.0s
EDTest_Demo						6	6	2	0.00	66.67	12.2s
BZX586VB_0406_00950						0	0	0	-	-	-
VARV2				Version D (14)		0	0	0	-	-	-

Reparatur Beenden



**WesTest GmbH**  
Hegelsbergstraße 21  
34127 Kassel  
Tel.: 0561 98975-0  
Fax: 0561 98975-90

info@westest.de  
www.westest.de



## EDTest – Sequenzer

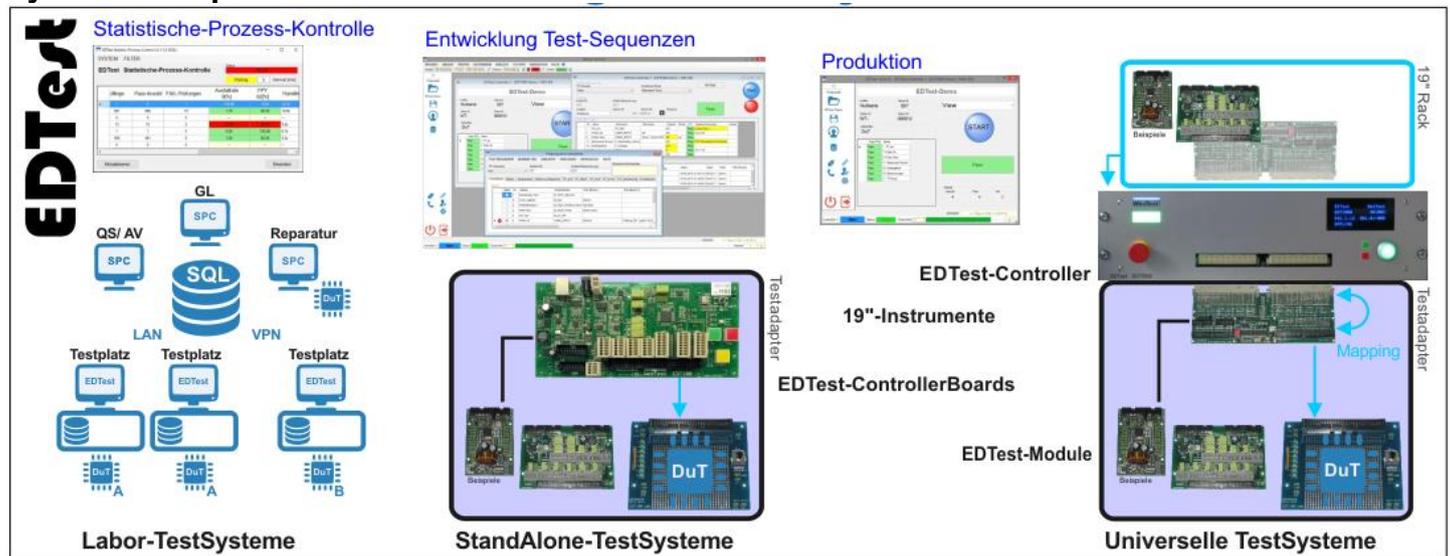
### Universeller Testsequenzer für Bauelemente, Baugruppen & Geräte

Mit dem EDTest-Testsequenzer können Abläufe zur Funktionsprüfung automatisiert werden. Dabei werden Anforderungen, wie Testprotokollierung, Dokumentation, Datenbank, Statistik, Seriennummernverwaltung, Prüferverwaltung, Instrumentenverwaltung, usw. unterstützt.

EDTest vereint das Programmierwerkzeug zur Beschreibung der Testschritte, als auch die Testoberfläche zur Testdurchführung. Die Bedienungsoberfläche kann an die Anforderungen z.B. für Programmierplatz, Reparaturplatz oder Serienprüfplatz angepasst werden.

Zur Stimulation und Messung können beliebige Instrumente mit USB, RS232, GPIB, usw. genutzt werden. Darüber hinaus stehen zahlreiche, speziell für die Funktionsprüfung optimierte EDTest-Module zum Aufbau von Testsystemen zur Verfügung. Die Programmierung der Testschritte erfolgt mit einer einfachen Kommando-Skriptsprache oder ergänzend mit VisualStudio C#-Funktionen.

### Systemkonzept



Ein Testsystem besteht aus der EDTest-Sequencer-Software auf einem Test-PC zur Test-Ablaufsteuerung, einem EDTest-Controller, evtl. weiteren Instrumenten (EDTest-Module oder Fremdhersteller-Komponenten) und einer Prüflings-Kontaktierung (Testadapter). Die EDTest-Instrumenten-Verwaltung ist Entwicklungsbasis für die Instrumentenauswahl inkl. Datenblatthandling und dem Instrumentencheck während der Testausführung.

In der Wartungsverwaltung werden Kalibrierungsintervalle von Instrumenten, Wechselintervalle von Prüfnadeln, Reinigungsintervalle usw. eingetragen und von EDTest kontrolliert, sowie Mails zur Durchführung versandt.

Mit den EDTest-Controllern kann eine externe Testablaufsteuerung (Start, Not-Aus, Anzeige Test-Run/ PASS/ FAIL) erfolgen. Das Not-Aus-Konzept erlaubt eine leitungsgebundene und softwaregesteuerte NOT-Aus-Betätigung mit gegenseitiger Auslösung.

EDTest ermöglicht die Testablaufsteuerung und die Erstellung bzw. Programmierung/ Debugging der Testprogramme. Die Testschritte können als Kommando-Script mit einfachen Kommandos der EDTest-Controller (z.B. „PS #2 5V 100mA“ setzt die zweite PowerSupply) oder mit frei programmierbaren Funktionen in einer Microsoft-VisualStudio-.NET-Programmiersprache erstellt werden. EDTest erzeugt ein C#-Quellcode-Gerüst, so dass nur die Testschritte ausprogrammiert werden müssen.

Das Testprogramm kann neben den Testschritten in einem Datencontainer auch Prüflings-Daten wie beispielsweise eine MikroController-Firmware, DLLs oder in den Testschritten aufgerufene EXE-Dateien enthalten. Das gesamte EDTest-Testprogramm besteht aus nur einer Datei, so dass der Fertigungsprozess schnell und risikofrei aktualisiert werden kann.

Zur Entwicklung eines vollständigen Funktionstestsystems sind Fremdhersteller-Editoren für Schaltpläne und Etiketten in EDTest integriert.

Zur Nutzung in Fertigungsbetrieben verfügt EDTest über einen Testmodus mit prüferorientierter Bedienungsführung und Prüferverwaltung inkl. Rechtfreigaben. Es steht eine flexible Seriennummernverwaltung und ReTest aus der Datenbank zur Reparaturhandhabung bereit.

Aus dem Testprogramm kann eine vollständige Testprogramm-PDF-Dokumentation erstellt werden. Zur dokumentenechten Protokollierung der Testabläufe erzeugt EDTest PDF- Prüf- und Historie-Listen. Zur Ermittlung von Prozesskennzahlen zur Eignung von Testsystemen, steht ein Tool zur MSA – MessSystemAnalyse bereit, das den Prüfer durch den Prozess führt und ein MSA-Protokoll erstellt.



## Produktion

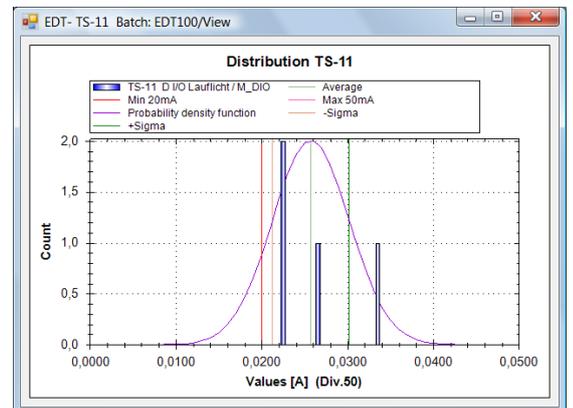
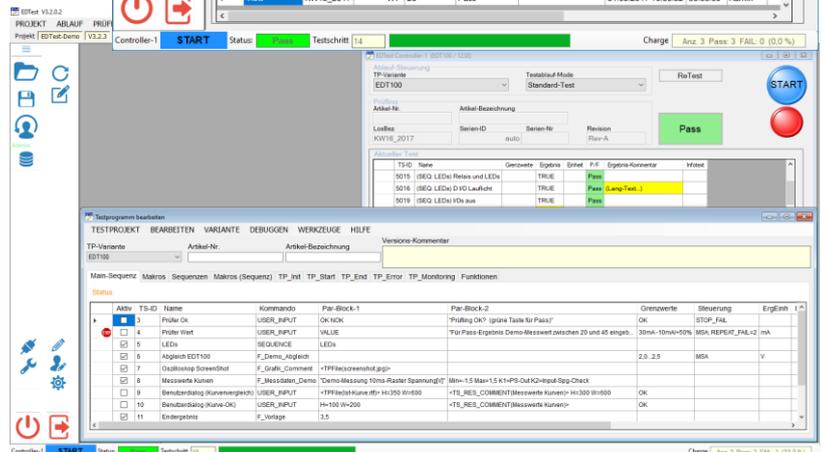
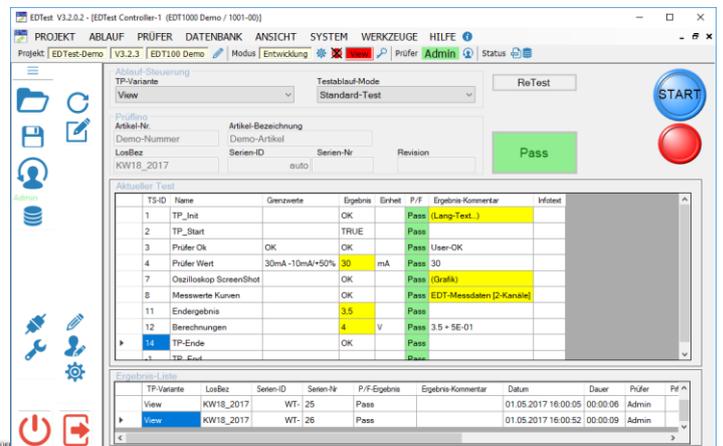
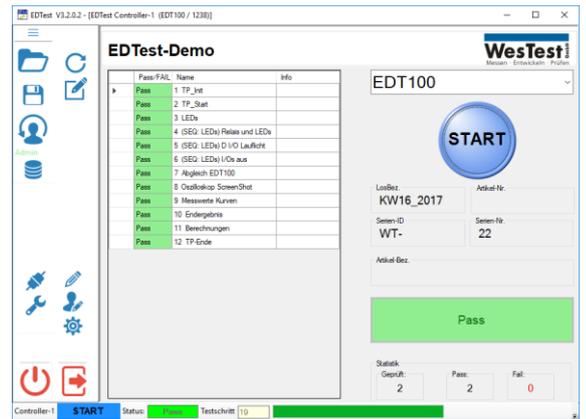
- **Projektoptimierte Bedienungsoberfläche** für TouchScreen-Bedienung (frei gestaltbar)
- **Standard-Bedienungsoberfläche** für schnelle Entwicklungs- und Reparaturprozesse
- **Direkt-Steuerungs-Oberfläche** für Analyseprozesse
- **Reparatur-Management** ReTest aus Datenbank, FAIL-Etiketten und FAIL-Protokolle am Reparaturarbeitsplatz
- **Retouren-Prüfung**
- **Prüferdatenbank** für Nutzungsberechtigungen
- **Seriennummern-Verwaltung/ Datenbank**
- **Einzeltest und Dauertest/ Datenbank** für Serienfertigung, Programmierung, Design-Verifizierung und Validierungs-Prüfungen
- **Test-Protokollierung** Datenbank und PDF-Protokolle
- **Testprogramm-Dokumentation** „1-Klick“
- **Test-Parallelbetrieb**
- **Mehrplatz/ Netzwerkfähig**, mehrere Prüfplätze für ein Prüflings-Fertigungslos

## Entwicklung

- **Testschritt-Tabelle** für Kommando-Skriptsprache und Grenzwerte
- **Funktionsübersicht** (C#-Programmiersprache) mit VisualStudio
- **Instrumenten-Verwaltung** Datenblätter/ Blockschaltbild v. Instrumenten
- **Wartungs-Verwaltung** Überwachung und Meldung von Wartungsintervallen
- **Mapping-Schaltplan** mit eCAD-Tool
- **Terminal** zur Ausführung von Einzelkommandos und Teilsequenzen
- **Debugging** (Script/ C#/ Terminal)

## Statistik

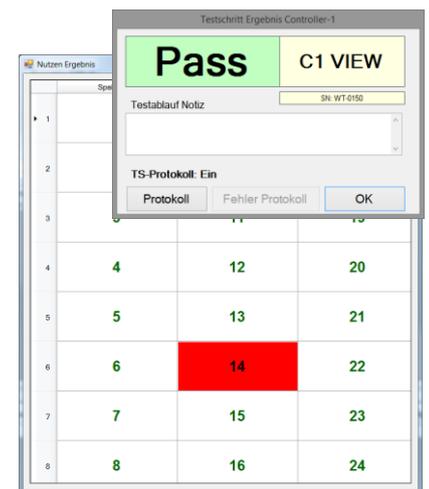
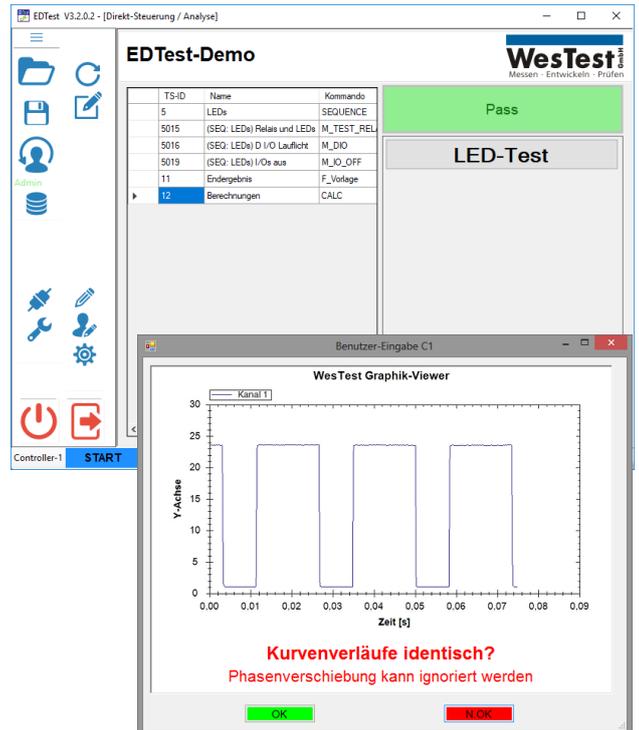
- **Statistische Prozesskontrolle** Ausfallquote und FPY mit Admin-Meldungen per eMail
- **Statistik-Protokolle** Ausfallstatistik, Fehlerverteilung, Testdauer/ Handlingsdauer, Numerische Grenzwert-Statistik
- **TREND-Analyse** über Testzyklen oder Zeit
- **MSA-MessSystemAnalyse:** Tool mit Benutzerführung, Prozesskennzahlen





## Ablaufsteuerung

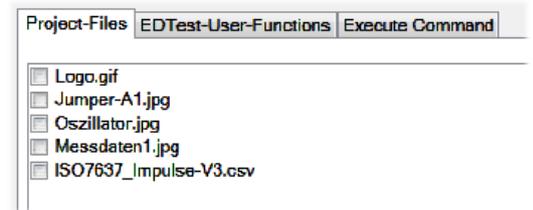
- Fertigungsoptimierte Bedienung mit Test- und Entwicklungsmodus
- Bedienungsoberfläche frei mit VisualStudio programmierbar
- Direkt-Steuerungs-Oberfläche Varianten mit Analyse-Attribut können unabhängig von Serientestanforderungen (Serialisierung, Datenbank, Protokoll) ausgeführt werden.
- Hardware-Steuerung (START/ STOP/ Not-Aus) und Betriebsanzeige (PASS/ FAIL/ Run) mit EDTest-Controller
- Systemcheck bei Test-Start (Test der Instrumente/ Schnittstellen, Prüfung Kalibrierungen, usw.)
- Testmodus mit Prüferidentifikation
- Einzeltest (Fertigungstest)
- Dauertest, Anzahl Abläufe einstellbar (RunIn/ BurnIn/ Design-Verifizierung, Validierung)
- ReTest zur Testwiederholung (mit unveränderter SN)
- ReTest von FAIL-Testschritten (Reparatur-Funktion)
- ReTests manuell und aus Datenbank
- ReTests aus Vorgängerlosen mit dem zugehörigen Testprogramm
- Einzelschritt/ Debug-Modus (Script- und Funktionsprogrammierung)
- Mehrplatz bzw. netzwerkfähig (Parallelzugriff auf Test-Datenbank, Verteilung von Seriennummernkontingenten, usw.)
- Parallelbetrieb: Ablauf eines Testprogramms auf bis zu 4 EDTest-Controllern
- Sprache: Deutsch, Englisch (im Testbetrieb umschaltbar)
- Fehler-Lokalisierungsinformationen (Fail-Infotexte)
- Serientest mit Seriennummernverwaltung
  - Manuelle SN-Eingabe/ Scan von SN-Aufklebern
  - Automatische Übernahme der SN aus Prüfling
  - Automatischer Scan der SN mit Testadapter
  - SN-Generierung mit SN-Druckfunktion (z.B. auf Etikett) aus SN-Kontingenten oder SN-Datenbank
- Manuelle Variantensteuerung (aus Liste oder Schaltknopf)
- Automatische Variantensteuerung (z.B. alle x-mal Variante A, sonst B, GoldenSample-Test, usw.)
- Ergebnis-Darstellung (Gesamtergebnis und Nutzen-Ergebnisse)





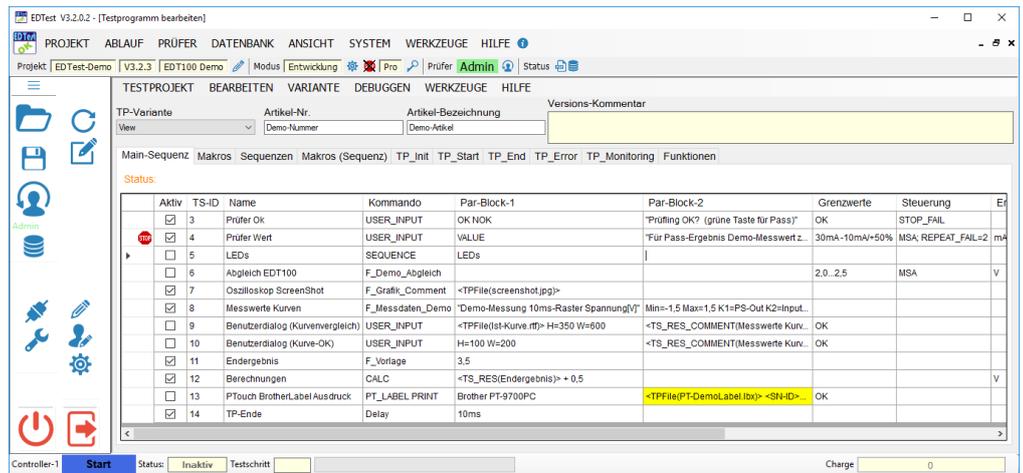
## Testprogramme

- Testprogramm kompakt in einer Datei: Einstellungen, Testschritte (Testablauf, Makros, Sequenzen), programmierte Testschritte/Befehle (DLLs), frei programmierte Testoberfläche, Prüfling-/ Test-Firmware, sowie prüflingspezifische Daten
- Testprogramme verschlüsselt (z.B. für Lizenzfertigung)
- Automatisches Testprogramm-Versionsmanagement mit Backup aller Vorgängerversionen.
- Instrumenten-Treiber und Bibliotheken frei programmierbar (mit identischem Programmiermodell, entsprechend Testschritt-Programmierung)

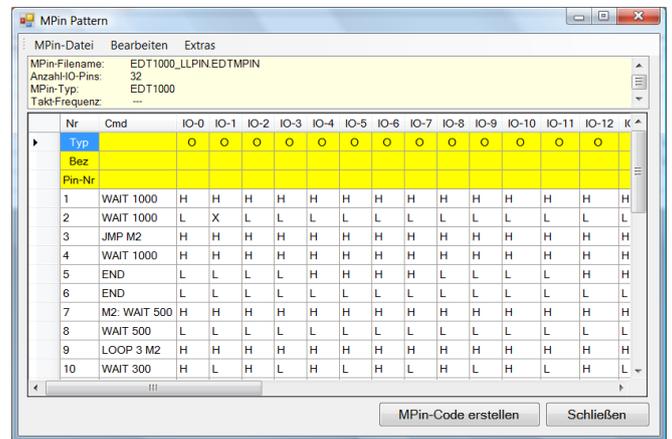


## Testschritt-Programmierung

- Kommando-Skriptsprache
- Kommandos mit Bibliotheken erweiterbar
- Makro-Sequenzen
- Import/ Export von Makros
- Dynamisch ladbare Testsequenzen (LOAD\_SEQUENCE)
- Frei programmierbare Testschritt-Funktionen (Microsoft Visual Studio .NET)
- Logiktabellen

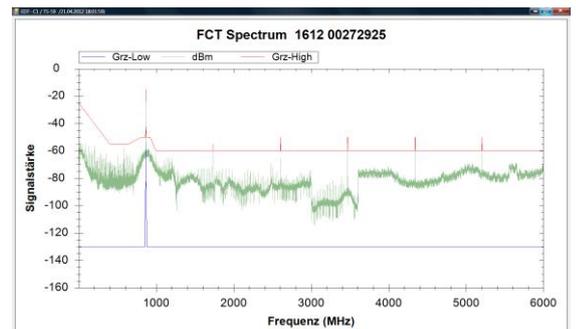


- Codesequenzen/ Makros für Testprogramm-Init/ Start/ Ende/ Fehler (TP\_INIT/ TP\_START/ TP\_END/ TP\_ERROR)
- Codesequenzen bei Testprogramm-Laden/ Schliessen (TP\_OPEN/ TP\_CLOSE)
- Codesequenz für Monitoring von Test-Randbedingungen (TP\_MONITORING)



- Kommandos
  - Kommunikation zu Instrumenten RS232/ USB S\_COM,...
  - Standard-Dialoge: USER\_INPUT (für Text/RTF-Dateien/ Grafik/ mehrkanalige Messwertdaten)
  - EXECUTE, DELAY, BREAKPOINT
  - IF/ ELSE/ GOTO, usw.
- Testschritt-Steuerung REPEAT, STOP\_FAIL, REPEAT\_FAIL/ ...

- Varianten eines Testprogrammes (beliebige Auswahl von Testschritten)
- Varianten-Attribute: Admin/ Direkt-Steuerung/ Init/ Start/ End
- Automatische Testprogramm PDF-Dokumentation
- Messwert-Darstellung als Text (numerisch/ ASCII), Grafik und mehrkanaligen Kurvenverläufen
- Ergebnis-Auswertung
  - Min/ Max Messwert mit automatischer Einheitenumrechnung, Prozentangabe
  - Logik (TRUE/ FAIL/ OK / NOK, ...)
  - String (numerisch, alphanumerisch, ASCII-Tabelle, ...)
  - Hüllkurve
- Dynamische Grenzwerte mit Grenzwertvariablen





## Debugging

- Debugging Kommando-Skript (Haltepunkte mit Zugriff auf Variablen u. Terminal)
- Debugging Testschritt-Funktionen in VisualStudio

## Instrumentenverwaltung

- Instrumenten-Check bei START (HW/SW-Version, Wartung/ Kalibrierung, usw.)
- Systemmeldungen per eMail an Administratoren
- Zugriff der Testprogramme auf Instrumente über Alias-Namen (z.B. RS232 erfordert keine COM-Port Zuordnung)
- Zugriff auf Online-Datenblätter der Instrumente
- Zugriff auf Schaltplan-Modul
- Terminal zum direkten Zugriff auf Instrumente parallel zur Entwicklung der Testabläufe
- EDTest-Controller-Boards EDT100BRD/500SET zur Integration in Testadapter (StandAlone Testsysteme)
- EDTest-Controller EDT100/ EDT500/ EDT1000 (19"-Rack Universal-Testsysteme)
- EDTest-Sub-Controller
- EDTest-Module (PowerSupplies, Multiplexer, ...)
- USB-/COM-Port Komponenten (z.B. Chip-Programmer, Bridge zur Prüfling-Testfirmware, usw.)
- GPIB-Instrumente
- Ethernet-IP-Instrumente
- Instrumente mit Netzwerkpfad (z.B. USB-Stick, Server, usw.)
- Drucker (Windows-Drucker-Warteschlange)
- Instrumente-Treiber als VisualStudio-DLL (VI)

## Wartungsüberwachung Instrumente

- Meldung von Wartungsbedarf abhängig von: Testzyklen, Datum, Dauer
- Wartungs-Vorwarnung

## Werkzeuge

- Etikettendesign grafisch (Bartender oder PTouch-Editor)
- Verdrahtungsplan/ Mapping mit Schaltplan-Editor (Eagle)

## Datenbanken

- Datenbank für Testdurchläufe und Messergebnisse
- Keine Datenbankinstallation erforderlich (MS Access), optional Anbindung an andere Datenbanken: SQL, SAP, ...
- Separate Test-Datenbanken für Einzeltests und Dauertests
- Gespiegelte Export-Datenbank (z.B. für Zugriff durch SPC-Tools (Statistical Prozess Control))
- Prüferdatenbank mit Freigebe für bestimmte Testprogramme und diverse Berechtigungen, zeitlich limitierte Freigaben durch QS-Verantwortliche, Passwortverwaltung, usw.
- Instrumentenliste mit Kalibrierungsdatum, Zertifikat, usw.
- Backup-Datenbank, automatisches Backup

