

eurosource

En plug and play lösning för billig styrning och kontroll i SOHO miljö.

Abstract

Projektet syftar till att på billigaste enklaste och optimalt standardiserade sätt bygga moduler som är intelligenta, "legolika" och kan användas i styr och mät sammanhang. Varje modul byggs upp enligt samma öppna grundprincip
<http://www.vscp.org> <http://www.eurosource.se/catalog>

- Modulen kan generera ett eller flera Event.
- Modulen kan reagera på en (eller flera) Action(s).
- Modulen kan fatta ett eller fler beslut utifrån mottagna Event eller kombinationer av dessa och antingen generera ett nytt Event eller en Action.

En modul är autonom till sin funktion och har en grund funktionalitet som är väldefinierad även innan den har konfigurerats.

En modul har ett globalt 128-bitars ID som unikt identifierar den.

Moduler kan kopplas samman i segment (nätverk) med en CAN implementation VSCP (Very Simple Control Protocol) (<http://www.vscp.org>) som gör att sammansatt funktion, där flera moduler samverkar med varandra, kan åstadkommas på ett enkelt sätt. VSCP är designat för att fungera väl på CAN, TCP/IP, Ethernet, RF, IR och RS-232/422/485.

All konfiguration av moduler använder sig av den abstraktion som så framgångsrikt använts inom hårdvaru-industrin. Man kan läsa och skriva register. Enheten kan begära service med interrupt.

Systemet är plug-and-play och moduler kan läggas till eller tas bort under drift.

Modulerna paketeras i normkapslingar. Fördelen är att lådor redan finns tillgängliga utan att plastverktyg behöver tas fram samt att kapslingar för olika miljökrav också finns tillgängliga på hyllan. Normkapslingen är dessutom känd och accepterad på marknaden.

Sammantaget får man en modul i en standardiserad förpackning med ett lågt pris som kan utföra en väl specificerad enkel styrfunktion. Flera sammankopplade moduler kan sedan lika enkelt och med ett minimum av konfiguration utföra komplexa styruppgifter.



En grunden att stå på.

Hur tanken växte fram

Sommaren 1984 började tanken växa fram om autonoma elektroniska enheter som kunde användas för olika styrändamål. Tanken var att konstruera enheter som var så billiga att de kunde sitta i varje lampa och i varje liten apparat. Eftersom Motorola processorerna användes i min dagliga gärning så var det naturligt att försöka använda dessa processorer till de intelligenta noderna.

Komponent kostnaderna för en nod hamnade någonstans i området 2K-3K SEK indikerande att man skulle få produkter med ett försäljningspris per nod på 6K-9K. Knappast en prisnivå som gjorde modulerna attraktiva som komponenter i lampor och andra lågpris applikationer.

Några tankar växte fram under den här tiden som fortfarande är vägledande för visionen.

1. En nod skall ha en autonom och väldefinierad funktion dvs. den är en fungerande enhet även utan inblandning från yttre enheter. Ett typiskt exempel kan vara en termostat som styr på och frånslaget av en last utifrån inställda temperatur gränser.
2. Genom att koppla ihop noden med andra enheter så kan man få en ny eller utökad funktion.
3. En nod skall vara unikt identifierbar bland andra noder.
4. Systemet skall automatiskt anpassa sig till att nya noder adderas/tas bort till/från systemet.

Under de år som följde har sedan många protokoll, buss topologier, kretslösningar och tekniker förkastats p.g.a. kostnader eller tekniska begränsningar.

M.U.M.I.N. - Multi Unit Micro Interface Network

M.U.M.I.N. växte fram under den här tiden. Ethernet och CAN och liknande var för dyra att använda i SOHO lösningar och RS-485 baserade nät kvarstod som den enda lösning som med god säkerhet skulle kunna ge en tillförlitlig lösning i extremt billiga noder för styrsystem.

M.U.M.I.N. definierar ett protokoll som uppfyller de fyra punkterna ovan men som också definierar en ny punkt.

Alla noder beskrivs med en abstraktion som är välkänd (och använd med framgång) från elektronikvärlden nämligen den som IC kretsar använder. Man kan läsa och eller skriva från/till register samt att enheten kan begära service genom att generera ett interrupt.

Vi definierar en enhets funktion genom betydelsen för dess register. Detta betyder att alla mekanismer för att hantera noden i övrigt blir gemensam för samtliga noder.

I M.U.M.I.N. abstraktionen används också ett 128-bits id för att åstadkomma ett GUID (Globally Unique ID)

M.U.M.I.N. kan användas även för PLC (Power Line Communication), RF, IR och andra lösningar.

M.U.M.I.N. har ersatts av VSCP och byggts in helt i detta protokoll.

En model för styrning av inbäddade system

Under Hösten, vintern 2000/2001 deltog jag i ett projekt för Frontyard koncernen (<http://www.frontyard.com>) där vi för e2home's (<http://www.e2-home.com>) räkning skissade på lösningar för "intelligenta lägenheter". Elektrolux Screenfridge var en del i det här konceptet som fick mycket press. Under det här arbetet, där LON, EIB var nyckelkomponenter och där OSGi (Open Service Gateway initiative, <http://www.osgi.org/>) var en mycket viktig komponent blev behovet av en billigare och enklare teknik uppenbar.

För att försöka lösa de problem som dessa tekniker brottades med så var vi några stycken som tog fram en lösning som i motsats till det Java centrerade OSGi inte var låst till en utvecklings miljö och plattform. En lösning som dessutom lämpade sig för system där hårdvarukostnaden är kritisk.

Den specifikation som vi kom fram till **EDA** (*Event, Decision, Action*) publicerades som en öppen specifikation på Sourceforge projektet (<http://www.vscp.org>).

EDA (Event, Decision, Action) Hur och varför?

EDA togs fram av **Marcus Rejås, Behzad Ardakani** och **Åke Hedman** under hösten 2000.

Varför EDA?

Att bygga system som är extremt stabila och tillförlitliga kräver ett annat tänkesätt än det som är brukligt vid traditionell systemutveckling. Det traditionella sättet att tänka är att utveckla kontrollerbara och förståeliga del komponenter som på olika sätt interagerar med varandra. Säkerhet och kontrollfunktioner hängs på i olika lager för att se till att systemet arbetar som det är tänkt.

I inbyggda system måste man ofta tänka på ett annat sätt i så måtto att den funktion som utförs alltid måste göra sitt jobb så länge den fysiska hårdvaran fungerar. Jämför t.ex. en server i ett nätverk och en mobiltelefon. Hos en server är det oftast acceptabelt att starta om en eller kanske ett par gånger per år utan att betrakta den som en dålig systemenhet. En mobiltelefon däremot skall alltid utföra sin uppgift och behöver den någon form av återstart så skall den utföra den helst utan att vi märker det. Funktionaliteten skall alltid finnas tillgänglig när användaren behöver den. Den senare måste dessutom vara extremt kostnadseffektiv.

Detta sätt att tänka är standard för dem som utvecklar realtidssystem och inbyggda system. Utmaningen idag är inte att komplexiteten växer, jämför de inbäddade styrsystemet för en skogsprocessor eller stridsflygplan typ JAS, utan att fler inbäddade produkter behöver tas fram på kortare tid. Detta skall göras av utvecklare som inte har erfarenhet av sådan utveckling utan istället är van vid traditionell Unix/Windows programmering.

En annan viktig aspekt som vi måste ta hänsyn till är att världen förändras i en rasande fart. Vi måste nå flexibilitet utan att göra avkall på funktion och stabilitet. Skapar vi rigida system eller satsar på "fel" standard utan möjlighet att anpassa oss skapar oundvikligen stora framtida kostnader i form av nyutveckling.

Tre punkter måste följaktligen vara uppfyllda

Stabilitet skall finnas inbyggd i systemet utan att lägga hela ansvaret på utvecklaren.

Systemet skall vara modulärt uppbyggt med abstraktioner mot omvärlden så att det lätt och billigt kan anpassas till nya standarder/trender.

Vi skall underlätta för ”traditionella” programmerare att skapa komponenter som är byggda med det synsätt som är vanligt i inbäddade system.

Hur löses detta med EDA?

EDA som står för *Event, Decision, Action* är ett försök till lösning på dessa problem. Det är viktigt att förstå att EDA anpassar sig till världen och de olika standarder som finns i den. Modellen är en utvecklingsmodell och inget försök att själv skapa en standard för t.ex. residential gateway teknologi.

EDA har ett bestämt sätt att se på världen och har därför ett antal lager som anpassar världen så att den stämmer med EDA's synsätt.

EDA tror att alla sensorer och kontroll enheter (SECO devices från Sensor/Control devices) är intelligenta. I EDA's värld behöver man bara tala om för en seco enhet att den skall rapportera en händelse när den inträffar (larm, leverans av temperatur mätning, en dörr som öppnas). Det enda som en sensor kan göra är därför att generera en händelse, ett EVENT. Till denna händelse kan någon form av data vara kopplad. EDA bryr sig inte om formatet på denna data utan förväntar sig att mottagaren (som har intresse av eventet) har (eller skaffar sig) denna kännedom.

EDA är beredd att reagera på något sätt på en händelse. För att kunna reagera så måste ett beslut (DECISION) tas som talar om vilken reaktion som skall göras. Beslut om vad som skall göras när en händelse inträffar har en benägenhet att förändras över tiden. EDA tar hänsyn till detta. I princip behövs ingen programmerare för att ändra ett beslut. Beslutsunderlagen lagras i en beslutsmatris.

Varje beslut resulterar i att något utförs. Detta kan vara en eller flera saker samtidigt eller ett antal saker som sker i följd. I EDA terminologi är detta en ACTION.

Modellen ger dessutom abstraktioner för ett antal tjänster som förväntas bli använda av del komponenter i systemet.

Den tillåter yttre styrning/kontroll/inladdning/urladdning av del komponenter i systemet oberoende vilken standard som används av världen utanför. (UPnP, OSGi etc.).

Den ger möjlighet att enkelt och billigt förändra händelsemönstret som skall utföras när en specifik händelse inträffar. I princip krävs ingen kvalificerad personal för att göra sådana förändringar.

Den ger en enhetlig metod för att läsa/skriva/ändra konfigurations parametrar.

Den ger en enhetlig metod för att skriva till en databas och logga systeminformation.

EDA modellen ställer dessutom inga förväntningar på den hårdvara den körs på. Är det ett UNIX system, en Windows' burk eller ett rent inbäddat system så exporteras samma tjänster och samma grundfunktioner finns programmerad i systemet. Samma utvecklingsmodell och klasser används i de olika miljöerna.

Grundtanken är att man når stabilitet genom att koncentrera så mycket som möjligt av funktionaliteten till en enda stabil del som är statisk utvecklingsmässigt över tiden. När man senare, under ett utvecklingsprojekt, tillför funktionalitet till systemet så kan detta göras i diskreta testbara steg.

Modellen gör att samma del komponenter kan användas för att lösa mycket olika problem. Samma kod kan t.ex. användas för att läsa temperaturen för vattnet i en tvättmaskin som den kod som används för att kontinuerligt avläsa temperaturen i vardagsrummet i en villa.

EDA är en öppen lösning som finns tillgänglig både som specifikation och i källkodsform på

<http://www.vscp.org> Tanken med detta är att få fler intresserad av att jobba med modellen samt att få kritik, idéer och utvecklingsinsatser från fler människor runt om i världen. Detta är ett sätt att snabbt och kostnadseffektivt nå stabilitet och maximal funktionalitet.

EDA beskriver och implementerar bara modellen för ett kostnadseffektivt styrsystem den säger inget om hårdvara eller kommunikationsvägar. Trådburen-, Radio-, IR-kommunikation platsar alla i konceptet.

VSCP – Very Simple Control Protocol

Att kommunicera över kabel är den naturliga lösningen om man ändå skall dra en kabel till en enhet. Anledningen till att kabel behövs är oftast att enheten behöver strömförsörjning och att man inte vill använda batteridrift. Väldigt många fasta installationer hamnar i detta område.

CAN (Control Area Network) från Bosch har länge setts som den perfekta lösningen för styrning och kontroll i tuffa miljöer som fordon och liknande. Finns t.ex. i alla fordon från personbilar till lastbilar och militära fordon idag. Tills alldeles nyligen har det varit omöjligt att implementera CAN i kostnadskritiska lösningar men detta har ändrats radikalt under det senaste året. Idag ger en CAN lösning även en extremt kostnadseffektiv lösning om man siktar på det vid konstruktionen.

För att göra det enkelt och smidigt att åstadkomma lösningar så har VSCP tagits fram. Detta protokoll är optimerat för att fungera bra på CAN men kan med fördel även användas i PLC, Radio, IR och över TCP/IP nät.

De fem punkterna från ovan är uppfyllda

1. En nod skall ha en autonom och väldefinierad funktion dvs. den är en fungerande enhet även utan inblandning från yttre enheter.
2. Genom att koppla ihop noden med andra enheter så kan man få en ny eller utökad funktion.
3. En nod skall vara unikt identifierbar bland andra noder.
4. Systemet skall automatiskt anpassa sig till att nya noder adderas/tas bort till/från systemet.
5. Alla noder beskrivs med en abstraktion som är välkänd (och använd med framgång) från elektronikvärlden nämligen den som IC kretsar använder. Man kan läsa och eller skriva från/till register samt att enheten kan begära service genom att generera ett interrupt.

VSCP är en öppen specifikation och finns öppet tillgänglig tillsammans med CAN drivare och CANAL (CAN Abstraction Layer) på <http://www.sourceforge.net/projects/can>).

Öppna specifikationer

Tanken med dom öppna specifikationerna är att ge andra möjlighet att ge synpunkter och komma med idéer samtidigt som det ger en väg in för andra att tillverka komponenter. Detta är positivt efter som fler tillgängliga kompatibla komponenter ger ett attraktivare system ur kommersiell synvinkel. M.U.M.I.N., EDA och VSCP är tillsammans med de vedertagna tekniker som finns idag USB, CAN, TCP/IP, Bluetooth mm grundstenarna i detta koncept.

System lösning.

Att inte bygga på accepterade standarder är detsamma som att begå självmord. Så alla system lösningar bör i hörligaste mån bygga på en eller fler accepterade standarder. Måste man bygga proprietära lösningar och inte har obegränsade ekonomiska resurser för att tvinga igenom dem så att de blir accepterade standarder så skall man göra detta öppet. På detta sätt kan man få andra att acceptera lösningen och kanske få fler att bidra med del lösningar och kunskap och på så sätt snabbt nå kritisk massa.

Vårt koncept riktar sig till SOHO (Small Office, Home) lösningar i första hand. Idag kan man nästan förutsätta att det finns ett TCP/IP nät som går över Ethernet samt en eller flera PC datorer i dessa miljöer. Kännedom om och användande av USB är inte heller främmande. Bluetooth, WLAN och liknande är, om inte redan känt, på väg in med stormsteg.

Fördelen med alla dessa tekniker är att det finns billig utrustning och kunnande för att koppla samman dem. En HUB eller switch för ett Ethernet eller en trådlös Ethernet access punkt kan köpas för en billig penning i vilken databutik som helst. Att koppla ihop utrustningen är inte heller ett stort problem och om det upplevs som det så behövs ingen dyr specialkompetens hyras in för att få det hela att fungera.

Ethernet, USB kan snabbt identifieras som två nyckeltekniker på denna nivå. Bluetooth är antagligen en starkt kommande teknik. Olika former av PLC lösningar är också på väg in. Men de två första måste stödjas.

Med VSCP kan Ethernet stödjas och man får en fullständig *Plug-and-Play* lösning. **OSGi**, **UPnP**(Universal Plug And Play), **JINI** kan enkelt stödjas med samma moduler. En existerande PC miljö kan stödja systemet eller utnyttja sig av dess funktioner.

USB ger tillgång till en rik flora av tillbehör så att inte stödja tekniken är att kasta bort mycket gratis mervärde. En kontrollmodul med Ethernet interface kan t.ex. få en kamerafunktion genom att ansluta en standard USB kamera. Variationerna är i det närmaste oändliga bara genom att använda standardkomponenter.

Att använda Ethernet på den allra lägsta nivån är normalt inte kostnadseffektivt. Här är CAN en bättre lösning. En modul kan skapas för en låg kostnad och säljas med god marginal. Med VSCP nås fullständig *Plug-and-Play* funktionalitet även här. Dallas Semiconductors 1-Wire och LIN kan användas för att nå ännu mer kostnadseffektiva lösningar. Detta kan t.ex. vara sensorer för att känna om ett fönster är öppet eller för att mäta en temperatur.

Hur fungerar modulerna?

I EDA så definieras en SECO-enhet (Sensor Control) som en enhet som kan generera ett antal event och som kan kontrolleras med ett antal styrkommandon. Vi har också i M.U.M.I.N. och VSCP specificerat att kommunikationen med en enhet skall ske med en register modell.

En typisk enhet kan alltså sända noll eller flera meddelanden. Typiska exempel kan vara

- Temperaturen är nu 10 grader Celsius.
- Strömförbrukningen är nu 3.1KW.
- Knappen "*knapp*" har trycktes in.

En enhet som är intresserad av dessa meddelanden måste veta vad de innebär och kan i så fall lyssna på den och enligt EDA modellen ta ett beslut och utföra en handling.

En enhet kan också styras med noll, ett eller flera kommandon. Typiskt kan vara att man konfigurerar den genom att skriva ett värde i ett register eller att man ger den ett kommando genom att skriva i dess kommando register.

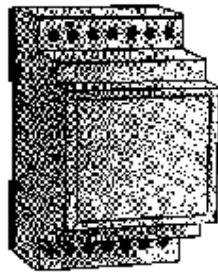
Typiska kommandon kan vara

- Dimma ljus till 43%
- Sätt övre tröskelnivå för temperatur.
- Aktivera larm.

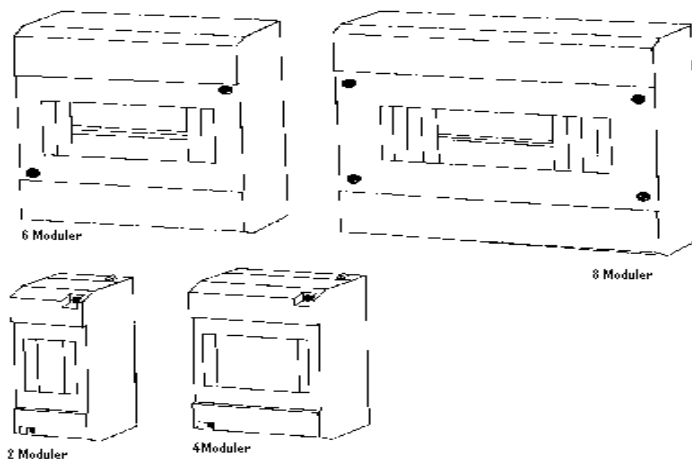
Moduler till salu.

Förpackningen

Normkapslingar på DIN skena har blivit enormt populära under senare år. Inte minst därför att de flesta fastigheter som får en ny el-installation får den i form av normkapsling. Normkapslingen syns också ofta inom styr och regler industrin. Det finns flera fördelar.



- Det finns ett stort antal färdig modul lådor (som passar i normkapsling) för konstruktioner tillgängliga. Dvs. det finns inga behov av att initialt ta investera i dyra plastverktyg.
- Branschen känner till normkapslingen och är vana att hantera utrustning förpackad i denna form.
- Normkapslingarna finns för olika IP klassningar. Det intressant är att det är normkapslingen och inte modulen som sköter miljö skyddet. Detta får till följd att samma modul kan användas inomhus som utomhus om olika normkapslingar används.
- Normkapsling finns redan installerad hos många potentiella kunder varför man snabbt kan bygga ett nytt system. Fler kommer att ha normkapslad utrustning i framtiden dvs. Trenden är ökande.



Lego för styrsystem lösningar

VSCP & EDA ger ett sant modulära system när det gäller de mjuka delarna. Modul kapslingen och normkapslingen ger en modularitet när det gäller de hårda delarna.

Traditionellt tänker man sensorer, I/O portar. Sensorer skall avläsas. I/O portar skall sättas/återställas. Till detta använder man olika styrsystem som programmeras av någon programmerings kunnig. Här behövs detta i princip inte alls (även om det är möjligt). Detta kan enklast åskådliggöras med ett antal exempel.

Exempel: Energi styrning i fastighet.

Systemet består av två moduler.

Effektmätare-module

Event:

1. Momentant ström-uttag på fas1-3 i A.
2. Momentant RMS effekttuttag i KW/h.
3. Ackumulerat RMS effekttuttag i KW/h,

Actions:

1. Nollställ ackumulerat effekttuttag.
2. Ange hur ofta ström-uttag event skall sändas (t.ex. varje sekund).
3. Ange hur ofta effekttuttag event skall sändas på bussen.
4. Ange hur ofta totalt effekttuttag event skall sändas på bussen.

Apparatmodul

Event:

1. Tillslag/frånslag

Actions:

1. Slå till eller från apparat

Decision

1. Om strömuttag på fas 1 är $\geq z$ slå av (event 1 för elmätare modul)
2. Om strömuttag på fas 1 är $< z$ -hysteres slå på (event 1 för elmätare modul)

Med två moduler och minimal programmering har man skapat ett system där en adaptiv styrning av en fastighets apparater enkelt utförs. Detta kan t.ex. vara att tvätt maskinen inte körs om strömuttaget är för stor eller kanske man stänger av varmvattenberedaren.

Notera att modulariteten består även i så måtto att flera apparater kan kontrolleras lika enkelt genom att addera apparatmoduler.

Exempel: Radiatorstyrning för ett rum

Systemet består av en kombinerad temperatur/hygrometer modul och en eller flera apparatmoduler enligt ovan.

Temperaturmätare-module

Event:

1. Temperatur + förändring i temperatur.
2. Relativ fuktighet + förändring i fuktighet.

Actions:

1. Ange hur ofta temperatur event skall sändas (t.ex. varje sekund).
2. Ange hur ofta relativ fukt event skall sändas på bussen.
3. Sätt temperatur larmpunkter (max/min).

Apparatmodulen enligt ovan kontrollerar en radiator (vatten eller el spelar ingen roll). I VSCP systemet sänds tids information varför dygns variationer och veckovariationer i styrmönstret enkelt kan konfigureras.

Skall flera radiatorer styras adderas bara en till apparatmodul.

Vill man ta hänsyn till förändringar i ute-temperatur så adderar man en temperaturmätarmodul som mäter utomhustemperatur. Man kan då enkelt få systemet att förbereda sig när ute temperaturen t.ex. faller mycket.

En vindsensormodul kan adderas för att ta hänsyn till köldeffekten .

Exempel: "God natt"- knapp

Ett antal moduler kan läggas in för att få en "god natt" funktion.

RC5-mottagar modul (Standard IR kontroll)

Event:

1. RC5 tecken mottaget.
2. RC5 styrsekvens mottagen

Actions:

1. Definiera en RC5 styrfrekvens.

Generell IO modul

Event:

1. I/O förändring.
2. "Custom event"

Actions:

1. Konfigurera "custom event" för "förändring".

Att antal apparat moduler ingår också i systemet.

Man kan nu enkelt konfigurera ett system som släcker ett utvalt antal lampor, sätter på larmet, ser till att spisen är avstängd osv, osv när god natt knappen vid sängen aktiveras eller när TV's fjärrkontroll ger ett konfigurerat kommando.

Några andra konfigurerings exempel...

Intelligenta larm kan enkelt och billigt byggas. Generella I/O moduler kan anslutas till befintliga larmsystem, magnet detektorer, IR-detektorer. Apparat moduler kan aktivera eller stänga av apparater eller larmkällor. SMS moduler kan meddela externa larm mottagar källor. Kamera moduler kan ta bilder.

Man kan bygga system som automatiskt ser till att spisen eller vattnet är avstängt när man lämnar hemmet.

Man kan enkelt bygga intelligenta hjälpsystem för äldre så att man kan bo hemma istället för på servicehus längre.

Arkitektur.

Lego-moduler i all ära är bra men är kostnaderna för en modul för hög så faller hela tanken med systemet. Därför är det viktigt att välja lösningar som ligger i huvudfråan teknik- och marknads-
mässigt.

CAN har tidigare varit en dyr men säker lösning för att koppla ihop noder. Idag är det inte längre så. CAN är numera, tack vare att fler aktörer har anträt marknaden med nya kretslösningar, också en mycket kostnadseffektiv teknologi. En nod kan konstrueras och säljas för några tior om man inte har för stora krav på dess intelligens. VSCP konstruerades speciellt med tanke på CAN just pga. den inbyggda feltoleransen som protokollet har. CAN är en självskriven kandidat

Ethernet och TCP/IP är naturliga val just pga. den mångfald av komponenter och lösningar som finns. En Ethernet nod kostar marginellt mer än en nod byggd med annan teknik. WLAN lösningar finns som ger mycket kostnadseffektiva trådlösa lösningar. VSCP fungerar utmärkt även här.

PLC (Power Line Communication) lösningar börjar duka upp liksom billiga RF lösningar. VSCP fungerar bra i båda dessa fall.

Koncepter kan samverka med andra SOHO lösningar eller bli en integrerad del i dem. OSGi, Home Plug and Play, EIB , LON, X10 kan alla samverka i större system där våra moduler är delkomponenter.

Ett större system byggs lämpligen med Ethernet så att de allmänt kända och kostnadseffektiva systemkomponenterna kan användas. På lägre nivå används CAN i segment där en nod är Ethernet ansluten och fungerar som master på segmentet.

I mindre system behövs bara ett CAN segment. Om behov finns att utöka funktionaliteten kan detta enkelt göras med en Ethernet modul som kopplar ihop segmentet med

Sammanfattning

Det finns ett behov av enkla styrsystem för SOHO marknaden. Idag finns väldigt få system på marknaden som löser de typiska SOHO problemen på ett enkelt, effektivt och billigt sätt. X10 är den teknik som kommer närmast men ändå inte hela vägen fram. LON och EIB och liknande tekniker ligger kostnadsmässigt långt borta för de flesta. Andra lösningar har kommit fram som mer inriktat sig på tekniken som sitt egentliga värde istället för problemlösning (IP-lösningen, Linux-lösningen, Radiolösningen är typiska). Dessa kommer att förbli marginella aktörer på marknaden eftersom kunderna inte förstår försäljningsargumenten.

Det system som beskrivs här skulle kunna fylla ett tomrum som finns på den globala styr och regler marknaden idag.



Dokument historia

2002-10-25 akhe@eurosource.se
Inledande version av dokumentet.

