

COM-Schnittstellen – Schlüssel zur Messhardware

Individuell programmierte Messablaufsteuerungen stellen heute immer höhere Forderungen in Puncto messtechnischer Funktionen, Echtzeitfähigkeit, Vernetzbarkeit, Datenmanagement und intuitiver Bedienung. Die zum Einsatz kommende Hardware wird stets vielkanaliger und intelligenter und längst sind einfache PC-Einsteckkarten für viele professionelle Aufgaben überfordert. Außerdem stellt die Messaufgabe und Ablaufsteuerung schon lange nicht mehr die alleinigen Programmfunktionen.



Bild 1: μ -Musycs als universale Messhardware und Entwicklungsplattform für Prüfsandaufgaben

Üblicherweise besteht ein professioneller Prüfstand heute aus einem autark arbeitenden Automatisierungssystem das den „Prüfstand“ bewegt, dem Messsystem, einem zentralen PC der die Bedienoberfläche vorgibt, dem Datenmanagement und der Vernetzung der Komponenten untereinander bis hin zu einer zentralen Datenbank im LAN des Anwenders. Da alle Prüfstandskomponenten miteinander integriert funktionieren, darf die Wahl der Programmiersprache nicht von einer der Komponenten bestimmt sein. Vielmehr steht die Effizienz der Programmierung und Systemintegration im Vordergrund.

Die Programmierung von Messgeräten, mit dem Ziel Messablaufsteuerungen zu automatisieren, ist weit verbreitet. Ende der 70er Jahre begann der Siegeszug des IEC-Bus der an keinem Messgerät fehlen durfte. Die Programmierung erfolgte damals in Basic auf Commodore-Rechnern und CPM. Ein mühseliges Unterfangen das Ende der 80er Jahre durch die Programmierung auf dem PC und von MS-Windows Programmiersprachen abgelöst wurde. Mit dem Siegeszug der PC-Messtechnik unter Windows stiegen aber auch gleichzeitig die Ansprüche. Der Anwender fordert eine intuitive, grafische Bedienoberfläche. Die Anzeige der Daten soll online erfolgen. Signalanalysen und Auswerteprotokolle sollen ebenfalls automatisch erstellt werden. Diese Forderungen sind gleich-

bedeutend mit einer wesentlichen Komplexitätserhöhung der Prüfprogramme. Der zeitliche Aufwand zur Erstellung solcher Programme ist heute enorm gewachsen. Pflege, Update und Erweiterung sind aufwendig und teuer. Die heute verwendeten Programmierwerkzeuge in messtechnischen Aufgaben sind häufig Programmgeneratoren wie z.B. LabVIEW von National Instruments oder auch die neuen gut handhabbaren Programmiersprachen wie Microsoft's Visual Basic oder Borland's Delphi.

Komplette Messhardware

Neben der verwendeten Programmiersprache liegt der Schlüssel zur effektiven Programmierung einer Messhardware neben den Gerätetreibern im Funktionsumfang der verwendeten Hardware. Individuell programmierte Prüfstände sind meist Unikate und weisen daher ein hohes Funktionsrisiko auf. Dieses lässt sich minimieren indem möglichst viele, möglichs wiederverwendbare Komponenten (HW und auch SW) eines Herstellers in einem System vereint sind, weil man davon ausgehen kann, dass diese auch optimal zusammenarbeiten. In Bezug auf das Messsystem bedeutet dies, dass die komplette Messaufgabe

mit einem System gelöst werden sollte. Das heisst, dass folgende Komponenten integraler Bestandteil des Messsystems sind:

- ▷ Signalkonditionierung (Vorverstärker, Filter, Potenzialtrennung),
- ▷ Digitalisierung (AD-Wandlung),
- ▷ Echtzeitfähigkeit (Verrechnen von Kanälen, Grenzwertüberwachung etc.),
- ▷ lokale Speicherung,
- ▷ Vernetzung (Ethernet oder industrielle Feldbusse).

In der Praxis wird man sich einer Automatisierungsaufgabe optimalerweise so nähern, dass man zunächst die Messung manuell vornimmt. Problematisch ist dies, wenn die Messhardware keine eigene Bedienoberfläche besitzt, mit der man manuell und im Sinne eines Labormessgerätes arbeiten kann. Es ist daher zu fordern, dass das Messsystem eine vollständige Bediensoftware standardmäßig mitbringt.

Das System μ -Musycs von IMC, Berlin (Vertrieb: Additive) ist ein solches Messsystem.

Mit seiner standardmäßigen Bedien- und Anzeigesoftware ist es sowohl im Laborbetrieb als auch innerhalb von Prüf- und Automatisierungsaufgaben einsetzbar. In der Vorbereitung zur Prüfstandsprogrammierung lassen sich alle



all-electronics.de
ENTWICKLUNG. FERTIGUNG. AUTOMATISIERUNG



Entdecken Sie weitere interessante Artikel und News zum Thema auf all-electronics.de!

Hier klicken & informieren!



Geräteeinstellungen, Messkurvendarstellungen, Messdokumentationen etc. manuell voreinstellen und testen. Dies spart enorm Zeit bei der Prüfprogrammerstellung da spezifische Messungen zunächst manuell erprobt werden können und eine Veränderung der Einstellung sehr schnell geht. Solche vordefi-

nenleistung des Prozessorarrays in μ -Musycs begrenzt. Es stehen eine Vielzahl mathematischer Grundfunktionen inkl. Trigonometrischer und Statistikfunktionen sowie Steuerkonstrukte zur Verfügung. Spezielle und normierte Analyseverfahren wie FFT, Korrelation, Klassierung, Ordnungsanalyse u.v.m. sind genau-

Jahrzehnts entwickelte Programmierschnittstelle. Der Begriff COM steht dabei für eine Gruppe von Softwareerweiterungen zu den auch OLE, ODCX oder das durch Internetanwendungen bekannte ActiveX gehören. Bei der Definition von COM wurden folgende Ziele gesetzt:

1. nahezu beliebige Erweiterbarkeit von Software durch die Verwendung einer standardisierten Schnittstellen,
2. Verwendung von Modulen (Komponenten) die abgeschlossen getestet sind und stabil funktionieren und
3. einfache Verwendbarkeit in beliebigen Programmierumgebungen.

Alle diese Anforderungen erfüllen, was eingangs für die Software bzw. Treiberkomponenten in Prüfstands- und Automationsanwendungen gefordert wurde. Somit liegt es auf der Hand: Die zeitgemäße Programmierschnittstelle muss in die Gruppe der COMs eingeordnet sein.

Die beste Entwicklungsumgebung

Fragt man zehn Entwickler nach der besten Entwicklungsumgebung, so erhält man wahrscheinlich zehn verschiedene Antworten. Warum? Die gängigen Programmiersprachen unterscheiden sich in ihren Möglichkeiten nur unwesentlich, sie alle haben

ihre Stärken und Schwächen. Daraus ergibt sich, dass es nicht *die* beste Entwicklungsumgebung für die Erstellung einer Systemlösung gibt. In den meisten Fällen ist das Werkzeug am besten, mit dem der Benutzer am meisten vertraut ist. Das erhöht die Effizienz. Er muss weder die Zeit investieren, um sich mühsam einzuarbeiten, noch die Stärken und Schwächen der neuen Software erkunden.

Die Realität heute ist jedoch, dass der Markt vielfältigste Messhardware anbietet. Diese ist oft nur mit speziellen Entwicklungsumgebungen zu programmieren. Von zeitgemäß kann hier nicht die Rede sein. Im Gegenteil - der Anwender möchte und muss ohne großen Zeitaufwand sein technisches Problem lösen, wobei es nicht ökonomisch sein kann, sich jedes mal mit einer neuen Entwicklungsumgebung zu befassen.

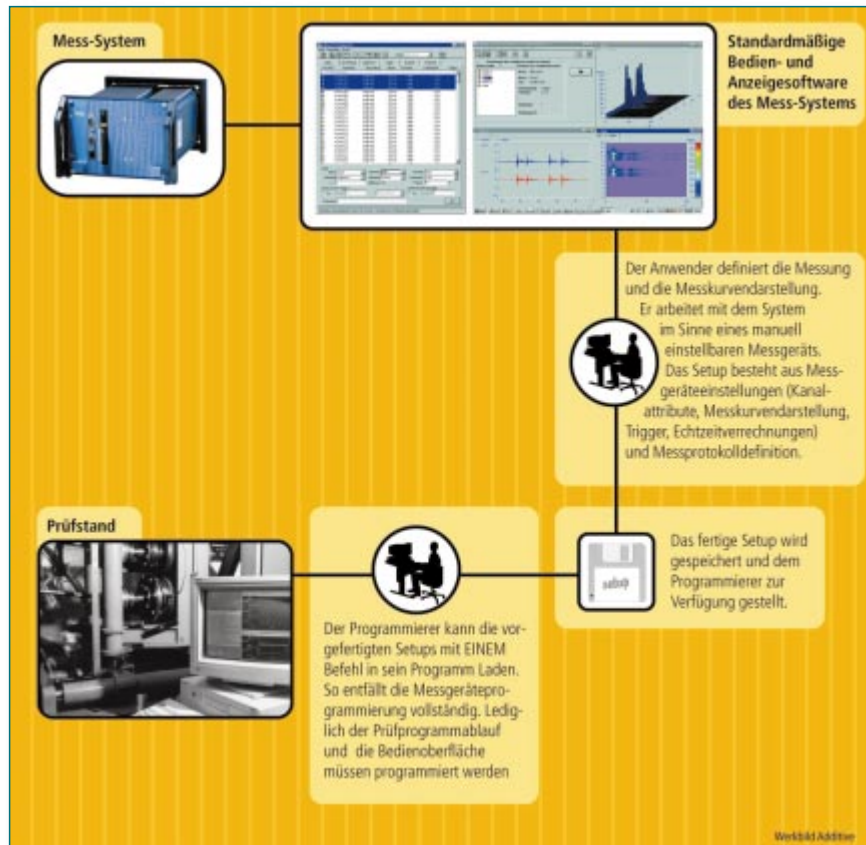


Bild 2: Schematischer Arbeitsablauf

nierten Setups lassen sich speichern und können mit einem Befehl ins Prüfprogramm integriert werden. Eine weitere Forderung ist, dass moderne Messgeräte nicht nur registrieren, sondern auch eine Datenreduktion und/oder Analyse in Echtzeit vornehmen. Durch eine Triggermaschine mit 48 unabhängigen Triggern kann jeder Kanal separat gestartet und gestoppt werden. So lassen sich unnötige Datenmengen vermeiden, da man nur dann misst, wenn wirklich etwas passiert.

μ -Musycs ist darüber hinaus ein Personal Analyser, wo sich mit der Funktion *Online-Famos* beliebige Echtzeitfunktionen realisieren lassen. Der Entwickler benötigt dabei keinerlei Kenntnisse über die Programmierung von DSPs, da die Rechen- und Steueranweisungen klar-schriftlich in einer Formelzeile eingegeben bzw. über einen Assistenten aufgerufen werden. Klammerebenen und auch Echtzeitprogramme über mehrere Zeilen werden unterstützt. Die Leistungsfähigkeit ist nur durch die Re-

so wie beliebige digitale Filter auf Knopfdruck aufrufbar. Das Echtzeitprogramm des Personal Analysers wird wie alle anderen Messgerätefunktionen im Setup abgespeichert.

Durch den Personal Analyser wird das μ -Musycs zu einem intelligenten Messgerät, das mit seinen Aufgaben wächst, ist in seinem Funktionsumfang praktisch beliebig erweiterbar. Heute ein Störwerterschreiber und morgen ein Onlineklassierer. Der Anwender bestimmt die Funktionen.

Der Treiber macht's

Die Arbeitszeit ist heute der am stärksten limitierte und kostenintensivste Faktor bei der Systemintegration. μ -Musycs stellt in Verbindung mit seiner COM-Programmierschnittstelle ein leicht zu benutzendes Werkzeug für solche Aufgaben dar. COM bedeutet übersetzt *Component Object Modelling* und ist eine von Microsoft bereits Anfang des letzten

Was ist anders bei IMC-COM?

Die IMC-COM-Programmierschnittstelle weist in Verbindung mit der Standard Bedien- und Anzeigesoftware und der hohen Leistungsfähigkeit der Hardware und Firmware einen anderen Weg. IMC-COMs lassen sich in alle modernen Programmiersprachen wie z.B. *Delphi*, *Visual Basic* oder *Visual C++* einbinden und ermöglichen so ein einfaches Verwalten, Parametrieren und Visualisieren von Messdaten. Der Kunde arbeitet in seiner favorisierten Entwicklungsumgebung und kann sich voll und ganz der technischen Aufgabenstellung widmen.

Nun stellt sich die Frage: Was ist anders bei den IMC-COMs? Üblicherweise erfolgt die Programmierung einer Messhardware Schritt für Schritt; Zeile für Zeile. So werden die Eigenschaften aller Kanäle mühsam über Codezeilen programmiert. Schließlich gilt es für jeden Kanal Abtastrate, Skalierung in physikalische Einheit, Messdauer, Triggerbedingungen, Echtzeitverrechnungsfunktionen, usw. zu definieren. Dann sind noch viele Zeilen Code nötig, um die Messkurvendarstellung zu definieren, die Kommunikation mit dem Messgerät herzustellen und die Daten zur Speicherung wieder richtig in Dateien zu verpacken. Bevor diese Schritte nicht erledigt sind, kann der Entwickler sein Programm meist noch nicht einmal testen.

Wird dann der erste Test durchgeführt, gibt es oft „lange Gesichter“, weil es wie immer nicht auf Anhieb funktioniert.

In diesem Szenario ist es schwierig die fehlerhafte(n) Zeile(n) zu finden. Der Fehler kann an mehreren Stellen versteckt sein. Falsche Kanalattribute führen dazu, dass Kanäle falsch oder gar nicht gemessen werden. Auch im Bereich der Messkurvendarstellung lassen sich beliebige Fehler machen.

Mit Hilfe der standardmäßigen Bedien- und Anzeigesoftware lässt sich die „Programmierung“ des Messgeräts wesentlich vereinfachen. Mit ihr arbeitet der Entwickler im Sinne eines manuell einstellbaren Labormessgerätes. Er definiert alle für die Messung entscheidenden Parameter wie Kanalattribute, Messkurvendarstellungen, Triggerbedingungen, Echtzeitverrechnungsfunktionen und die anschließende Dokumentation der Messdaten. Die Ergebnisse können am Bildschirm direkt kontrolliert werden. Sind alle Messgeräteeinstellungen, Kurvendarstellungen und Dokumentationen vorgenommen und getestet, werden sie als Konfiguration gespeichert. Im zu programmierenden Messprogramm sind diese Konfigurationsdateien dann über die Programmierschnittstelle einfach aufzurufen. Das

mühselige Zeile für Zeile programmieren der Einstellungen entfällt vollständig.

Programmschwerpunkte eines Prüfprogramms

1. Einstellung des Messgeräts,
2. Trigger: Start-/Stopbedingungen,
3. Echtzeitfunktion,
4. Darstellung der Messdaten,
5. Speicherung der Messdaten (lokal, extern oder im Netzwerk),
6. Prüfablauf,
7. Kommunikation mit anderen Prüfstandskomponenten (z.B. Automatisierungssystem, weitere Messgeräte),
8. Bedienerschnittstelle (User interface),
9. Dokumentation des Prüfablaufs und der Messdaten.

Hier entlastet die IMC-COM-Programmierschnittstelle den Entwickler indem wesentlich Punkte (1, 2, 3, 4, 5 und 9) der Prüfstandssoftware *nicht* explizit programmiert werden müssen. Sie werden mit der Bedien- und Anzeigesoftware des Messgeräts eingestellt, getestet und als Setup abgespeichert. Im Code wird das Setup dann über einen Befehl der IMC-COM-Programmierschnittstelle geladen.

Daraus ergeben sich für den Entwickler große Vorteile:

- ▷ Zeitersparnis beim Erstellen des gesamten Programms,
- ▷ Zeitersparnis beim Debuggen des Programms,
- ▷ Vermeidung von Fehlern,
- ▷ das Programm ist kürzer, einfacher und überschaubarer,
- ▷ Wartung, Modifikation, Weiterentwicklung etc. sind vergleichsweise einfach und damit preiswert.

Die imc-Philosophie

- ▷ Erstellen aller Messeinstellungen in der Bedien- und Anzeigesoftware,
- ▷ das Ergebnis kann direkt überprüft werden,
- ▷ Abspeichern der Messeinstellungen als Setup,
- ▷ Laden des Setups über das COM in der Programmiersprache,
- ▷ Erstellen und verändern der Prüfablaufs in der Programmiersprache.

