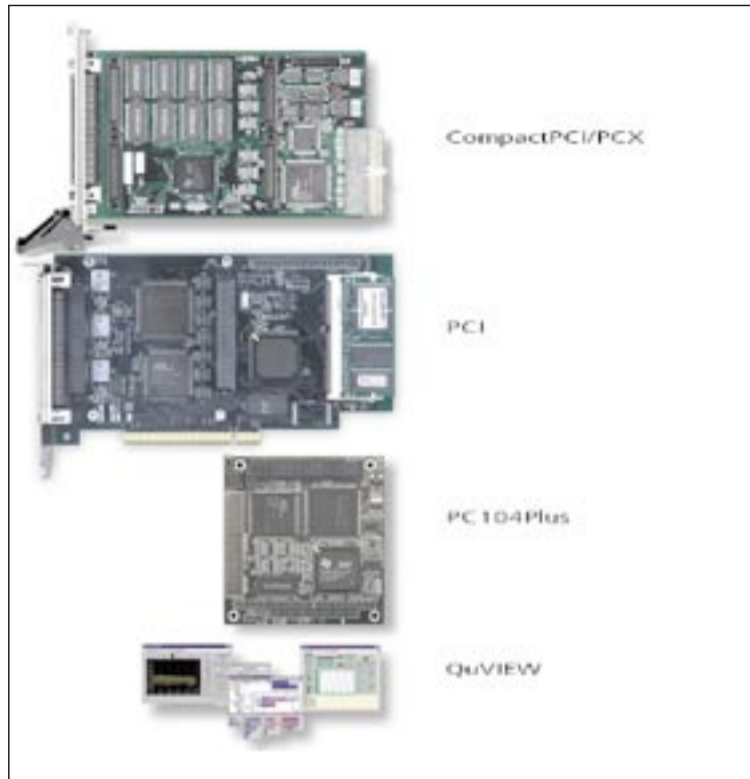


# Programmera DSP för real-tid

Att skapa virtuella instrument som bygger på PC är kostnadseffektivt och ger breda möjligheter. Men vanliga operativsystem begränsar resurserna. Här beskriver Mikael Lindner, MyTech, hur man med specialprogramvaran QuX tillsammans med mätprogramvara som LabVIEW eller VisualBasic enkelt och effektivt kan programmera DSP med realtids-egenskaper.

För många test- och mätapplikationer kan en PC idag vara en kostnadseffektiv lösning. En PC har idag ett mycket bra förhållande mellan pris och beräkningskraft. Det finns även ett stort utbud av I/O-kort på marknaden och de stödjer enkla utvecklingsprogram som t ex LabVIEW från National Instrument. Men även med en multi-GHz Pentium dator begränsas arbetet den kan utföra för applikationen ifråga. Det beror delvis på resurskrävande operativsystem som t ex Microsoft Windows XP. Med ett sådant operativsystem och om applikationen har ett komplext användargränssnitt, samtidigt som mätdata ska samlas in och analyseras för att sedan rapporteras, då räcker inte datorkraften till. För kritiska realtidsapplikationer kan dessutom inte en PC uppfylla de krav som ställs.

En lösning på problemet är att köra realtidsfunktioner i programmet på en andra processor. Men även en kraftfull DSP är inte mycket till hjälp om den hela tiden måste avväntas och kommunicera med en PC för att kunna utföra nästa funktion. Ett



QuX (QuVIEW och QuBASE) och SI-DDK levereras tillsammans med varje DSP-kort från Sheldon Instruments.

alternativ till detta finns utvecklat av Sheldon Instruments. Denna lösning är en kombination av hård- och mjukvara. Alla funktioner för mätinsamling, kontroll och analys laddas på ett separat DSP-kort där realtidsfunktionerna kan exekveras utan att störas av PCn som kan göra andra saker samtidigt. Ännu enklare blir det om man som i det här fallet kan använda ett enkelt programmeringsverktyg som LabVIEW från National Instrument eller Visual Basic från Microsoft.

## ANVÄNDARVÄNLIG DSP-PROGRAMMERING

Produkten som kan erbjuda detta kallas för QuX och dess mål är enkelt: att ge såväl novisen som experten möjlighet att bygga de mest komplexa och sofistikerade applikationer utan att behöva lägga ut en förmögenhet på dyra realtids-OS som ofta förknippas med långa inlärningsperioder och

udda utvecklingsmiljöer.

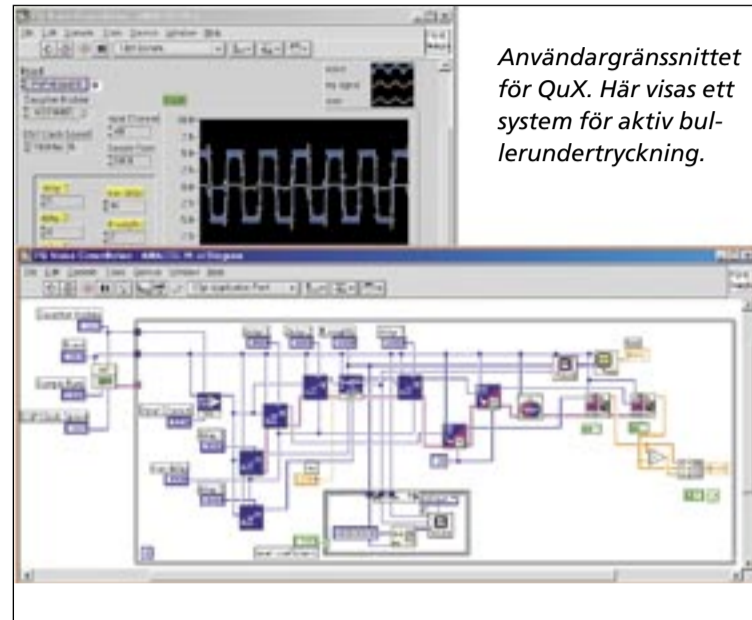
QuX har funnits länge och är välkänd i USA. I Europa är det marginellt känt och använt. Sedan introduktionen 1991 har QuX utvecklats från enkla funktioner för mestadels styr- och reglerapplikationer i realtid, till den kompletta realtidsplattform den är idag. QuX är idag ett flertrådig realtidsoperativ där man kan programmera funktioner för både mät punkt per mät punkt och som kompletta mätblock och exekvera dessa på ett DSP-kort med tillhörande multifunktions I/O-kort. De funktioner som används för att bygga ett program består av en stor mängd olika verktyg, t ex vektorer, matriser, filter, signalhantering för tids- och frekvensdomän, Z-transformationer för att implementera polynom direkt i z-domänen. Alla dessa verktyg tillsammans är nog för att kunna uppfylla de mest komplexa krav för en avancerad mät- och kontrollapplikation.

## IDEALISK ARBETSFÖRDELNING

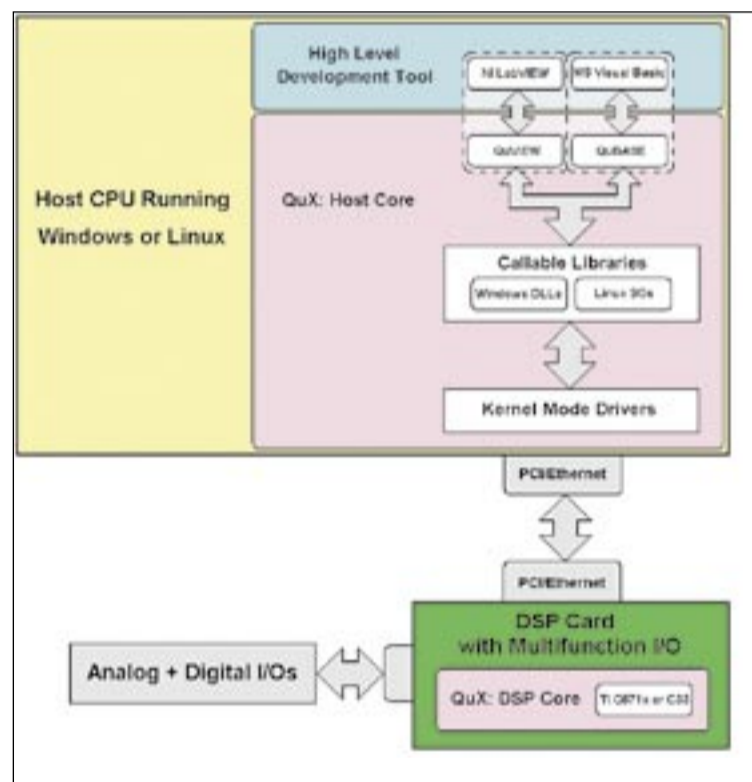
Kärnan i QuX består av två huvudkomponenter: en del för PCn (värden) och en del för DSPn. Interaktionen mellan dessa är synkroniserad med ett dynamiskt schema och en minnes-pool.

Värdkomponenten består av:

- Lågnivå-drivrutiner för att kommunicera med DSP-hårdvaran.
- Ett flertal DLLer för Windows eller gemensamma objekt (SO) för Linux som fungerar som en brygga mellan lågnivå-drivrutinerna och användargränssnittet.
- Ett användargränssnitt/utvecklingsverktyg som fungerar som ett skal för de DLLer som behövs för att kommunicera med DSP-kortet.



Användargränssnittet för QuX. Här visas ett system för aktiv bul-lerundertryckning.



Uppbyggnaden av ett mätsystem.

QuX erbjuder två val för att bygga en applikation med användargränssnitt på en PC: LabVIEW från National Instrument (grafiskt utvecklingsverktyg) och Visual Basic från Microsoft för programmering med traditionell text-syntax. Programvaran för LabVIEW kallas QuVIEW och för Visual Basic kallas den QuBASE. QuX kan köras på Win9x/NT/2000/XP eller Linux.

DSP-komponenten består av ett DSP-program (en COFF-fil) som består av realtidsoperativet, funktionsbiblioteken som applikationen baseras på och alla funktioner som behövs för att kommunicera med värden (PC).

## FÅ FÖRKUNSKAPER KRÄVS

Ingen erfarenhet av C-kompilatorer, assembler-språk eller DSP behövs för att använda QuX. Sheldon Instruments har utvecklat det till ett nyckelfärdigt mät- och kontrollsystem. Det behövs ingen C-kompilator, ingen assembler eller länkare och det behövs heller ingen kunskap om hur en DSP

fungerar. QuX är extremt snabbt och effektivt. Beroende på applikation så kan t ex ett LabVIEW-program accelereras upp till 100 gånger, ännu mer om flera DSP-kort används i samma program.

## PUNKTER, VEKTORER ELLER STRUKTURER

Det finns två distinkta metoder för databehandling tillsammans med programflöde och strukturering implementerade i DSPn, därför är funktionsbiblioteken uppdelade på ett sätt som åskådliggör detta på ett så bra sätt som möjligt.

Den första metoden, som kategoriseras som "Point Queue", är lämpad för kontinuerlig datahantering i realtid. Data hanteras på basis av "sample per sample", dvs ett mätvärde returneras så fort det är inläst från t ex en A/D och kan behandlas innan nästa mätvärde returneras. En typisk applikation är rörelsekontroll.

Den andra metoden, som kategoriseras som "Vector Queue", är bättre lämpad för att databehand-

la stora block av mätvärden. En typisk applikation är dynamisk signalanalys.

Slutligen, programflödet kan kontrolleras med universella villkorliga strukturer som är implementerade med hjälp av traditionella "Case", "Conditional Execute/Branch" och "While Loop".

Nedan visas några typiska exempel på vad som kan göras med QuX i realtid:

- Hämta mätvärden från ett godtyckligt antal analoga och digitala ingångar
- Använda digitala filter, aritmetik och differentialräkning
- Processkontroll med PID-reglering eller "State Space Algorithms"
- Generera godtyckliga vågformer
- Kontinuerlig datalagring till hårddisk med hastigheter upp till 800kHz
- Samtidig visning av tid- och frekvensdata

Data som behöver överföras mellan värden (PC) och DSP, för grafisk presentation eller uppdatering av parametrar i realtid, överförs i bakgrunden med hjälp av DMA.

Tack vare att QuX körs på DSP-kortet och dess autonoma kärna, blir realtidsapplikationen helt fristående från värden (PC) och behöver således inte någon sådan för att fungera. DSP-korten kan bestyckas med flash-minne för att köras helt fristående.

**QUVIEW: TURBOLADDA LABVIEW**

För att kunna accelerera LabVIEW med dessa DSP-kort så har Sheldon Instruments skrivit om ca 75 procent av programmets (LabVIEW) matematik- och analysbibliotek samt adderat ytterligare funktioner. "Point Queue"- och "Vector Queue"-funktionerna ligger som separata ikon-grupper precis som de gör vid standardanvändandet i LabVIEW. Användaren kan använda och blanda dessa olika grupper helt godtyckligt, dvs utnyttja funktioner för att använda i PCn (standard LabVIEW-ikoner) eller i DSPn (QuVIEW-ikoner).

**QUBASE: REALTID I VISUAL BASIC**

För kostnadsmedvetna utvecklare kan Visual Basic vara tilltalande tack vare dess text-baserade paradigm och möjligheten att skapa mindre, exekverbara program. "Point Queue"- och "Vector Queue"-funktionerna är precis de samma som för QuVIEW och används som enkla funktionsanrop med unika prefix för varje funktionstyp. Användaren kan godtyckligt blanda funktionsanrop som ska exekveras lokalt på värden (PC) eller på DSP-kortet (QuBASE).

För utvecklare som behöver implementera sin applikation i

ett annat språk levererar Sheldon Instruments ett komplett paket med utvecklingskit och källkod som kallas SI-DDK. Här medföljer kod-exempel som visar hur t ex Code Composer från Texas Instrument används för att använda DSP-resurserna på bästa sätt. På värd-sidan (PC) medföljer stabila drivrutiner ("kernel mode drivers") för Windows och Linux. Med dessa drivrutiner medföljer även källkoden, ett komplett API-bibliotek och funktioner för att ladda COFF-filen till DSP-kortet. Allt detta medföljer varje

DSP-kort från Sheldon Instruments.

**HÅRDVARAN BAKOM MJUKVARAN**

QuX fungerar med alla DSP- och multifunktions I/O-kort från Sheldon Instruments. DSP-korten baseras på processorer från Texas Instrument. Processorerna är 32 bit flyttalsprocessorer från familjerna C33 och C671x som kan ge prestanda upp till 1800 MFLOPS vid 300 MHz DSP-klocka. Kortet finns i flera utföranden, t ex som PCI och PC104/

p. Inom kort kommer även kort med Ethernet som kommunikationsgränssnitt.

Multifunktions I/O-korten, som kallas SI-MODxx, sätts ihop med DSP-kortet. Några av alla de funktioner som erhålls med en enhet är:

- 4-64 analoga ingångar med förstärkning 1-1000
- Upp till 16 analoga utgångar
- 16 bit upplösning
- Samplingshastighet upp till 250 kHz, vid samtidig datainsamling, och upp till 1MHz additiv sampling.

- 36 digitala I/O
- Två PWM-utgångar
- Ingångar för att mäta puls/frekvens, kvadratur-enkodare (quadrature encoders)
- \* Flexibel klockhantering med 3 DDS (direct digital synthesizers) och 4 händelse-räknare (event counters).

För att utöka antalet analoga ingångar kan flera DSP-kort synkroniseras i samma applikation.

MIKAEL LINDNER, MYTECH