

Integration of Hardware with Simulation Platform



Emil Vestman

Dept. of Industrial Electrical Engineering and Automation
Lund University



Integration of Hardware in Simulation Platform

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

INTEGRATION OF HARDWARE WITH SIMULATION PLATFORM



Integration of Hardware in Simulation Platform

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

ABSTRACT

This thesis was conducted at CC-Systems in Uppsala, Sweden. CC-Systems develop and deliver control systems for demanding environments.

For development and testing of the software for embedded systems, CC-Systems provide simulator software that enables the ECU (Electronic Control Unit) code to execute on a PC rather than on the target hardware. In testing it is common to execute the controller software on a PC together with simulated machine models in order to be able to observe the behaviour of the controller.

In some later stages of the development the code is moved to the target hardware for further testing. In order to be able to use the same tools and simulated machine models as in former stages a signal interface between the PC and the target hardware is needed.

This thesis explores the possibility of using COTS (Commercial off-the-shelf) products instead of building custom electronics for generating and acquiring these signals. The product range is in this thesis limited to the C-series from National Instruments.

The result is a demonstrator for a solution based on CompactRIO hardware with communication to and from the CC-System simulation platform, CCSimTech. The suggested solution satisfies the request for a user friendly and extendible way of communicating.



Integration of Hardware in Simulation Platform

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

SAMMANFATTNING

Detta examensarbete har utförts hos CC-Systems i Uppsala. CC-Systems utvecklar och levererar styr- och kontrollsystem för krävande miljöer. I all utveckling av mjukvara för inbyggda system är testning en mycket viktig komponent. CC-Systems levererar därför även verktyg för att underlätta testning.

Testningen av mjukvaran för inbyggda system kan med dessa verktyg utföras i en simulerad miljö på Pc:n utan att behöva den faktiska hårdvaruenheten tillgänglig.

I vissa stadium av utvecklingen kan det dock bli aktuellt att flytta programvaran till enheten för fortsatta tester. Det är en stor fördel om denna flytt kan göras obemärkt för resten av testmiljön på Pc:n så att alla testverktyg från tidigare tester kan fortsätta att användas.

För att göra en sådan flytt möjlig måste de signaler som fanns in och ut från den simulerade hårdvaran nu ersättas med riktiga signaler, vilket innebär en så kallad brygning av signaler.

Detta arbete utreder hur man kan ersätta ett befintligt system för brygning som innehåller egenutvecklad signalkonditionering, med ett mindre system som använder sig av färdigbyggda utbytbara moduler istället. Hårdvaran kommer från National Instruments och utgör den så kallade C-serien.

Arbetet utreder den möjliga hårdvaran för att undersöka om hårdvaran kan leverera det som krävs för en brygning. Även den nya mjukvaran som tillkommer utreds och en arkitektur för ett system baserat på enheten compactRIO föreslås.

Resultatet av arbetet är en demonstrator som visar förslag på ett system där verklig hårdvara kommunicerar med CC-Systems simuleringsplattform, CCSimTech. Systemet uppfyller kraven som ställts på användarvänligt och utbyggbarhet.



Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

INNEHÅLL

Innehåll 4

 1.1 *Problemformulering*..... 9

 1.2 *Mål och syfte*..... 10

 1.3 *Avgränsning*..... 11

 1.4 *Disposition*..... 11

2 Studie av ett gränssnitt för blandad simulering..... 12

 2.1 *Tidigare system*..... 13

 2.2 *Kravinsamling*..... 14

3 Hårdvara..... 15

 3.1 *I/O-moduler i C-serien*..... 15

 3.2 *Compact DAQ (Data Aquisition)*..... 15

 3.3 *Compact RIO (Reconfigurable Input/Output)* 16

 3.3.1 *Realtidsdator* 16

 3.3.2 *FPGA* 16

 3.4 *Evaluering av hårdvara* 17

4 Mjukvara 19

 4.1 *CCSimTech* 19

 4.2 *C/C++*..... 19

 4.3 *LabVIEW*..... 19

5 Arkitektur för brygging av signaler..... 21

 5.1 *Arkitektur för FPGA* 21

 5.1.1 *Svårigheter med återanvändningsbar kod*..... 21

 5.1.2 *Val av arkitektur* 21

 5.2 *Arkitektur för kommunikation mellan RT och PC*..... 22

 5.2.1 *TCP/IP*..... 22

 5.2.2 *Shared Variables* 23

 5.3 *Val och testning av vald teknik* 23

 5.4 *Sammanfattning av mjukvaruarkitektur*..... 26

6 Diskussion och Slutsatser (Teknikstudie) 27

7 Tankskötaren Thomas..... 28



Integration of Hardware in Simulation Platform

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

8	Diskussion och Slutsatser kring Tankskötaren Thomas	30
9	Framtida arbete	30



Integration of Hardware in Simulation Platform

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

FÖRORD

Denna rapport utgör sista anhalt på vägen till min civilingenjörsexamen i maskinteknik.

Jag vill tacka CC-Systems i Uppsala för att ha givit mig möjligheten att genomföra mitt examensarbete hos dem. Det har varit 20 angenäma veckor och jag har blivit väl omhändertagen och lärt mig mycket.

Ett särskilt tack till Per Brolin som har varit min handledare hos CC-Systems och som sett till att arbetet alltid hållit rätt kurs, han har också varit en klippa på att få tekniska och programmeringsmässiga farthinder ur vägen.

Jag vill även passa på att tacka Gunnar Lindstedt på institutionen IEA på LTH för hjälp och uppmuntran, inte bara under examensarbetet utan även under tidigare projekt.

Ett sista personligt tack för tålamod och uppmuntran till min familj, mina vänner och min Emma, utan ert stöd hade världens ingenjörer fortfarande varit en man kort.
För ert stöd lovar jag er idel underverk.

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

Inledning

Cross Country Systems (CC-Systems) utvecklar och levererar styrsystem och produkter med specialisering mot tunga fordon i krävande miljöer, bland kunderna återfinns bl.a. John Deere, Bombardier Transportation, Atlas Copco, Bromma Conquip, Rolls-Royce Marine, BAE Systems Hägglunds och Volvo CE.

Företaget grundades 1991 men har haft en god tillväxt och har idag verksamhet i Alfta, Uppsala, Västerås, Örnköldsvik, Tammerfors och i Penang, Malaysia.

CC-Systems erbjuder ett antal färdiga hårdvaruenheter till försäljning men utvecklar också skräddarsydd mjukvara och hårdvara. Några exempel på de vanligaste hårdvaruprodukterna ur sortimentet kan ses i Figur 1-1.



Figur 2-1

CC Pilot XS som är en stöttålig PC med tryckkänslig skärm och ett flertal anslutningsmöjligheter. XS har en 533MHz Intel XScale-processor och upp till 256MB RAM. XS kan levereras med något av operativsystemen Windows CE 5.0, Linux eller RTLinux.

CrossFire MX1 och **FX1** är robusta kontrollenheter som är anpassade för flera olika typer av in- och ut signaler. Enheterna kan både användas fristående genom att programmeras med ett beteende och som I/O-moduler som bara skriver och läser signaler på begäran från andra enheter, en så kallad slav.

CrossCode är enklare I/O-moduler vars typiska tillämpningar ofta kan återfinnas i förar-/operatörshytter där de används till att sköta kommunikation mellan människa och maskin.



Integration of Hardware in Simulation Platform

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

Exempel på sådan kommunikation är genom knappar, styrspakar och lysdioder. Dessa moduler erbjuds även i trådlösa versioner.

CC-Systems produkter följer ett antal standarder och utgör olika så kallade noder i en generisk plattform för styrsystem. Denna samlade plattform av produkter går under namnet CrossTalk.

Innan ett styrsystem kan sättas i drift måste utförlig testning och verifiering av mjukvaran i komponenterna och systemet som helhet utföras för att i största mån undvika fel och obehagliga överraskningar i styrningen. En sådan testning kan vara svår, olämplig eller rent utav omöjlig att utföra med den verkliga hårdvaran på plats i den verkliga tillämpningen. (Nabi et al., 2004) Detta gäller i hög grad många av CC-Systems tillämpningsområden som innefattar tunga maskiner och militära fordon. Det är därför en stor fördel att kunna testa beteendet i en simulerad miljö. För att underlätta en sådan simulering har CC-Systems utvecklat mjukvaran CCSimTech.

CCSimTech beskrivs som en plattform för emulering av hårdvara vilket gör det möjligt att på en PC exekvera stora delar av den kod som sedan kommer att användas i t.ex. en kontrollenhet.

Genom att använda simulering i olika former genom hela utvecklingskedjan för mjukvaran förbättras flödet, detta på grund av mjukvaruutvecklingens minskade beroende av tillgänglig hårdvara. Testningen blir också mer kraftfull då det är lättare att använda sig av avancerade testverktyg på en PC än i målhardvaran.

Under utvecklingen av mjukvara förekommer fyra typer av tester som alla på någon nivå inkluderar CCSimTech.

- Test av undersystem. Ofta testas enstaka undersystem i fristående tester tidigt i utvecklingen av styrsystem. Undersystemens närmiljö är då det enda som simuleras.
- Test av komplett system. Testning av ett komplett system innebär att alla signaler och all kommunikation i systemet simuleras, ofta tillsammans med en modell av något skeende eller en maskin. Även grafiska komponenter som kontrollpaneler med knappar och visare inkluderas i simuleringen.
- Blandad simulering. I blandad simulering förekommer både simulerad och verklig hårdvara. Hårdvara ansluts till den simulerade miljön via ett hårdvarugränssnitt. Vid blandad simulering inkluderar testningen de hårdvaruberoende komponenter som inte kan testas i den rent simulerade miljön. Testningen kan tack vare den i övrigt intakta simulerade miljön utföras i ett kontrollerat steg-för-steg förfarande. Ett ytterligare scenario där blandad simulering kan vara aktuell är där enheten endast finns tillgänglig som verklig hårdvara och saknar simulerad motsvarighet.
- Test på målhardvara. Testning kan utföras med hela styrsystemet i hårdvara. Vid sådan testning kan bl.a. modeller och användargränssnitt fortfarande vara simulerade på PC:n.

Denna rapport berör endast området blandad simulering. (Möller et al., 2005)



Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

1.1 Problemformulering

Elektronik och mjukvara förekommer i allt större utsträckning i alla möjliga olika produkter. Produkternas funktionalitet blir mer och mer beroende av allt mer komplicerade system. Dessa system står också för en allt större del av utvecklingskostnaden. Detta leder till att kraven på att elektronik och mjukvara ska fungera korrekt växer från både användare och utvecklare.

Många tillverkare, inte minst i fordonsbranschen har insett att det finns pengar att spara genom att investera i ordentlig testutrustning. Att likt CC-Systems använda sig av simulering i utvecklingens alla steg är i många branscher snarare regel än undantag och när det kommer till blandad simulering har det senaste åren dykt upp allt mer avancerade system hos olika tillverkare. Som exempel kan tas lastbilstillverkaren Scania som har byggt ett helt laboratorium där upp till 33 kontrollenheter kan anslutas i ett CAN-nätverk (Controller Area Network) och systemets beteende kan testas samtidigt som det illustreras med en detaljerad 3D-modell av en lastbil. (Adenmark, 2008, s.33-35)

CC-Systems kunder har vanligtvis inte krav på simuleringar i samma skala som t.ex. Scania men man ser ändå att behovet av blandad simulering växer. De system som CC-Systems tidigare har utvecklat för blandad simulering har varit skräddarsydda system som utvecklats utifrån specifikationerna på de exakta enheter som kunden valt att ansluta till sin simulering.

En elektronisk kontrollenhet använder sig vanligtvis av ett flertal olika signaltyper och signalnivåer för kommunikation med t.ex. sensorer och ställdon. I ett system för blandad simulering måste därför finnas stöd för dessa olika signaler om enheterna skall gå att ansluta till den simulerade miljön. På grund av detta har tidigare egen signalkonditionering har fått byggas för de olika signalerna.

Nackdelarna med dessa tidigare system är att de dragsits med höga utvecklingskostnader samt att de har tagit tid att utveckla. Systemen har även haft den nackdelen att de varit stora och otympliga att hantera.

CC-Systems har på grund av detta fått upp ögonen för en serie mät- och styrenheter som använder sig av utbytbara signalmoduler. Det innebär att man kan köpa en uppsättning moduler utifrån de signaltyper/-nivåer man kommer att använda sig av och plugga i dessa moduler i en mät-/styrenhet. Förhoppningen är att mät-/styrenheterna kan användas på ett sådant sätt att det går att ersätta de tidigare systemen. Enheterna är både mindre och billigare än de komponenter som tidigare användes.

Vad som gör serien ytterligare mer intressant är att en av mät-/styrenheterna även innehåller en FPGA (Field Programmable Gate Array). En FPGA gör det möjligt att programmera det beteende man vill ha för stunden och sedan exekvera beteendet med många av de fördelarna skräddarsydd hårdvara hade gett med avseende på hastighet och parallellitet.

Traditionellt kallas det system som skall utredas för en bryggning av signaler men denna beskrivning är något missvisande då det inte bara handlar om att föra signaler vidare via ett



Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

gränssnitt utan systemet måste även vara kapabelt att generera och analysera ett antal komplexa signaltyper som i den simulerade miljön endast representeras som numeriska värden.

En av de mest fundamentala skillnaderna jämfört med de gamla systemen är det möjliga införandet av en FPGA. Detta kommer att innebära en omarbetning av hur systemet byggs upp i mjukvara. I arbetet ingår därför att utreda på vilket sätt en FPGA kan användas som en komponent inom blandad simulering. Till vilken grad är det möjligt att tillgodoräkna sig FPGAs fördelar vid generering och läsning av komplexa signaler samtidigt som man behåller systemets konfigurerbarhet?

1.2 Mål och syfte

Syftet med arbetet är att utreda och dokumentera de olika möjligheter och hinder som finns med att använda de olika mät- och styrenheterna från National Instruments C-serie till brygning av signaler mellan en elektronisk kontrollenhet och simuleringsplattformen CCSimTech på en PC.

De utredande delarna av arbetet kommer att ha följande mål:

- Att ta fram en kommunikationsmodell för hur signaler, på ett effektivt sätt, ska bryggas mellan CCSimTech och mät-/styrenheten? Ska brygningen vara händelsestyrd eller tidsstyrd? Vilka kommunikationsmodeller erbjuder mät-/styrenheterna?
- Att ta fram en exekveringsmodell för brygningen av signaler, dvs var ska brygningen göras? Inuti simuleringskärnan eller som en separat applikation eller någon annanstans?
- Att ta fram en arkitektur och design för hur bryggan ska implementeras. Arkitekturen ska möjliggöra:
 - Enkel utökning med nya kontrollenheter samt nya typer av moduler från C-serien.
 - Effektiv exekvering och överföring av signaler mellan simuleringsplattformen och hårdvaran.
 - Enkel administration och konfiguration av nya testsystem.
- Identifiera behov av ytterligare stödapplikationer, t ex administrationsverktyg för att administrera signaler eller ett användargränssnitt för enkel uppstart av ett konfigurerat testsystem.

Målet är att efter utredningsfasen kunna påvisa en princip för ett framtida system. Baserat på denna princip skall sedan en teknikdemonstrator tas fram där man kan använda riktiga kontrollenheter ur CC-Systems produktsortiment tillsammans med simulerade noder och modeller. I demonstratorn ska I/O-signaler bryggas mellan riktiga kontrollenheter och simuleringsplattformen. Denna teknikdemonstrator kommer att kunna användas vid mässor, internutbildningar, externutbildningar och kommer att vara mycket attraktivare som simulerings- och testplattform på marknaden än tidigare tidigare system.



Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

1.3 Avgränsning

Utredningen av mät- och styrenheter har begränsats till National Instruments sortiment och specifikt till de enheter som använder I/O-moduler från C-serien. Anledningen till denna avgränsning är att CC-Systems sedan tidigare har positiva erfarenheter av samarbete med National Instruments och vill undersöka om dessa produkter medför några fördelar jämfört med tidigare system.

Signalernas karaktär kommer på grund av tidsbrist inte att studeras mer ingående än till spänningsnivå.

1.4 Disposition

Denna rapport börjar med att i kapitel 2 ta fram en kravspecifikation för det önskade systemet. I kapitel 3 utreds sedan den aktuella hårdvaran och det slutgiltiga valet av hårdvara motiveras utifrån kraven som ställdes upp i kapitel 2. I kapitel 4 beskrivs all den mjukvara som på något sätt ingår i systemet. Kapitel 5 beskriver hur mjukvaran kan användas för att bygga det önskade systemet. Valen motiveras utifrån användarvänlighet och prestandatester. I kapitel 6 redovisas sedan mina slutsatser om den slutgiltiga lösningen.

Avslutningsvis byggs en teknikdemonstrator baserat på de valda teknikerna. En genomgång av demonstratorn och de slutsatser jag drar efter utvecklingen kan ses i kapitel 7 och 8.



Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

2 STUDIE AV ETT GRÄNSSNITT FÖR BLANDAD SIMULERING

Lösningar för utveckling och testning av elektroniska kontrollenheter erbjuds av ett antal aktörer på marknaden. Den mest namnkunniga av dessa är kanske det tyska företaget dSPACE som utvecklar både egen mjukvara och hårdvara. National Instruments har även ett avtal med tillverkaren Prevas om att leverera systemet xMove som hårdvara för kommunikation med ECUer vid fordonstester. xMove bygger i grund och botten på en compactRIO-enhet ur C-serien. Dessa system är mer avancerade än vad CC Systems kräver och är inte heller avsedda att styras från en Pc-baserad simulering. De är därför inte intressanta att undersöka vidare.

Eftersom CC-Systems tidigare har utvecklat system för blandad simulering fanns kunskap att tillgå inom företaget angående vilka krav som kunde tänkas ställas på ett sådant system.

Ett av de senaste systemen går under namnet AMOS och har utvecklats åt Hägglunds som tillverkar militära fordon.

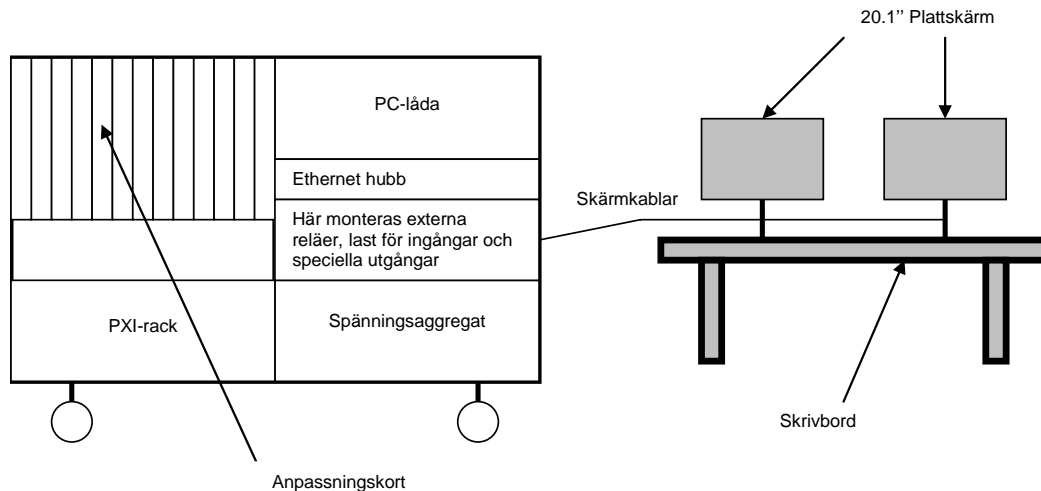
Efter att ha studerat tekniken bakom AMOS-systemet gjordes en kravinsamling för att få en bättre bild av vilken typ av signaler som vanligtvis behöver hanteras vid blandad simulering.

I samband med examensarbetets uppstart höll CC-Systems även kontakt med en potentiell kund (nedan kallad kunden) som sade sig vara intresserad av möjligheten att kunna inkludera sina egna kontrollenheter vid simuleringar. Kundens applikation ansågs vara en typisk tillämpning och de aktuella kontrollenheternas specifikationer fick därför ligga som mall för kraven på systemet.

Utifrån den hårdvara som uppfyllde de ställda kraven utreddes sedan hur det i mjukvara är möjligt att implementera en brygging av I/O-signaler.

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass
Handledare Per Brolin	Datum 08/07/04	Rev PA1
		Filnamn 5256.doc

2.1 Tidigare system



AMOS byggdes 2003-2004 som ett system för blandad simulering hos Hägglunds. Bryggningen av signaler mellan PC och ECU har i detta system gått via ett PXI-rack (National Instruments PCI eXtensions for Instrumentation), bestyckat med ett antal olika I/O-kort tillsammans med specialbyggd anpassningselektronik. CC-Systems har varit nöjda med funktionaliteten i detta system men hårdvaran har varit både skrymmande och dyr att framställa.

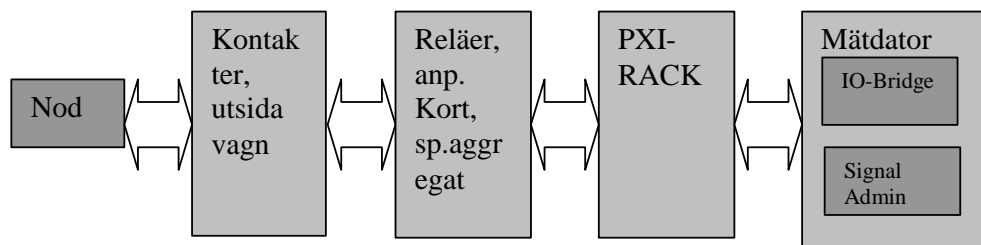
Med AMOS är bryggning av flera typer av kommunikation möjlig bl.a. RS232 (seriell kommunikation). I det system som denna rapport utreder är vi dock bara intresserade av I/O kommunikationen så endast den delen studerades.

På Pc:n finns i AMOS ett program som sköter hanteringen av I/O-bryggningen. Programmet, som heter IOBridge, gör det möjligt för testaren att i en lista ange om en signal används från mjukvara till mjukvara (SW-SW) eller om den ska bryggas via PXI-racket (SW-HW/HW-SW).

Denna arkitektur gör att det snabbt går att byta mellan olika typer av simulering utan större ansträngning för operatören.

I mjukvaran betyder detta att en signallista med information om signalers bryggning analyseras. Beroende på vart varje post pekar gör IOBridge ett anrop till antingen CCSimTech eller till motsvarande funktionalitet i det gränssnitt, kallat NI-DAQmx, som National Instruments använder till mycket av sin hårdvara bland annat PXI-rack. (Sporrong, 2004)

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc



2.2 Kravinsamling

Efter dialog med Per Brolin hos CC-Systems definierades ett mål för lämplig kommunikationshastighet mellan PC och mät-/styrenheten. Kravinsamling utfördes sedan genom kontakt med kunden samt instudering av de kontrollenheter som denne använder sig av samt en av kontrollenheterna i CC-Systems eget sortiment.

Blandad simulering med CCSimTech används sällan för tidskritiska tester och därför ställs inga hårda krav på hastighet och determinism i systemet, något som inte heller är möjligt när en vanlig Pc används för simuleringen. Överföringen av ett signalvärde mellan PC och kontrollenhet bör ändå vara i storleksordningen 10ms för att inte påverka testningen allt för mycket. Just eftersom kravet inte är strikt poängterades noga att kommunikationshastigheten snarare är en punkt för utredning än ett absolut krav.

Kontrollenheterna hos kunden kommer från den finska tillverkaren EPEC och har modellbeteckningen 2024. Den enhet som valdes ur CC-Systems sortiment var CrossFire MX-1 som ansågs vara den vanligast förekommande och som har flest tillämpningsområden på marknaden.

Signaltyp	Egenskap 1	Egenskap 2	Egenskap 3	Egenskap 4	Egenskap 5
Digital in	0-5 V	0-24 V	V _{IH} 4,8-30 V	-	-
Digital ut	0-5 V	0-24 V	-	-	-
Analog in	0-5 V	0-32 V	0-20 mA	0-1100 mA	0-22,7 mA
Analog ut	-	-	-	-	-
PWM	40-2550 Hz @ 24 V	0-100% duty cycle, 10kHz	-	-	-
Räknare	0-60 kHz @ 5V, 24V	-	-	-	-

Tabell 2-1

Tabell 2-1 visar en sammanställning över de signalnivåer som kunden använder sig av. På de punkter fler nivåer förekommer eller om MX-1 använder sig av en annan nivå för motsvarande signaltyp så redovisas även de. Enheternas fullständiga manualer bifogas som bilaga I och II.

Nivåerna är i vissa fall ställbara och tabellen visar i så fall den minimala och maximala nivån.



Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

3 HÅRDVARA

3.1 I/O-moduler i C-serien

C-seriens I/O-moduler finns i över 40 olika utföranden och stödjer många av de vanligaste signaltyperna inom industri och fordon. På modulerna finns skruvterminaler, BNC-, eller D-Subkontakter för anslutning och de har inbyggd signalkonditionering. Modulerna är i storleksordningen av en kortlek och tillverkas och säljs av National Instruments.

För specialtillämpningar finns även tomma ”skal” att köpa för utveckling av egna moduler. Man kan via National Instruments hemsida hitta företag som säljer vidare sina specialbyggda moduler. Ett exempel är det amerikanska företaget Drivven* som använder C-seriens moduler i utvecklingen av motorstyrenheter och har utvecklat ett 10-tal moduler för speciella motortillämpningar.

National Instruments har ett antal mät- och styrenheter där en eller flera moduler ur C-serien kan kopplas in för anpassning till olika tillämpningar. De flesta modulerna är kompatibla med samtliga mät-/styrenheter men undantag finns.

Varje modul kan som mest innehålla 32-kanaler och ha upp till 16-bitars upplösning. På grund av antalet signaler i CC-Systems tillämpningar är vi endast intresserade av mät-/styrenheter som kan använda sig av minst fyra moduler samtidigt.

(*”Deploying from USB to Embedded with C-Series Hardware”*, 2008)

3.2 Compact DAQ (Data Acquisition)

CompactDAQ (nedan benämnd cDAQ) är en enhet som ansluter till PC:n via USB. cDAQ har plats för totalt 8 I/O-moduler och kan hantera upp till 256 signaler. ”Hi-Speed”-USB-anslutning till Pc gör enheten lätt att hantera och erbjuder en överföringshastighet på upp till 480Mbit/s.

cDAQ levereras med drivrutiner som går under namnet NI-DAQmx och är samma som används i AMOS. En av de stora fördelarna med NI-DAQmx är att den har ett väl utvecklat API (application programming interface) som gör cDAQ-enhetens samtliga funktioner tillgängliga i flera olika utvecklingsmiljöer som t.ex. NI LabVIEW och C/C++.

För signaler som är beroende av en mer exakt klockning än vad som kan åstadkommas via USB-kommunikation så finns möjlighet att använda sig av två inbyggda räknare. Detta betyder att man kan generera pulser eller räkna inkommande frekvenser med hög noggrannhet. Dessa räknare kan dock bara användas av en ”process” åt gången vilket t.ex. gör att man maximalt kan generera två högfrekventa pulser med olika frekvens åt gången.



Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

3.3 Compact RIO (Reconfigurable Input/Output)

CompactRIO (nedan benämnd cRIO) är en fristående mät-/styrenhet som är byggd för att stå emot tuffa yttre påfrestningar. cRIO finns i utföranden för 4 eller 8 I/O-moduler. Förutom de anslutna I/O-modulerna innehåller enheterna två huvudsakliga delar, en realtidsdator samt en FPGA (Field Programmable Gate Array).

3.3.1 Realtidsdator

Realtidsdatorn i cRIO-enheterna finns i dagsläget med tre olika processorer på upp mellan 195 och 400MHz. Datorerna har mellan 32 MB och 128 MB DRAM samt upp till 2GB för lagring med möjlighet till extra lagringsutrymme via USB-port.

På datorn körs realtidsoperativsystemet VxWorks från WindRiver, samma operativsystem som används av NASA i fordon för utforskning av mars.

cRIO används vanligtvis som ett inbyggt system d.v.s. att den oftast programmeras för en viss funktion och är inte beroende av en användares styrning via t.ex. mus eller tangentbord. På realtidsdatorn finns därför bara en möjlighet att kommunicera med en PC och det är via 10/100Mbps Ethernet. För att ändå öka kommunikationsmöjligheterna har realtidsdatorn inbyggd http-server och ftp-server.

3.3.2 FPGA

FPGA-kretsen i cRIO kommer från tillverkaren Xilinx.

En FPGA är en enhet bestående av rekonfigurerbar grindlogik. En FPGA kan programmeras att utföra logiska operationer efter önskemål. FPGA:ers användningsområden växer ständigt. Pris och prestanda gör att man kan stjäla marknadsandelar både från områden som tidigare krävt processorer och från områden som krävt specialutvecklade kretsar.

Till skillnad från en processor som bara kan utföra en beräkning åt gången erbjuder FPGA full parallellitet och till skillnad från specialbyggda kretsar kan funktionen hos FPGA:n ändras via mjukvara.

FPGA:erna i cRIO har en klockfrekvens på 40MHz och finns i utföranden med en eller tre miljoner grindar.

FPGA:n i cRIO har en direktanslutning både till realtidsdatorn och till I/O och kan därför utföra avancerade I/O-operationer med mycket hög hastighet parallellt med att datautbyte med realtidsdatorn sker. Detta gör FPGA:n till en idealisk komponent för klockning och trigging av I/O. (*Developing Custom Measurement and Control I/O Hardware with the LabVIEW FPGA Module and Reconfigurable I/O Hardware*, 2007)

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

3.4 Evaluering av hårdvara

En jämförelse visar att modulerna i C-Serien över lag ger en god matchning mot de krav som ställts upp för systemet.

Tabell 3-1 visar förslag på de moduler som matchar kraven. Sammanställningen visar på vissa brister hos cDAQ-enheterna som ledde till beslutet att inte utreda en sådan lösning vidare.

Signaltyp	Egenskap 1	Egenskap 2	Egenskap 3	Egenskap 4	Egenskap 5
Digital in	0-5 V NI 9401	0-24 V NI 9474	V _{IH} 4,8-30 V NI 9474	-	-
Digital ut	0-5 V NI 9401	0-24 V NI 9423	0-30 V NI 9423	-	-
Analog in	0-5 V NI 9263	0-32 V N/A	0-20 mA NI 9265	0-1100 mA N/A	0-22,7 mA N/A
Analog ut	-	-	-	-	-
PWM	40-2550 Hz @ 24 V NI 9423	0-100% duty cycle, 10kHz @ 24 V NI 9423	-	-	-
Räknare	0-60 kHz @ 5V, 24V NI 9401, NI 9474	-	-	-	-

Tabell 3-1, Matchning. Insignaler på kontrollenheten matchas mot moduler med motsvarande utsignaler. Modul finns för: (cDAQ+cRIO), (begränsad användning i cDAQ), (endast till cRIO)

De nackdelar som cDAQ uppvisar är:

En begränsning i att bara kunna använda två olika externt klockade signaler. Detta är en nödvändighet för att kunna läsa och generera pulssignaler och andra komplexa mönster med bra upplösning.

National Instruments tekniska säljare, Johan Olsson¹, ställer sig också tveksam till att använda sig av mer än en cDAQ-enhet kopplad till Pc:n åt gången, främst på grund av osäkerhet över hur prestanda för enheterna påverkas. Detta måste vara möjligt då fler signaler än cDAQ-enhetens maximala 256 kan komma att förekomma i ett system.

Modulerna i C-serien uppfyller alla utom tre i krav i listan. Det är kravet på en modul för spänningsstyrning 0-32V samt det krav som kunden har angivit om att kunna leverera

¹ Johan Olsson, Sr. District Sales Manager, National Instruments, Sverige. Möte och e-post korrespondens under vecka 6 2008.



Integration of Hardware in Simulation Platform

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

strömstyrning på 0-1100mA och 0-22,7mA som inte kan tillgodoses. Sådana moduler saknas i National Instruments sortiment och kan inte heller hittas hos andra tillverkare.

Kravet på strömstyrning upp till 1100mA kommer direkt från kundens specifikationer men anses inte av CC-Systems vara en typisk signalnivå i ett styrsystem (jfr strömstyrning 0-20mA på MX1) och man godtar därför tillsvidare den modul som levererar 0-20mA som tillräcklig.

Kravet på spänningsstyrning på 0-32V kommer direkt från databladet för MX-1 och kan inte uppfyllas av modulerna i C-serien. Som en väg runt detta problem kan man tills vidare utnyttja möjligheten på MX-1 att kunna ställa analoga ingångarna till att acceptera antingen 0-5V eller 0-32V. För att i framtiden kunna använda sig av nivån 0-32V kan det bli nödvändigt att utveckla sin egen modul för detta ändamål. Med dessa tre begränsningar i åtanke togs beslutet att ändå arbeta vidare med compactRIO som potentiell lösning.



Examensarbetare Emil Vestman	Dok Nr 1	Säk klass
Handledare Per Brolin	Datum 08/07/04	Rev PA1
		Filnamn 5256.doc

4 MJUKVARA

4.1 CCSimTech

CCSimTech är en plattform för emulering av hårdvara och körs på en vanlig PC. Plattformen möjliggör, som tidigare beskrivits, utveckling och testning av kontrollmjukvara utan tillgång till målhårdvaran. Genom att kunna ersätta alla hårdvaruspecifika delar i kontrollapplikationen med simulerade motsvarigheter har man möjlighet att testa den exakta koden som sedan kommer att flyttas till kontrollenheten. Detta går ofta relativt lätt att göra då kontrollapplikationens kod ofta är uppdelad i hårdvaruberoende och hårdvaruoberoende delar.

Detta tillvägagångssätt är ett bra verktyg men har dock vissa brister:

- Drivrutiner måste fortfarande utvecklas på målhårdvaran.
- Timingen vid exekvering av koden är olika mellan PC och målhårdvara.
- Hänsyn måste tas till operativsystemets olika egenskaper vid t.ex. synkronisering.

Är man medveten om dessa brister i CCSimTech så går det ofta att vidta åtgärder för att simuleringen skall hamna tillräckligt nära verkligheten för de flesta testsituationerna. (Möller et al. 2005)

Den del av CCSimTech som berör detta arbetet är hanteringen av I/O, SimIO. I SimIO har varje signal ett namn (ett "handle") och ett signalvärde bestående av ett 32 bitar stort heltal. Vad värdet verkligen representerar kan vara olika beroende på applikation. Det kan t.ex. vara frekvens eller duty-cycle hos en PWM-signal eller bara representera ett analogt värde. SimTech gör ingen skillnad på dessa signaltyper utan det är upp till användaren att använda värdena som sig bör i sina applikationer.

4.2 C/C++

CC-Systems använder sig ofta av programmeringsspråket C/C++ för att skapa komponenter till simuleringen såsom maskinmodeller och kontrollpaneler. Genom CCSimTechs API kommer man åt de signaler som kontrollmjukvaran använder sig av och det blir lätt att få komponenterna att interagera.

I det tidigare AMOS-systemet var bryggningsapplikationen IOBridge skriven i C++.

4.3 LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) är ett grafiskt programmeringsspråk från National Instruments. Den grafiska miljön med "klippa-klistra"-komponenter gör att man lätt kommer igång och snabbt kan utveckla nya program. Utvecklingsmiljön i LabVIEW består av två delar, "front panel" och "block diagram". Alla applikationer utvecklade i LabVIEW består av dessa två komponenter. För att förklara



Integration of Hardware in Simulation Platform

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass
Handledare Per Brolin	Datum 08/07/04	Rev PA1
		Filnamn 5256.doc

uppbyggnaden av en sådan applikation kan de båda komponenterna liknas vid ovan- och undersidan på ett operatörsbord.

”Front panel” representerar ovansidan och innehåller endast grafiska representationer av knappar, visare, textfält och andra grafiska komponenter. Ingen logik finns på ”front panel”. ”Block diagram” är som att kika in på baksidan av operatörsbordet, varje komponent som placerats på ”front panel” går att komma åt på ”block diagram”. Utifrån detta kan man sedan bygga sin applikation.

För användning med cRIO krävs två utbyggnadsmoduler till LabVIEW.

LabVIEW RT (Real Time) är en avskalad version av LabVIEW med vissa tillagda funktioner som kan vara användbara i realtidsapplikationer för synkronisering av uppgifter. Det finns också funktioner för kommunikation med en FPGA. LabVIEW RT används, som namnet antyder för att utveckla applikationer till realtidsdatorn i cRIO-systemet.

LabVIEW FPGA Module används för att bygga FPGA:ns beteende. Traditionellt konfigureras FPGA:er med ett lågnivåspråk som VHDL eller Verilog som beskriver det beteende man vill åstadkomma eller den hårdvarustruktur man vill efterlikna. Dessa språk har en relativt hög inlärningströskel och utvecklingen kan bli tidskrävande.

LabVIEW FPGA erbjuder FPGA-programmering med samma syntax och med många av de programmeringskomponenterna som finns tillgängliga på PC och RT. Tillsammans med speciella I/O funktioner förkortar detta utvecklingstiden avsevärt och gör FPGA-tekniken tillgänglig för fler användare. När LabVIEW-programmet är färdigutvecklat görs en automatisk analysering och översättning till VHDL som sedan kompileras till den bitström som matas till FPGA:n.

Vid utveckling av kod till FPGA kan LabVIEW-koden påminna mycket om den på en PC men det finns en del grundläggande begränsningar att ha i åtanke. Dessa begränsningar grundar sig i de komponenter som finns tillgängliga på FPGA:n.

Några av de begränsningar som kan ha stor inverkan är:

- Utrymmet (antal grindar) på en FPGA är alltid begränsat. Det kan medföra att man kan få göra avkall på mer avancerad kod när man bygger större applikationer.
- En FPGA kan bara arbeta med heltal. Det är möjligt att låta en del av heltalet representera vad som skulle ha varit decimaler men operationer på sådana tal kan vara svåra att genomföra.
- Division finns inte tillgänglig som operation. Det går i många fall att genomföra en division med hjälp av andra operationer men ett sådant förfarande kan visa sig vara långsamt och/eller ta upp stor plats på FPGA:n.

LabVIEW RT och LabVIEW FPGA är väl inbäddade i LabVIEW-plattformen och helheten underlättar mycket vid utveckling av applikationer som involverar funktionalitet hos hela cRIO-systemet.



Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

5 ARKITEKTUR FÖR BRYGGNING AV SIGNALER

För arkitekturen ställdes först upp önskemål och avgränsningar. Sedan utreddes de olika delarna i arkitekturen för att försöka uppfylla dessa önskemål.

5.1 Arkitektur för FPGA

FPGA:n är den enhet som har direkt anslutning till I/O. Därför måste den programmeras att hantera detta även om det bara är genom att införa ett enkelt beteende som att vidarebefordra signaler/värden mellan FPGA:n och realtidsdatorn. Ytterligare beteende som klockning och trigging av signaler bör också göras i FPGA-koden om man vill uppnå hög noggrannhet. (*"Developing Custom Measurement and Control I/O Hardware with the LabVIEW FPGA Module and Reconfigurable I/O Hardware"*, 2007)

FPGA:n måste alltid programmeras om utifrån varje ny uppsättning I/O-moduler i cRIO-chassiet, deras placering och användning. Därför bör en struktur eftersträvas där sådana ändringar är lätta för en operatör att utföra utan att behöva djupare förståelse i LabVIEW.

En sådan användarvänlighet kan åstadkommas till exempel genom att tillhandahålla färdigskrivna "block" med grafisk kod, operatörens uppgift blir då att klistra in de korrekta blocken utifrån den uppsättning moduler denne använder. Detta är för övrigt det vanliga sättet att återanvända LabVIEW FPGA-kod och block innehållande olika typer av funktioner kan delas mellan användare värden över. (*"LabVIEW FPGA Design for Core Modules (IP Cores)"*, 2006)

5.1.1 Svårigheter med återanvändningsbar kod

Att hålla FPGA-koden konfigurerbar medför vissa svårigheter. Svårigheterna uppstår främst i och med de I/O funktioner med vilka anrop till I/O måste göras. Dessa funktioner kräver att adresseringen till I/O-portarna i cRIO-chassiet är statiska vid kompilering d.v.s. inga ändringar i en ports beteende går att göra efter att koden kompilerats.

5.1.2 Val av arkitektur

Den arkitektur som ansågs mest användarvänlig resulterade i skapandet av ett antal färdiga funktionsblock. Blocken är specifika med avseende på I/O-modultyp och funktionalitet.

Vid användning av blocken blir operatörens uppgift att välja rätt block från en meny utifrån de moduler denne använder och den funktionalitet som önskas. Efter att ha valt den korrekta uppsättningen block måste operatören även göra en inställning per I/O-kanal. Denna inställning görs enkelt från en rullista på FPGA-programmets "front panel". Inställningen är nödvändig för att adresseringen av I/O skall bli korrekt och för att koden ska bli möjlig att kompilera.



Integration of Hardware in Simulation Platform

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

De nackdelar som kan ses med den valda arkitekturen är att den använder sig av komponenter som är krävande utrymmesmässigt på FPGA:n. Dessa komponenter är "Array" och "Cluster" och varje block innehåller en komponent av vardera. ("Using Clusters and Arrays in LabVIEW FPGA", 2006) Dessa valdes på grund av sin överlägsna användarvänlighet och med tanke på den ringa omfattningen av block som används (ett per I/O-modul) så bör detta inte medföra utrymmesbrist.

För att göra FPGA-koden mer effektiv är blocken, i de fall I/O-funktionerna stödjer det, avsedda att exekvera i en cyklisk struktur som kallas Single-Cycle Timed Loop (SCTL). Detta medför en automatisk utrymmeseffektivisering vid kompileringen samtidigt som koden är garanterad att ha en cykeltid på ett klocksteg (25ns vid 40MHz). I bilaga III, "fpga utilization chart", framgår hur stor utrymmesbesparing denna effektivisering medför. Dokumentet gäller för situationer där den totala användningen av FPGA:s utrymme är mellan 10% och 95 %. Vid större total användning ökar effektiviteten hos kompilatorn ytterligare men med följden att kompileringen blir mer tidskrävande.

5.2 Arkitektur för kommunikation mellan RT och PC

För kommunikation med simuleringens värddator finns på cRIO bara en möjlighet och det är via Ethernet. Möjligheterna att utforma denna kommunikation är desto fler. Två lösningar som är lätta att implementera på både realtidsdator och PC undersöktes. Lösningarna uppfyller förutom den efterfrågade enkelheten också kraven på att kommunikationen skall gå att utvidga i framtida system samtidigt som de tillåter hög kommunikationshastighet.

5.2.1 TCP/IP

TCP/IP är en vida implementerad kommunikationsarkitektur för kommunikation över Ethernet. Arkitekturen är omfattande men ger följande viktiga egenskaper:

TCP (Transmission Control Protocol) garanterar förlustfri överföring av data mellan två enheter.

IP (Internet Protocol) ser till att varje enhet på ett nät har en unik adress. IP-protokollet ser även till att hitta vägar genom nätet mellan två enheter.

TCP/IP har blivit populärt efter som det stödjer många av de vardagliga behoven för datautbyte som filutbyte och e-post. ("Building Networked Applications with the LabWindows/CVI TCP Support Library", 2006)

Implementering av TCP/IP-kommunikation kräver inga stora programstrukturer vare sig på realtidsdator eller PC. De anrop och den datamanipulation som behövs på respektive sida är möjlig att göra i både C++ och LabVIEW. En aspekt som kan anses tala emot användningen av TCP/IP är att man alltid måste känna till vilken data som har skickats för att på mottagarsidan kunna återskapa korrekta datastrukturer. Efterforskningar visar att det i LabVIEW även går att använda funktioner vid kommunikationen som bortser från vissa delar i TCP/IP arkitekturen



Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

för att effektivisera snabbt utbyte av mindre mängder data ("Do LabVIEW TCP Functions Use the Nagle Algorithm?", 2008 och "A Simple TCP/IP Messaging Protocol for LabVIEW", 2007).

Hur snabbt kommunikationen i slutänden kan ske via TCP/IP är svårt att bedöma och kan påverkas mycket av vilken typ av data som skickas över nätet och hur den struktureras.

5.2.2 Shared Variables

National Instruments presenterar sin teknik, Shared Variables, som en allmän kommunikationsarkitektur. Kommunikationstekniken bakom Shared är i grunden baserad på TCP/IP men har ovanpå det en stack av ytterligare lager. Översta lagret under LabVIEW bygger i Shared Variables på en process som är väsentlig för att förstå hur Shared Variables fungerar och kan användas. Processen har benämningen Shared Variable Engine (SVE) och måste exekvera på minst ett av systemen i datautbytet. Funktionsmässigt fungerar SVE som ett bibliotek för de olika variablerna och gör så att variablerna alltid är tillgängliga för de processer som behöver dem.

Shared Variables finns som tre olika typer med olika tillämpningsområden.

Local Variables – En variabel som är allmänt nåbar mellan processer på en dator.

Time Triggered – En typ av variabel som är avsedd för deterministisk kommunikation mellan realtidsenheter.

Network Published Variables – En typ av variabel som möjliggör lätt utbyte av data oavsett om utbytet sker mellan två lokala loopar eller över nätverk till ett annat system.

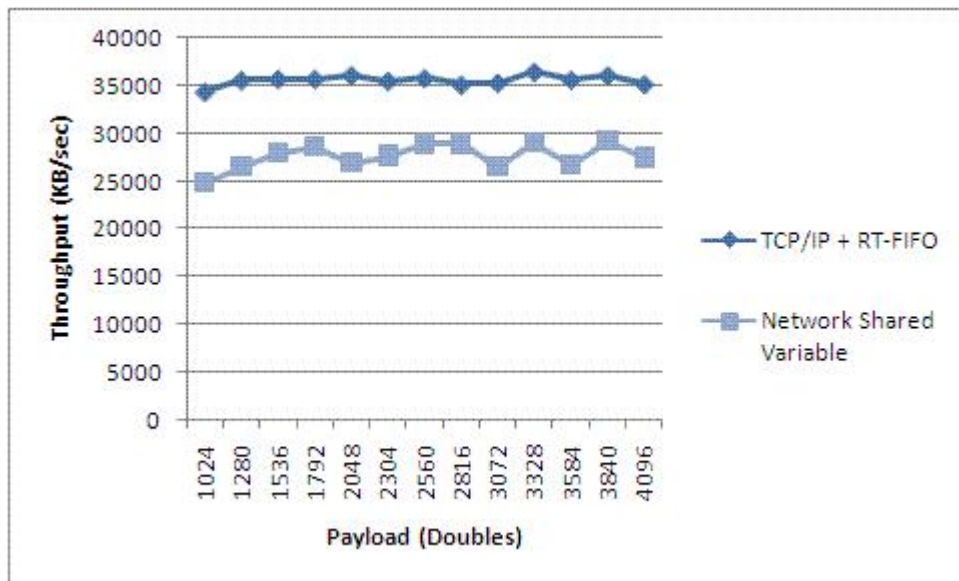
("Using the LabVIEW Shared Variable", 2008).

5.3 Val och testning av vald teknik

Valet mellan Shared Variables och TCP/IP är en avvägning av användarvänlighet kontra överföringshastighet. National Instruments har testat överföringshastighet och redovisar på sin hemsida typiska värden för de båda teknikerna i ett antal olika situationer medan användarvänligheten är svårare att bedöma.

En jämförelse mellan överföringskapaciteten hos Network Published Shared Variables och en enkel TCP/IP implementering kan ses i Fig 5-1. Vård- och måldator i testen består av en Pc och en realtidsdator med högre prestanda än den i cRIO. Mängden överförd data per sekund ligger konstant något lägre för Shared Variables. Skillnaden påverkas heller inte nämnvärt beroende på hur stor den totala mängden data är ("Using the LabVIEW Shared Variable", 2008).

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass
Handledare Per Brolin	Datum 08/07/04	Rev PA1
		Filnamn 5256.doc



Figur 5-1

CC-Systems beslutade utifrån denna information att gå vidare med att försöka få till stånd en lösning med Network Published Shared Variables. Beslutet motiverades med att överföringshastigheten var acceptabel och den enklare implementeringen skulle undvika att utvecklingen av det kompletta systemet blev lidande.

För att kunna använda kommunikation baserad på Shared variables från en applikation skriven i C++ (vilket kommer att vara fallet på Pc-sidan i och med gränssnittet mot CCSimTech) måste egna metoder för läsning och skrivning först utvecklas i LabVIEW. Dessa metoder måste gå att anropa från C++-applikationen. För att detta skall vara möjligt måste anropningsbar kod i form av en DLL (Dynamic-Link Library) skapas av LabVIEW-koden.

Denna DLL anropas sedan cyklisk för att uppdatera signaler och signalvärden och ger en struktur mycket lik den hos IOBridge i AMOS-projektet.

För att bekräfta Shared Variables som en möjlig lösning testades den faktiska överföringshastigheten. Testerna utfördes med en eller två variabler för dubbel- eller enkelriktad skrivning av data mellan Pc och cRIO. Data i variablerna strukturerades som en Array bestående av 32 bitars representationer av signalvärden. Av testningen konstaterades tidigt att överföringshastigheten förbättrades avsevärt om processen SVE exekverade på Pc:n istället för på realtidsdatorn.

Testerna utfördes genom att på ena sidan skriva ett nytt värde till den aktuella variabeln varje cykel och på andra sidan mäta tiden det tar innan ett nytt värde kan läsas.

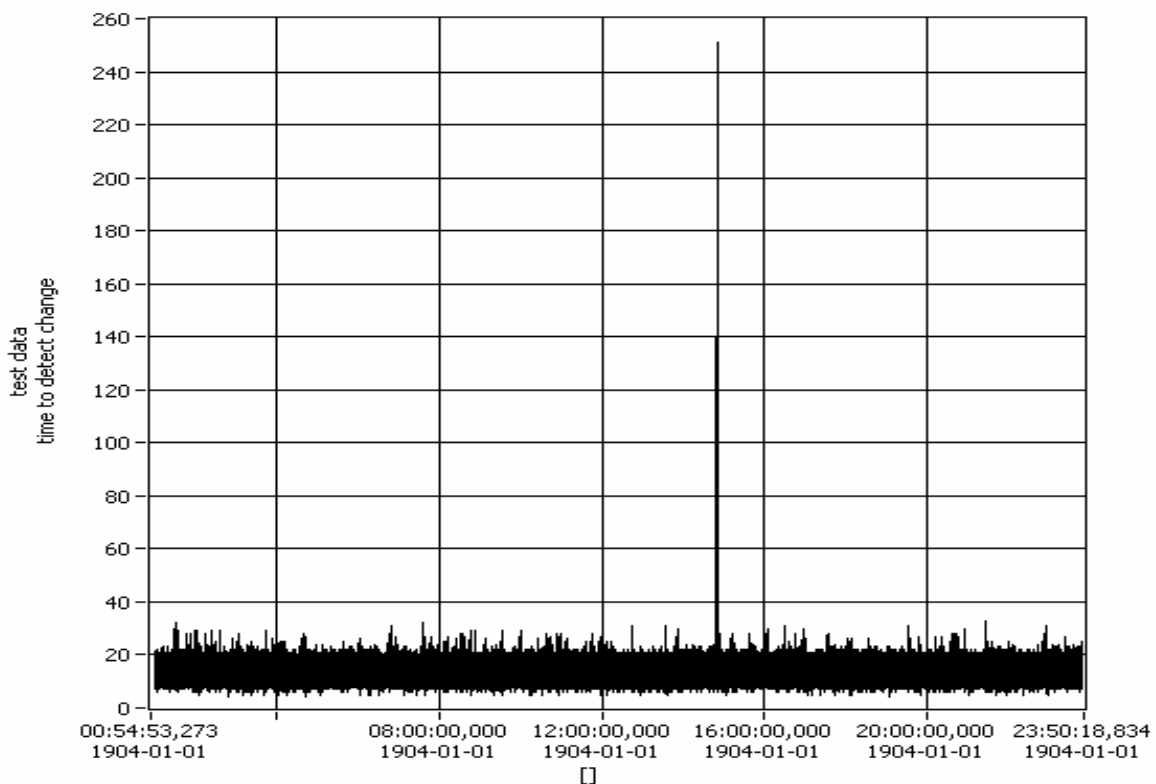
Tester gjordes för att utröna olika faktorer inverkan på kommunikationen. Tre faktorer visade sig ha stor inverkan på beteendet hos Shared Variables.

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

- Shared Variables använder sig av en inbyggd buffring av nätverkskommunikationen. Buffertens storlek är 8kB och om den inte fylls upp skickas data ändå efter 10ms. Anledningen till buffringen är att det är mer effektivt att sända stora datamängder än små.
- Antalet variabler i SVE.
- Cykeltidern för läs- och skrivlooparna. Olika värden testades för att hitta en så snabb kommunikation som möjligt.

Testerna visade att det genom att använda sig av minst 8KB data i variablerna går att få till stånd en omedelbar sändning av data utan påverkan av den inbyggda buffringen. Enligt testerna gav en extra utfyllnad av variablerna med data till 8KB ökade möjligheter att minska cykeltiden för skrivning och läsning.

När cykeltiderna minskar ner mot de nivåer vi önskade uppnå kunde dock ett oönskat beteende observeras i form av spikar. Spikarna uppstod till synes slumpmässigt och troddes bero på överbelastning av processen SVE och borde därför också vara beroende av prestanda på den Pc som SVE exekverar på. Spikarna uppstår endast då man använder sig av tekniken med datautfyllnad. Figur 5-2 visar ett exempel på en typisk spik i storleksordningen 250ms.

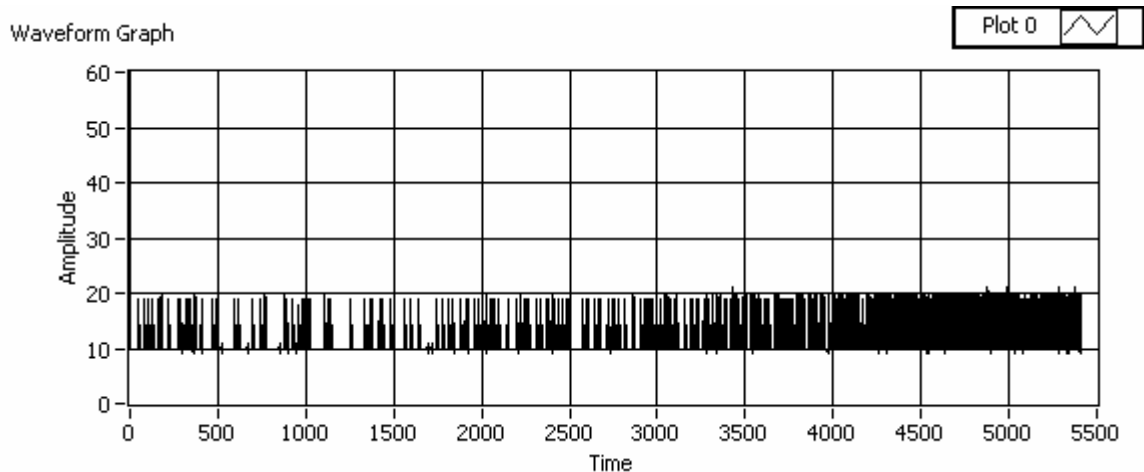


Figur 5-2

För att undvika spikar gjordes fortsatta tester utan utfyllnadsdata. Beteendet blev på detta sätt mer förutsägbart och med en skriv-loop och en läs-loop med cykeltider på vardera omkring 10

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

ms kunde optimal prestanda observeras. Detta betydde att en dataändring kunde detekteras på mellan 10 och 20 ms. Figur 5-3 visar detta beteende.



Figur 5-3

Genom att endast använda en variabel för nätverkskommunikation gick det att göra ytterligare prestandavinster. Denna struktur valdes dock bort på grund av att skrivning till samma variabel då skulle behöva ske på båda sidor vilket ger en risk att värden från en sida kan skrivas över med äldre värden på andra sidan.

5.4 Sammanfattning av mjukvaruarkitektur

Bryggningen från SimTech bör ske via en applikation skriven i C++. Detta för att övriga applikationer som kommunicerar med SimTech ofta skrivs i C++ och en lätt integrering med denna arkitektur är att föredra.

Detta innebär två olika möjligheter. Antingen implementeras kommunikation i C++ som kan kommunicera med LabVIEW-applikationen på realtidsdatorn, eller så implementeras kommunikationen i LabVIEW även på PC-sidan och gör denna kod anropningsbar från C++. För att kunna använda sig av Shared Variables är endast det senare fallet möjligt.



Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

6 DISKUSSION OCH SLUTSATSER (TEKNIKSTUDIE)

Studien visar på goda möjligheter att med cRIO och dess inbyggda FPGA utveckla ett gränssnitt för anslutning av elektroniska kontrollenheter till en PC. Det finns hos National Instruments moduler som stödjer flertalet av de signaler som kommer att vara aktuella för CC-Systems.

FPGAs kraftfullhet gör det möjligt att använda sig av både enkla och mer komplexa signaler. FPGA:n medför dock även nackdelar med avseende på användarvänlighet. Ändringar i testupställningen kommer med cRIO alltid att medföra behov av programmering. Genom att utveckla färdiga bibliotek för FPGA-programmeringen kan kraven på operatören minska till en acceptabel nivå.

De beräkningsbegränsningar som finns i och med FPGA:n kan ha viss påverkan på systemet. Bland annat så kan man komma att vilja representera signalvärden i SimTech med någon intuitiv enhet så som millivolt eller ms för en pulsperiod. Eftersom funktionerna på FPGA:n arbetar med binära nivåer för analoga signaler och antal klocktick (25ns) för pulser så behöver en omräkning till dessa enheter göras på vägen. Vill man ha denna funktionalitet är det enklaste att även för detta skapa standardiserade block som är avsedda för användning på realtidsdatorn. Görs detta förlorar man dock möjligheten att alltid använda samma applikation på realtidsdatorn.

För kommunikation mellan Pc och cRIO finns ett flertal vägar att gå. Efter att ha testat den lösning som framstår som den enklaste konstateras att redan den uppfyller de önskemål som finns på prestanda. Skulle en mer specialiserad TCP/IP-lösning behövas bör inte heller detta medföra några svårigheter att införa.

Att ha i åtanke när det gäller cRIO är även den kostnad det medför för kunden att alltid behöva tillgång till LabVIEW:s utvecklingsmiljö.



Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

7 TANKSKÖTAREN THOMAS

Teknikdemonstratorn, som utvecklats för att illustrera och pröva de resultat som teknikstudien gett, har jag valt att kalla Tankskötaren Thomas. Demonstratorn visar ett enkelt system för blandad simulering med en given uppsättning I/O. Hårdvaran som integrerats i simuleringen är en CrossCode CX, en av CC-Systems enklare enheter för I/O.

Demonstratorn använder de delar som utvecklats under studien. Det färdiga FPGA-biblioteket och en arkitektur för användning av Shared Variables på både PC och realtidsdator.

På PC-sidan görs Shared Variable-anropen i en DLL som anropas från en avskalad version av den ursprungliga IOBridge från AMOS-systemet. DLL:en är utvecklad i LabVIEW

CC-Systems önskemål på DLL:en var att skrivningen och läsningen skulle gå att göra portvis till och från cRIO-chassiet. Därför skapades metoder för läsning och skrivning som har modulnummer och portnummer som argument. Sammanlagt består DLL:en av fyra metoder med följande funktionalitet:

- Initiering av Shared Variables
- Läsning – av ett signalvärde från lokal variabel
- Skrivning – av ett signalvärde till lokal variabel
- Uppdatering – sorterar de lokala variabler som ska sändas i en Network Published Shared Variable samt läser den andra och sorterar data till lokala variabler.

Funktionen för uppdatering behöver veta hur lokala variablerna ska sorteras i de två nätverks variablerna. För detta skapades en enkel stödapplikation där användaren anger hur många kanaler det finns på varje modul och om den ska användas för insignaler eller utsignaler.

De moduler som används i demonstratorn är NI 9401 för digitala insignaler (5 V) och NI 9263 för analoga utsignaler ($\pm 10V$ men med CrossCode CX används endast intervallet 0-5V).

Den matematiska modellen i systemet består av en vattentank med ett inflöde, ett utflöde och ett värmeelement. Tanken är modellerad av Sebastian Krüger (CC Systems) i ett angränsande projekt. Modellen är byggd som en fristående applikation i LabVIEW och inte i C++ som annars vore typiskt. Modellen har 3 digitala och 3 analoga signaler:

Digitala insignaler:

- Öppna inflöde
- Öppna utflöde
- Aktivering av värmeelement

Analoga utsignaler:

- Vätskans nivå i tanken
- Vätskans temperatur i tanken
- Summa in-/utflöde



Integration of Hardware in Simulation Platform

Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

Alla 6 signaler skrivs och läses av tankmodellen direkt till/från CCSimTech.

Processen styrs och övervakas via en simulerad operatörspanel med tre knappar och tre indikatorer som utvecklats i det grafiska programmeringsspråket CoDeSys. Applikationen utvecklades av Tobias Andersson (CC Systems) i syfte att användas i demonstratorn.

Kontrollpanelen kommunicerar endast via CAN genom en USB-to-CAN adapter på Pc:n. CAN-trafiken innehåller värdet på in- och utportarna.



Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

8 DISKUSSION OCH SLUTSATSER KRING TANKSKÖTAREN THOMAS

Demonstratorn var tack vare det utvecklade biblioteket mycket lätt att framställa (och rekonfigurera med avseende på I/O). Även om demonstratorn är en väldigt enkel simulering som bara använder sig av grundläggande I/O så finns inga tecken på att systemet inte skulle gå att skala upp eller att det skulle innebära någon som helst skillnad att använda sig av mer avancerade signaltyper.

När hela systemet implementerats och provkördes kunde man ibland ana en liten fördröjning innan signalerna bryggats.

För att analysera beteendet gjordes tester. För testningen användes en digital utport som byglades till en digital inport. De båda portarna bryggades sedan till simulerade motsvarigheter.

Ett värde skrevs till den simulerade utporten och fördröjningen tills värdet kunde läsas på inporten beräknades. Fördröjningen visade sig vara ett till synes slumpmässigt värde mellan 40 och 90 ms. Det anses vara oförklarligt högt med tanke på tidigare tester som borde ge maximal fördröjning på 30-40 ms.

Felsökning har kunnat utesluta IOBridge-applikationen på PC:n. Om inte vidare felsökning kan finna någon felaktighet så kan detta ge anledning att se över kommunikationsmodellen.

9 FRAMTIDA ARBETE

Efter denna studie planerar CC-Systems att gå vidare med lösningen baserad på compactRIO. Man vill försöka utveckla brygningssystemet till en kommersiell produkt att användas tillsammans med CCSimTech.

Man har även börjar titta på möjligheten att utveckla sina egna I/O moduler till cRIO. Ett önskemål är att modulerna ska gå att göra skräddarsydda för anslutning till CC-Systems egna enheter.

Parallellt arbete utförs även med att kunna utnyttja LabVIEW bättre eftersom det i och med cRIO blir en obligatorisk komponent. Till exempel ser man över möjligheten utveckla eller integrera färdiga maskinmodeller och testsekvenser direkt i LabVIEW.



Examensarbetare Emil Vestman	Dok Nr 1	Säk klass	
Handledare Per Brolin	Datum 08/07/04	Rev PA1	Filnamn 5256.doc

REFERENSER

Tidskrifter och manualer

Adenmark, Mikael (2008), *Elektroniktest av komplett lastbil*, Elektronik i Norden, nr 7/2008, s. 33-35.

Sporrong, Ulf (2004), *Användarmanual AMOS bänktestsystem*, Rev. PA5, CC Systems.

Möller, Anders, Åberg, Per, Löwenhielm, Fredrik, Brundin, Jakob, Nolin, Mikael & Engblom, Jakob (2005), *Developing and testing distributed CAN-based real-time control-systems using a single PC*, Mälardalen University, Department of Computer Science and Electronics, 2005.

Nabi, Syed, Balike, Mahesh, Allen, Jace & Rzemien, Kevin (2004), *An Overview of Hardware-in-the-Loop Testing Systems at Visteon*, SAE International, 2004-01-1240

Internet

<Deploying from USB to Embedded with C-Series Hardware>, <Uppdaterad 2006-08-06>
<http://zone.ni.com/devzone/cda/tut/p/id/6418>

<Developing Custom Measurement and Control I/O Hardware with the LabVIEW FPGA Module and Reconfigurable I/O Hardware>, <Uppdaterad 2007-06-27>
<http://zone.ni.com/devzone/cda/tut/p/id/3261>

<Building Networked Applications with the LabWindows/CVI TCP Support Library>, <Uppdaterad 2006-09-06>
<http://zone.ni.com/devzone/cda/tut/p/id/3067>

<Do LabVIEW TCP Functions Use the Nagle Algorithm?>, <Uppdaterad 2008-01-08>
<http://digital.ni.com/public.nsf/allkb/7EFCA5D83B59DFDC86256D60007F5839>

<A Simple TCP/IP Messaging Protocol for LabVIEW>, <Uppdaterad 2007-09-20>
<http://zone.ni.com/devzone/cda/tut/p/id/4095>

<Using Clusters and Arrays in LabVIEW FPGA>, <Uppdaterad 2006-10-17>
<http://zone.ni.com/devzone/cda/tut/p/id/4520>

<Using the LabVIEW Shared Variable>, <Uppdaterad 2008-02-14>
<http://zone.ni.com/devzone/cda/tut/p/id/4679>

<LabVIEW FPGA Design for Code Modules (IP Cores)>, <Uppdaterad 2006-09-28>
<http://zone.ni.com/devzone/cda/tut/p/id/5340>