

# **ELEKTOR** Allt om **Elektronik**

Sveriges största specialtidning för tillämpad elektronik



**Mellanvågsradio  
i midgetformat**



**Digital  
volym-  
kontroll**

## **Laddningsregulator för solpaneler**

**Hastighetsmätare  
för modellbilar**

**32-kanals  
ingångskort  
för PC**

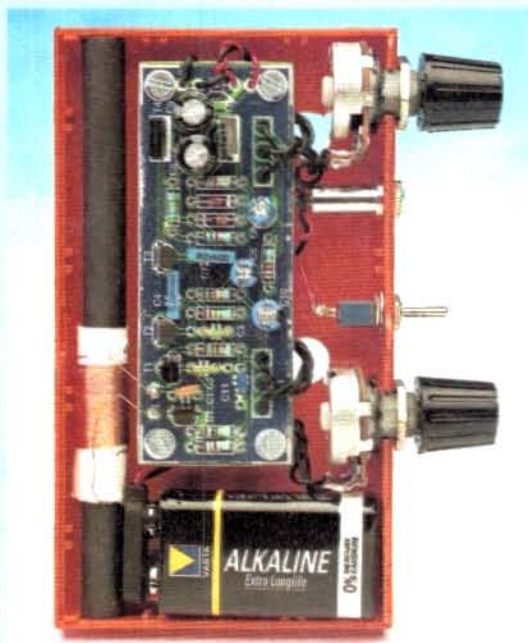
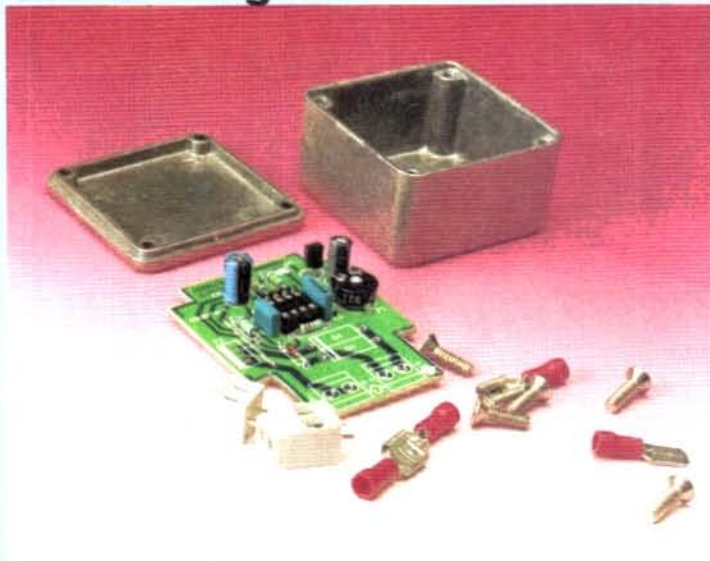
**Multimeter  
med Java**



INTERPRESS 0173-03  
SPECIALRETUR



## Solkraftsregulator



## Mellanvågsradio

### Digital volymkontroll



## Nummer 3 2000

- 4 Solkraftsregulator**  
*För paneler upp till 53 watt*
- 8 Mellanvågsmottagare i midgetformat**  
*Enkelt bygge för en regnig dag*
- 12 BASIC Stamp - programmeringskurs**  
*Del 7: subsumptionsprogrammering*
- 16 Digital volymkontroll**  
*För Audio DAC 2000*
- 21 Elektronik on-line**  
*Bygg din egen plotter*
- 22 Hastighetsmätare för modellbilar**  
*Med en cykeldator*
- 26 Temperaturmätning med DS1621**  
*En smartare sensor med bussinterface för I<sup>2</sup>C*
- 30 Nya audioformat**  
*Hög upplösning och nya egenskaper*
- 36 32-kanals ingångskort för PC**
- 40 I/U-kurvföljare**
- 43 ATX-omkopplare**
- 44 Javamultimeter**
- 46 LED-display**

# Laddningsregulator för solanläggning

## för paneler upp till 53 watt

En liten solkraftsinstallation består av åtminstone tre delar, en solpanel, ett lagringsbatteri och en laddningsregulator. Laddningsregulatorn begränsar den slutliga laddningsspänningen och skyddar också mot omvänt strömflöde.

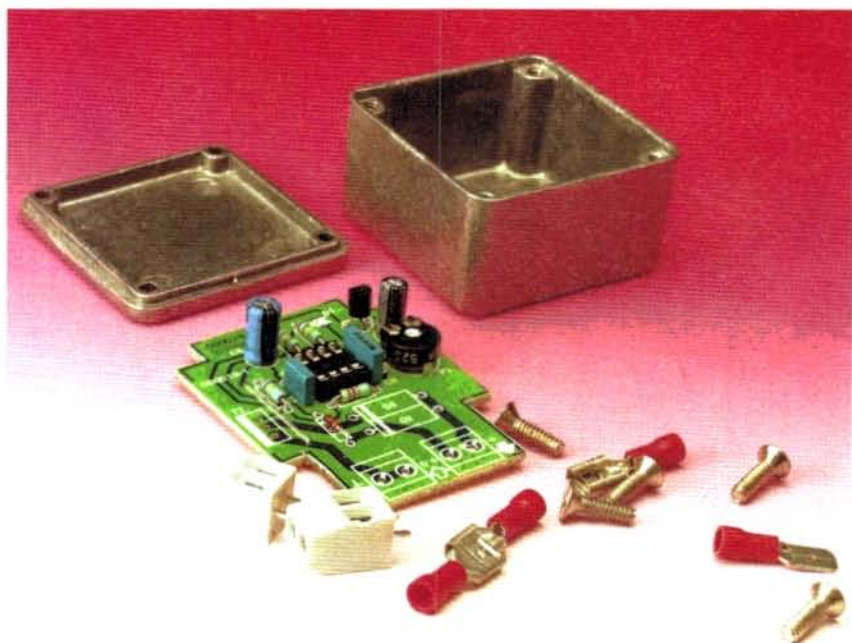
Design av H-Tronic, Hirschau, Tyskland

### Specifikationer för solladdningsregulatorn

Matningsspänning	12 V, från batteriet
Solpanel	upp till 53 W
Arbetsström	2.1 mA
Spänningsfall	0.43 V vid 3 A
Temperaturområde	-10 to +45 °C
Skydd mot kortslutning och omvänd polaritet	

Det var rätt länge sedan vi hade en artikel om solkraftsteknik, och eftersom utvecklingen gått framåt kände vi att det var dags för en sådan.

I en självförsörjande solkraftsinstallation som kan lämna elektrisk energi även när det är dåligt väder, eller när det är mörkt ute, är en energireserv i form av ett vanligt blybatteri absolut nödvändigt. För att förhindra att batteriet laddas ur via solpanelen när panelens terminalsänkning sjunker un-



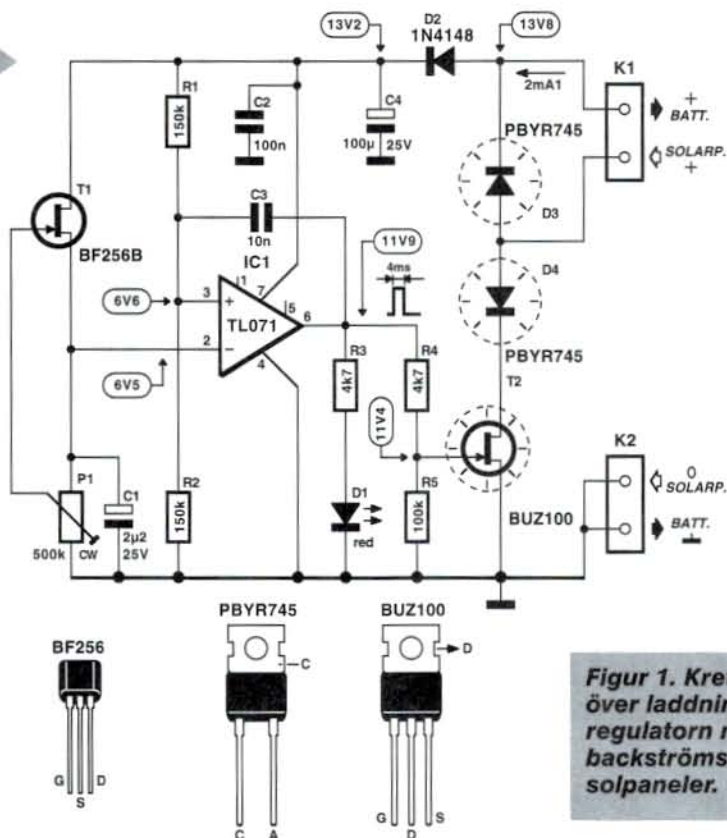
der batterispänningens nivå, är det nödvändigt med ett backströmsskydd. I sin enklaste form består en sådan 'solströmskran' av en diod. En Schottkydiod, som har ett lågt framspänningsfall, används normalt för att minimera förlusterna.

Tyvärr är terminalsänningen hos en 12-V solpanel betydligt högre än den nominella märksänningen när den utsätts för starkt solljus, så det går inte att undvika att polspänningen hos ett fullt laddat batteri överskrider när man bara använder en diod. Om man lägger en för hög spänning till ett batteri så produceras gas, vilket förkortar batteriets livslängd. Det kan också vara farligt eftersom denna gas är explosiv. Det är alltså nödvändigt med någon form av regulatorkrets, förutom dioden, för att begränsa laddningsspänningen hos batteriet till 2,3 V per cell (lika med 13,8 V för ett 12 V batteri). Den regulatorkrets som vi presenterar här uppfyller dessa två krav, skydd mot omvänt strömflöde och spänningsreglering, på ett elegant sätt.

### KORTSLUTNINGSKONTROLL

I kretsschemat för regulatorn, visas i figur 1, är det lätt att identifiera skyddet för omvänt strömflöde. Om batteriets terminalsänkning är högre än den hos solpanelen så förhindrar Schottkydioden D3 att det flyter ström från batteriets positiva pol till den positiva terminalen på solpanelen, oberoende av tillståndet i resten av kretsen. I den omvända situationen har laddningsströmmen fritt fram till batteriet. Spänningsfallet över dioden är 0,43 V vid en ström på 3 A.

Strömmen från solpanelen kan emellertid också flyta genom D4 och T2 när transistor T2 är inkopplad. Transistorn drivs av operationsförstärkaren (opampen) IC1, som är kopplad som en komparator (jämförare). Transistor T1 och potentiometer P1 ger en referensspänning, som filtreras av kondensator C1. Denna spänning är inställd på ungefär halva polladdningsspänningen hos batteriet. Opampen jämför referensspänningen med spänningen i



**Figur 1. Kretsschema över laddningsregulatorn med backströmsskydd för solpaneler.**

punkten R1-R2, som är halva batterispänningen minus spänningsfallet på 0,6 V över dioden D2. De exakta värdena visas i kretsschemat. Om batterispänningen är lägre än terminal-laddningsspänningen förblir utgången från opampen låg och transistor T2 är bortkopplad. LED-en D1 är således släckt, vilket indikerar att hela strömmen från solpanelen flyter in i batteriet.

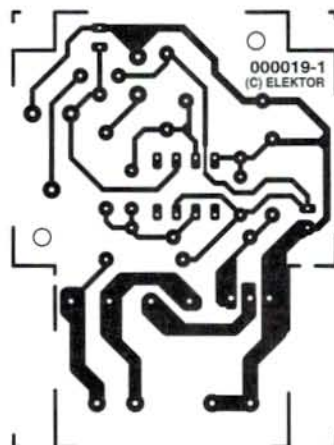
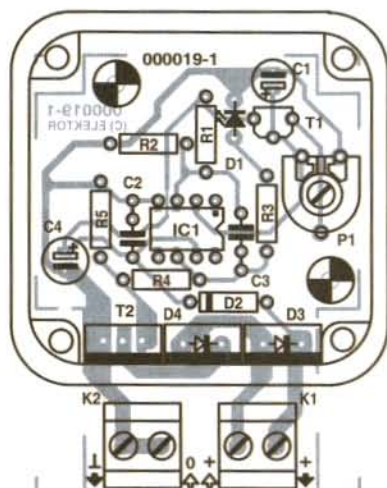
Om batterispänningen stiger ovanför terminal-laddningsspänningen blir komparatorns utgång hög (D1 till) och switchar till T2 så att utgången från solpanelen kortsluts. Eftersom en solpanel representerar en strömkälla, som kan leverera en-

dest en begränsad ström även när den är starkt belyst av solen, så är denna något brutala form av shuntreglering fullt acceptabel.

Trots detta så behövs det ytterligare någon form av skydd för att inte förlusterna skall bli för stora i T2 och D4 (vilket skulle kräva ett stort kyl-element). Detta ges av kondensator C4, som lämnar en kort positiv återkopplingspuls (som varar ungefär 4 ms) till opampen varje gång denna ändrar tillstånd. Detta förbättrar opampens switchningsegenskaper betydligt och flankerna på utgångssignalen blir betydligt brantare.

Effektörlusterna i en n-kanals MOSFET (som en BUZ100) är

**Figur 2. Kretskortet passar exakt i den rekommenderade lådan.**



som lägst när den antingen är fullt till eller full från, med andra ord, när den skickar en hög ström eller ingen ström alls. I den 'analoga' regionen mellan dessa två ytterlägen är effektörlusterna betydligt högre. Drivsignalens flanker skall därför vara så korta (branta) som möjligt. Detta är precis vad kondensatorn ser till.

När laddningsströmmen till batteriet faller från, på grund av att solpanelen kortsluts, så faller batterispänningen något. Detta får komparatorn att ändra tillstånd och batteriet börjar laddas igen. I praktiken betyder detta, att ju mer laddat batteriet är desto snabbare blinkar lysdioden.

## INBYGGNAD

Regulatorn för solpanelen kan byggas i en lämplig låda. Den lilla aluminiumlåda som specificeras i komponentlistan är speciellt lämplig eftersom den inte bara skyddar elektroniken mot yttre påverkan utan också fungerar som kyl-element för de tre effektt transistorerna. Det är endast en liten 'tunga' på kretskortet, där kopplingsplintarna sitter, som sticker ut från lådan.

Börja med att montera samtliga komponenter, utom de två kopplingsplintarna, på det kort som visas i figur 2. Tryck in de två Schottkydioderna och effektt transistorerna så långt i kortet som du kan. Därefter är du redo för en första test. Vrid P1 helt till vänster och anslut sedan ett justerbart nätaggregat (inställt på 13,8 V) till batteripolerna. Vrid därefter P1 sakta förbi mittläget.

## KOMPONENTLISTA

### Motstånd:

R1, R2 = 150kΩ  
R3, R4 = 4kΩ  
R5 = 100kΩ  
P1 = trimpot 500kΩ

### Kondensatorer:

C1 = 2μF 25V radial  
C2 = 100nF  
C3 = 10nF  
C4 = 100μF 25V radial

### Halvledare:

D1 = LED, röd, dagsljus  
D2 = 1N4148  
D3, D4 = PBYR745 (Philips)  
T1 = BF256B  
T2 = BUZ100\* (Siemens)  
IC1 = TL071CP eller TL081

### Övrigt:

K1, K2 = 2-pol kopplingsplint för kretskort, raster 5 mm  
Låda: Hammond 1590LB (50x50x32 mm)  
Kretskort 000019-1, se Läsaerservice på annan plats i tidningen

Om LED-en tänds fungerar kretsen som den skall.

Om LED-en inte tänds skall du kolla de två ingångsspänningarna till operampen och testa att komparatorn fungerar korrekt med båda ingångskonditionerna.

BUZ100 är förresten endast en förbättrad version av den gamla välkända BUZ10 som teoretiskt kan klara förluster upp till 250 W. Eftersom detta inte är fallet i denna krets så kan du ersätta BUZ100 med en BUZ10 om den försträmnda skulle vara svårt tag få tag på. Samma gäller Schottkydioderna. Vilken typ som helst som kan hantera minst 3 A kan användas.

## BORRA OCH FILA

Du måste göra en liten öppning i aluminiumlådan genom vilken kretskortets tunga kan sticka ut. Denna öppning skall vara 32 mm bred och 6 mm hög. Kretskortets layout är sådan att de tre effekthalvledarna ligger rätt emot lådans innervägg. Markera och borra de två hålen för att fästa kretskortet och montera sedan kortet i lådan på 5 mm höga distanser. Mät och markera sedan noga monteringshålerna för de tre halvledarna ovanför öppningen i lådan. Ta bort kortet, borra 2,5 mm hål och gänga dem för M3.

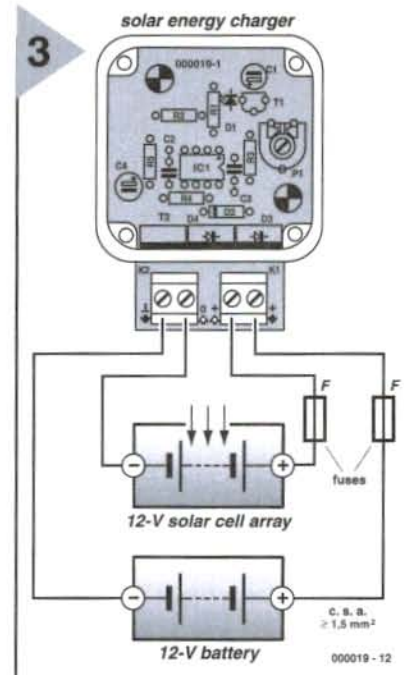
De två Schottkydioderna och MOSFET-transistorn måste isoleras från varandra och från lådan. Du kan använda mica eller plastbrickor som du får skära till så att de passar bredvid varandra och så att du kan stänga lådans lock.

Sätt nu tillbaka kretskortet igen och kolla att hålen i de tre effekthalvledarnas kyltungor passar exakt med de borrade hålen i lådan. Om du inte gör det måste du antingen fila ner distanserna något eller höja dem med hjälp av mellanlägsbrickor. När alla fem hålen passar exakt kan du dra åt alla skruvar utan att det blir någon mekanisk påfrestning. Glöm inte att använda kylpasta på båda sidor av isoleringsbrickorna och se till att det finns en isoleringsbussning (med krage) mellan skruven och kyltungan på varje halvledare. Kontrollera med en ohmmeter att det inte finns någon kortslutning! När du har kollat att allt är korrekt isolerat kan du montera kopplingsplintarna på kortets tunga. Innan du stänger lådans lock skall du justera P1 medan regulatort är ansluten till en riktig solkraftsinstallation (se figur 3), med regulatort placerad så nära batteriet som möjligt. Detta är nödvändigt för att ta spänningsförluster i kontakter och kablar med i beräkningen. Dessa är betydande även med relativt kraftiga kablar (1,5 mm eller mer), och för att uppnå optimala prestanda hos installationen. Glöm inte att lägga in lämpliga säkringar för båda kretsarna (se figur 3) eftersom det kan

**Figur 3. Så här integreras regulatort i en solkraftsinstallation.**

flyta en ström på upp till 100 A i händelse av en kortslutning!

(000019-1)



## Dimensionering av en självförsörjande solkraftsinstallation

De individuella komponenterna i en solkraftsinstallation måste vara korrekt dimensionerade för att den tillgängliga energin skall kunna utnyttjas optimalt.

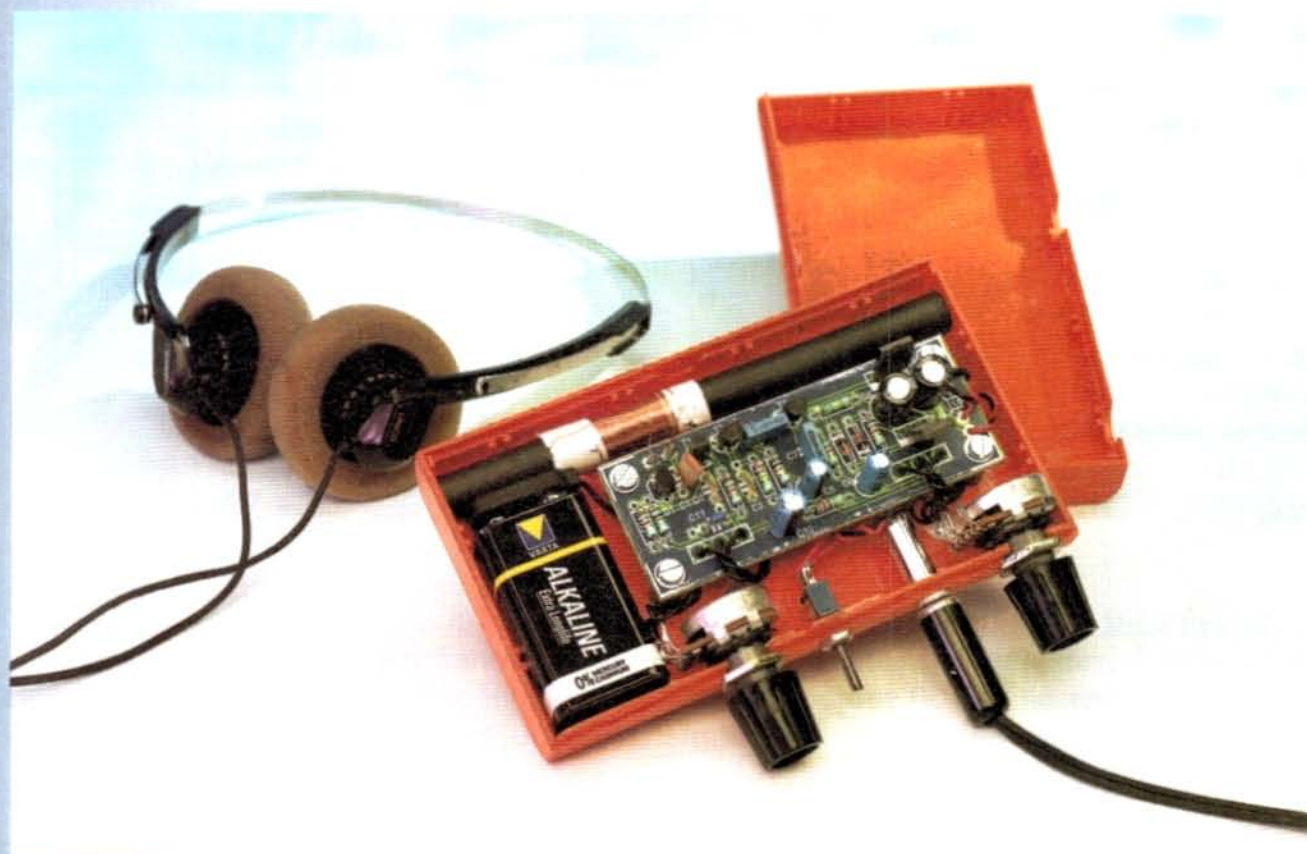
**Strömförbrukningen** (inklusive förlusterna i solregulatort), tillsammans med hur strömförbrukningen sker tidsmässigt, bestämmer i huvudsak installationens plandering. Baserat på denna information kan du beräkna den genomsnittliga strömförbrukningen per dag eller per vecka. I en mindre solkraftsinstallation, med en genomsnittlig förbrukning på som mest 3 kWh/dag, och en toppbelastning på ca 3 kW, kan du vanligtvis ha en systemspänning på 12 V. Med högre laster rekommenderas en högre systemspänning för att hålla kabeltjockleken och kabelförlusterna inom rimliga gränser. En spänning på 24 V-48 V, eller till och med upp till 230 V, kan erhållas genom att seriekoppla flera solpaneler. Nästa steg är att bedöma **storleken på solgeneratort**. Placeringen av solpanelerna och deras vinkel i förhållande till horisonten, solpanelernas verkningsgrad och naturligtvis den geografiska lokaliseringen av installationen (eftersom solen lyser starkare och oftare på vissa platser) ingår alla i denna beräkning. Resultatet blir antingen den behövliga nominella strömkapaciteten eller, i fallet med en viss given panel, antalet paneler (parallellkopplade).

Slutligen måste du bestämma **kapaciteten hos lagringsbatteriet**, vilket kompenserar för klimatvariationerna och således den tillgängliga solenergin. Det finns två kriterier här, antingen den redan nämnda dagliga eller veckovisa strömförbrukningen i ampéretimmar, och antal dagar som installationen skall kunna köras från enbart batteriet, vilket i stort sett är beroende på individuella applikationer. Här måste den tillåtna urladdningskapaciteten hos batteriet tas med i beräkningen.

Förutom dessa huvudkrav finns den en hel del andra saker som man också måste tänka på, maximalt tillfälligt strömuttag (toppar), spänningsreserv vid höga temperaturer, den mekaniska prepareringen av modulerna, finns det tillräckligt med plats, kostnader, bygglov, eventuella bidrag osv.

Om du är intresserad av att bygga en solkraftsanläggning så rekommenderar vi dig att först läsa någon av den litteratur som finns om detta. Det finns också ett antal mer eller mindre användbara PC-program som hjälper dig att beräkna kraven på komponenterna som skall ingå i installationen. En bra introduktion hittar du på CD-skivorna Solar Power och Solar Craft ([www.solarenergie.com](http://www.solarenergie.com)). På dessa finns introduktion, basinformation, adressregister, tips och konstruktionsinformation. Solar Power inkluderar ett enkelt beräkningsprogram, medan Solar Craft har ett interaktivt beräkningsprogram som steg för steg leder dig genom konstruktionen av en solkraftsanläggning för hushållsbruk.

# Mellanvågsmottagare i midgetformat



Den lilla mellanvågsmottagare (MV) som beskrivs i denna artikel är ett idealiskt byggprojekt att sysselsätta sig med en regnig dag. Löd fast några standardkomponenter på ett kretskort, anslut det färdigbyggda kortet till ett batteri och, färdigt!

Nuförtiden är det få byggprojekt, eller kommersiella apparater för den delen, som enbart använder diskreta komponenter. De flesta apparater som man stöter på har antingen minst en processor och/eller ett antal IC-kretsar. Även om dessa apparater har stora fördelar när det gäller pålitlighet och enkelhet, så finns det dock en hel del läsare som vi tror välkomnar ett projekt där man slipper ge sig av ut och leta efter specialkretsar.

Den här midgetradion består av fem vanliga transistorer, en spole och en handfull passiva komponenter av standardtyp. Flera av er har säkert dessa komponenter redan liggande i 'bra-att-ha' lådan.

För att förekomma frågor om det verkligen räcker med fem transistorer för att bygga en mottagare så kan vi säga att endast två av dessa används i själva

Design av G. Baars

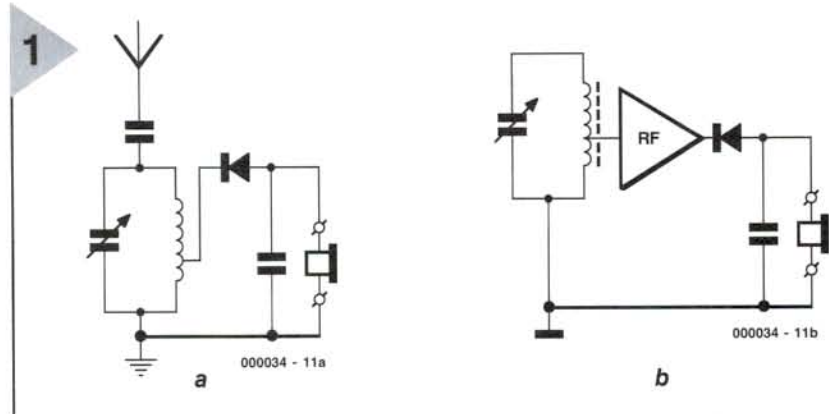
radiostegen. De övriga tre används för den inbyggda utgångsförstärkaren. Självklart kan man inte förvänta sig mirakel från två transistorer. Trots detta behövs det inte någon jätteantenn för att få den lilla signalen hörbar. Midgetmottagaren är fullt kapabel att ta emot lokala sändare utan någon yttre antenn. Avstämningsspolen som är lindad på en ferritstav kan plocka upp dessa rundradiostationer utan någon extern hjälp.

## DIREKTOMVANDLANDE MOTTAGARE

Designen är den hos en direktomvandlande mottagare. De flesta läsare vet att det i grunden finns två typer av mottagardesign, superheterodyn och direktomvandlande mottagare (också kallad TRF, Tuned Radio Frequency). Den superheterodyna mottagaren är den i särklass mest populära av de två. Det är en design i vilken ett antal avstämda kretsar används för att öka känsligheten och den mottagna signalen blandas med signalen hos en avstämd oscillator. Ingångskretsen för HF och oscillatorkretsen avstäms synkront för att hålla skillnaden mellan frekvenserna i de två kretsarna konstant. Denna skillnadsfrekvens kallas för mellanfrekvens, MF.

Det faktum att utgången från blandaren, dvs MF, har ett konstant värde har den fördelen, att de steg som ligger efter blandaren kan filtreras, utan att de relevanta LC-kretsarna (spole/kondensator) behöver stämmas av för alla olika signaler som tas emot.

Det här resonemanget visar tydligt att detta inte är någon lösning för en liten och enkel mottagare. En direktomvandlande design är dock perfekt för detta. I denna typ av design väljs den önskade signalen ut, förstärks ofta och demoduleras sedan. Detta betyder att det inte finns något behov av en



**Figur 1. En enkel diodmottagare (1a) kan förbättras betydligt om man lägger till en HF-förstärkare (1b).**

oscillator eller någon omvandling till en mellanfrekvens.

Figur 1 visar kretsschemat för en TRF-mottagare i sin enklaste form. I denna mottagare väljs den önskade frekvensen med en avstämd LC-krets från det spektrum med frekvenser som plockas upp av antennen. Den valda signalen demoduleras av dioden och den resulterande audiosignalen (LF) matas till ett par hörlurar.

Den enkla mottagaren i figur 1a behöver en ganska stor antenn. Notera att den avstämda LC-kretsen är ansluten mellan antennen och jord. Så för att inte belasta kretsen för mycket (vilket skulle påverka kretsens känslighet ofördelaktigt) är detektorn normalt ansluten till en avtappning på spolen.

En stor fördel med denna enkla mottagare är att den inte behöver någon matningsspänning. Å andra sidan är den inte speciellt känslig och dess prestanda beror i hög grad på antennen. Dessa nackdelar kan man i hög grad undvika genom att lägga till en förstärkare. Om man inte gör detta

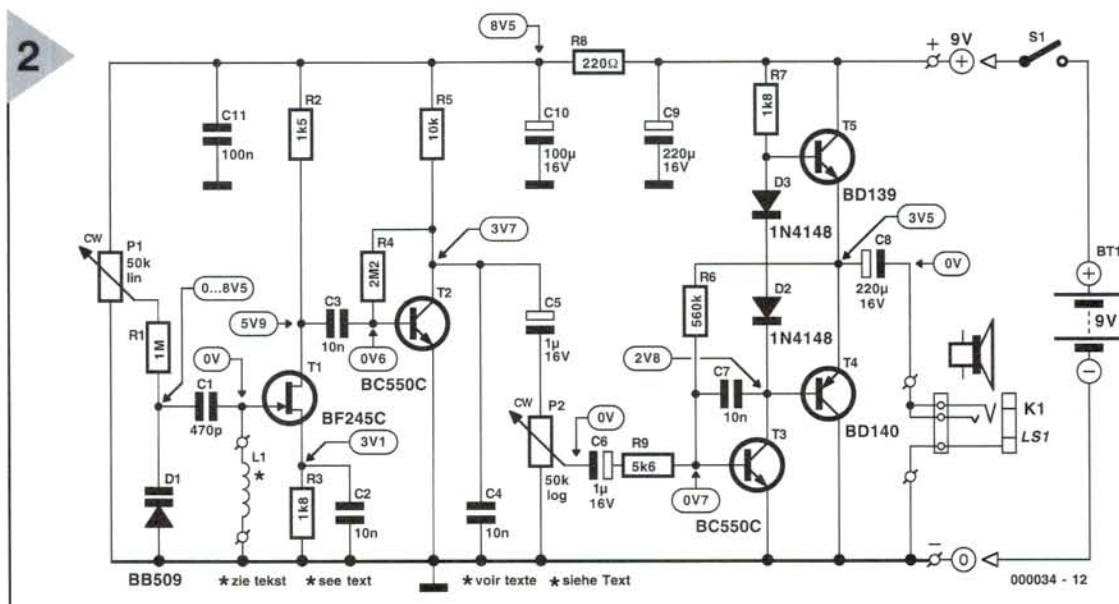
måste nivån hos den inkommande, önskade, signalen överstiga diodens framspänning, dvs den spänning som dioden måste ha för att kunna leda (kallas också för bias). När det gäller den germaniumdiod som används här är framspänningen 100-200 mV.

Adderingen av en HF-förstärkare visas i figur 1b. En sådan mottagare kan ha en känslighet som gör det möjligt att ta emot sändningar med hjälp av en spole som är lindad på en ferritstav.

Helt klart kan en direktomvandlande mottagare inte jämföras med en superheterodyn när det gäller känslighet och selektivitet. Trots allt så har mottagaren i figur 1b bara en avstämd krets och en HF-förstärkare. Inte desto mindre har den vissa fördelar framför en superheterodyn. En TRF-mottagare är till exempel enkel att bygga, billig att tillverka, inte kritisk att koppla upp, behöver inte mycket avstämning, är fri från störande visslingar och ger ett ljud med bra kvalitet.

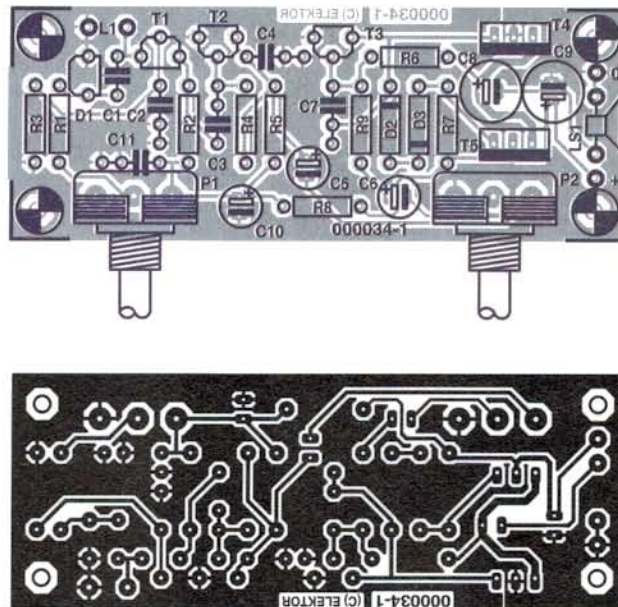
## KRETSBESKRIVNING

Kretsschemat för midgetmottagaren



**Figur 2. Kretsschemat över midgetmottagaren, som baseras på endast fem transistorer.**

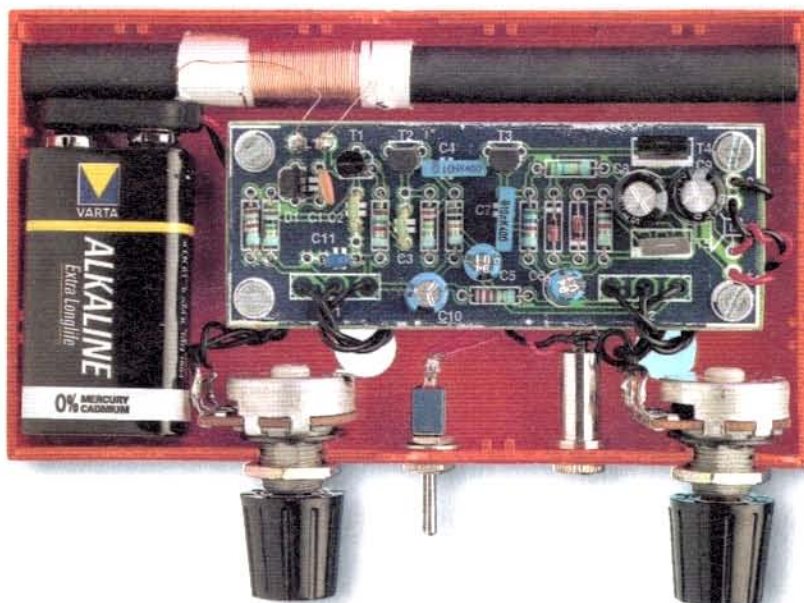
3



**Figur 3. Kretskortslayouten för mellanvågsmottagaren i midgetformat.**

visas i figur 2. Själva mottagaren, dvs HF-sektionen, finns till vänster om potentiometer P2. Även om den fortfarande är liten så har den dock en hel del fler komponenter än mottagarna i figurerna 1a och 1b. Den avstämda HF-kretsen består av spolen L1, kondensator C1 samt dioden D1. För att minska kostnaderna är C1 inte en variabel kondensator utan en kapacitansdiod (varicap). Kapacitansen hos en sådan diod varierar med den spänning som läggs över dioden.

Denna spänning varierar med P1 så att denna potentiometer kan användas för att stämma av mottagaren. Notera att spolen L1 som lindats på en ferritstav inte har några avtappningar som i figurerna 1a och 1b. Om man, som vi nämnde tidigare, ansluter demodulatorn eller HF-förstärkaren till en sådan avtappning minskar belastningen på spolen och detta förhindrar att Q-faktorn, och därmed kretsens selektivitet, försämras. Detta är speciellt viktigt i en mottagare som endast



**Figur 4. Prototypen passar exakt i lådan med de angivna dimensionerna.**

#### KOMPONENTLISTA

##### Motstånd:

R1 = 1M $\Omega$   
 R2 = 1k $\Omega$ 5  
 R3, R7 = 1k $\Omega$ 8  
 R4 = 2M $\Omega$ 2  
 R5 = 10k $\Omega$   
 R6 = 560k $\Omega$   
 R8 = 220 $\Omega$   
 R9 = 5k $\Omega$ 6  
 P1 = 50k $\Omega$  lin potentiometer  
 P2 = 50k $\Omega$  log potentiometer

##### Kondensatorer:

C1 = 470pF  
 C2, C3, C4, C7 = 10nF  
 C5, C6 = 1 $\mu$ F 16V radial  
 C8, C9 = 220 $\mu$ F 16V radial  
 C10 = 100 $\mu$ F 63V radial  
 C11 = 100nF

##### Spole:

L1 = 50 varv 0.3 mm lackerad koppartråd på ferritstav (10x100mm), se texten

##### Halvledare:

D1 = BB509  
 D2, D3 = 1N4148  
 T1 = BF245C eller BF256C  
 T2, T3 = BC550C  
 T4 = BD140  
 T5 = BD139

##### Övrigt:

K1 = 3.5 mm telejack  
 S1 = till/frånomkopplare  
 B1 = 9-V batteri med clips  
 Ls1 = örsnacka, hörlurar eller 8 $\Omega$  högtalare  
 Låda (ungefär 123x30x70mm)  
 Kretskort 000034-1  
 (Se Läsaerservice på annan plats i tidningen)

har en avstämd krets (som bestämmer mottagarens totala selektivitet). Det finns emellertid en nackdel med att använda en avtappning och det är att den signal som läggs till HF-förstärkaren eller demodulatorn inte är den maximalt tillgängliga. I midgetförstärkaren tas signalen över spolen men, eftersom förstärkaren T1 är en fälteffekttransistor, FET, som har en mycket hög ingångsimpedans, så belastas spolen endast lätt. Även om T1 ger en viss förstärkning så ges den huvudsakliga förstärkningen av transistor T2. Denna transistor fungerar också som demodulator, vilket är möjligt eftersom dess bas-emitterkoppling är en p-n diod. Restkomponenterna i bärvågsfrekvensen (signalen) kortsluts till jord av kondensator C4. Den demodulerade signalen läggs till utgångsförstärkaren

T3-T5 via kondensator C5 och volymkontrollen P2.

Det skulle vara möjligt att använda en IC-krets för utgångsförstärkaren men vi valde transistorer för att hålla hela konstruktionen i linje med HF-delen. Transistorerna ökar inte storleken på bygget. Transistor T3 fungerar som en spänningsförstärkare och drivare medan T4 och T5 bildar ett enkelt motakts utgångssteg (push-pull).

Dioderna D2 och D3 kompenserar för framspänningen hos T4 och T5 så att överhörningsdistorsionen är absolut minimum. I viloläge drar utgångssteget bara några få milliamp.

Utgångssteget klarar inte bara av att driva ett par hörlurar (öronsäckarna parallellkopplade) utan också en liten högtalare på 8 ohm. Trots att den maximala utgångseffekten bara är 1 W så räcker detta för att fylla en genomsnittligt vardagsrum med ljud.

## STRÖMFÖRSÖRJNING

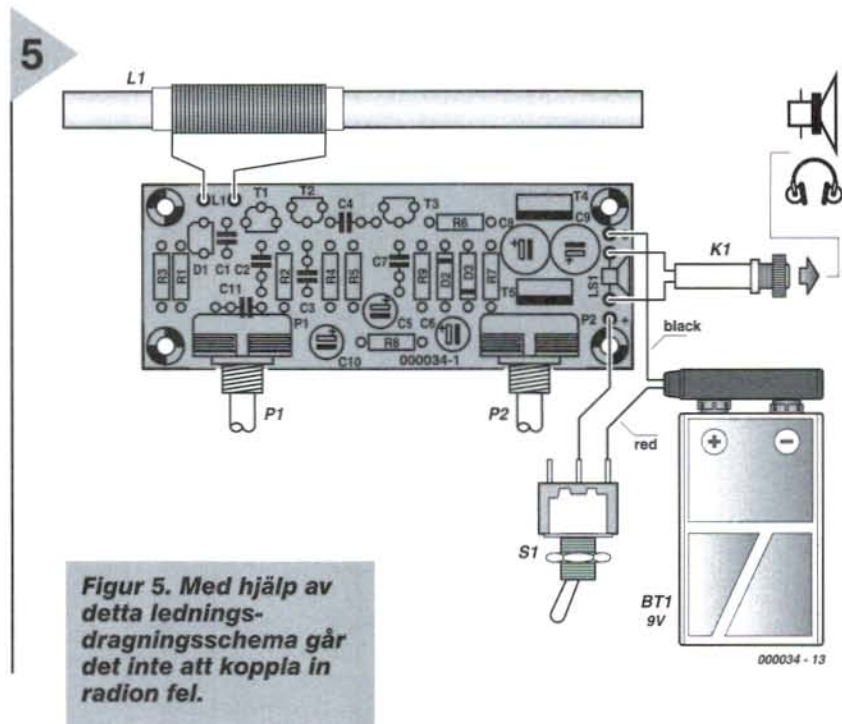
Mottagaren behöver en spänningsmatning på 9 V och eftersom strömförbrukningen under normala förhållande inte överskrider 30 mA så räcker det med ett batteri. Vid normal användning kommer detta att räcka länge.

Om du använder en nätdapter som strömkälla måste denna vara stabiliserad eftersom spänningen för kapacitansdioden D1 tas direkt från denna. Den absoluta nivån på matningsspänningen är inte särskilt kritisk. Prototypen fungerade bra med en spänning på 7 V, medan 12 V förbättrade känsligheten i viss mån.

## SJÄLVA BYGGET

Det är bäst att bygga midgetmottagaren på det kretskort som visas i figur 3. Kortet är en kompromiss mellan små dimensioner och hanterbarhet. Det är således mycket lätt att bestycka kortet. Börja med att montera de passiva komponenterna och sedan transistorerna. Kolla komponentlistan och komponentplaceringsritningen. Den enda komponent som inte skall lödas på kortet är spolen L1. Denna monteras utefter kortet med de två terminalerna kopplade till markeringarna L1 på kortet.

Spolen L1 måste du tillverka själv, men detta är inte svårt. Den är lindad runt en 10-12 cm lång ferritstav med en diameter på 10 mm. Tillverka en cylinder av tunn plast eller styvt papper med en inre diameter som gör att du kan skjuta den fram och tillbaka över ferritstaven. Tätlinda 50-55 varv och 0,3 mm lackerad koppartråd på cylindern. Detta skall du vara mycket noga med eftersom spolens kvalitet kommer att bestämma mottagarens prestanda. De utstickande trådarna (terminalerna) skall fästas med tape eller genom att du för dem genom små hål som borrats i cylindern.



Spolen är den enda del i mottagaren som behöver kalibreras, i detta fall till en rundradiostation på mellanvågsbandet. Beroende på konstruktionen kan spolens självinduktans variera från fall till fall. Denna variation kan du kompensera genom att föra spolen över ferritstaven. När den är i mitten är självinduktansen maximal medan den vid ändarna minskar något. Du behöver inga mätinstrument vid kalibreringen. Använd en annan mellanvågsmottagare som referens och flytta spolen över ferritstaven tills de två MV-banden är mer eller mindre identiska. MV-området sträcker sig ungefär från 530 kHz till 1605 kHz.

Om du inte kan få in området genom att flytta på spolen måste du anpassa antalet varv. Om området inte går ner till 530 kHz måste du lägga till ett par varv och om det inte går upp till 1605 kHz måste du ta bort ett par varv. Eftersom det är enklare att ta bort varv än att lägga till nya rekommenderar vi att du börjar med 55 varv.

## LÅDA

När kortet är färdigbyggt, och förstärkaren fungerar tillfredsställande, skall kortet monteras i en lämplig låda. Denna skall vara av plast (ABS) eller trä eftersom metall skärmar av ferritantennen. Idealiskt skall lådan vara ca 125x30x70 mm och borde inte vara svår att få tag på i en hobbyaffär eller elektronikaffär. Prototypen monterades i en låda med nästan exakta dimensioner (se figur 4). Tyvärr var vi tvungna att montera potentiometerna på lådan i stället för på kortet. Om ferritantennen inte kan klämmas fast i lådan så kan du använda skumplast eller liknande för att få fast den.

Om du vill använda hörlurar eller en högtalare bestämmer du själv. Prototypen används med hörlurar eftersom lådan var för liten för även den minsta högtalare. Vill du ha högtalare så är de som finns att köpa för bärbara mottagare (Walkman) idealiska. Dessa är inte speciellt dyra och ger ett förvånansvärt bra ljud.

## UPPKOPPLING

Även om de olika terminalerna är markerade tydligt på kretskortet så visar figur 5 för säkerhets skull exakt hur du skall ansluta batteriet, högtalare (om du använder en sådan) och ferritantennen till kortet. Detta som en liten hjälp för våra nybörjare inom elektronik.

[000034-1]

# BASIC Stamp

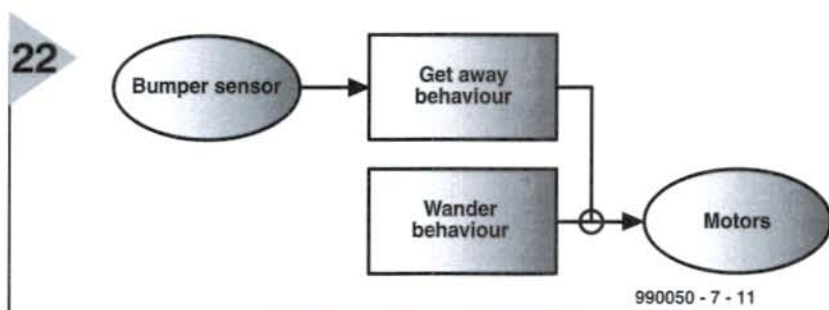
## programmeringskurs

### Del 7: subsumptionprogrammering

Av Dennis Clark

#### SUBSUMPTION- ARKITEKTUR OCH ROBOTBETEENDE

Att inordna en uppgift är att ersätta den med en annan uppgift som har en högre prioritet. När vi pratar om subsumption i samband med robotprogrammering så beskriver vi den process där ett beteende inordnas (eller underordnas) ett annat beteende baserat på en tydlig prioritet som vi har definierat. Denna beteendearkitektur beskrevs först av Dr. Rodney Brooks i artikeln "A robust layered control system for a mobile robot" in *IEEE Journal of Robotics and Automation*, RA-2, April, 14-23, 1986. I den *Brooksianska* ideala subsumptionarkitekturen behöver ett beteende inte veta någonting om något annat beteende. En robot som programmerats på detta sätt reagerar reflexivt till sin omgivning med hjälp av enkla beteenden. Vad detta också betyder är att nya beteenden kan adderas till en robot utan att behöva ändra något annat beteende. Därmed blir det mycket enkelt att förbättra en robots programmering. Ett exempel på subsumptionsprogrammering i vår lilla X-Bot vore att ha ett beteende som helt enkelt slumpvis väljer en riktning att färdas i och en viss tid att färdas i denna riktning. När detta utförts skall subrutinen (nu ett beteende) välja en ny riktning och en ny tid. Men vi vill inte fastna mot en vägg eller ett bordsbän så vi lägger till ett beteende som kollar en stötfångare och, om vi stöter



Figur 22. Enkel visning av procedurer baserade på subsumption.

990050 - 7 - 11

på någonting, får vår robot att backa och svänga. Vi vill också

att stötbeteendet (bump) skall ha prioritet över körbeteendet (wander) så att det inordnar detta beteende och tar över kontrollen av hjulmotorerna för att komma bort från hindret. Eftersom stötbeteendet har en högre prioritet än körbeteendet så kan vi vara säkra på att vår lilla X-Bot kan ta sig förbi varje hinder som blockerar dess väg. Dessutom, om körbeteendet har satts till en mycket lång tid vid den senaste riktningförändringen så kommer körbeteendet att ta över kontrollen igen när stötbeteendet är färdigt med sin uppgift att få bort roboten från hindret och den kommer att fortsätta köra som om ingenting hänt.

Eftersom varje beteende är självständigt, och flera av dem helt enkelt reagerar på robotens omgivning, kan vi skapa en uppsättning beteenden vars samverkan med varandra inte är programmerade och vi kommer att få se kombinationer som vi inte förutsett. Detta kallas för *framväxande beteende* (emergent) eftersom vi inte planerat för det utan det uppstår som ett resultat av robotens samverkan med sin omgivning. När man tillåter framväxande beteenden verkar det som om roboten handlar helt på egen hand och inte bara gör samma sak om och om igen. Vad som är viktigare i den *verkliga* världen är att vissa beteenden kan programmeras att utföra en verklig upp-

gift, hämta ett föremål till exempel, och inte bry sig om rutinarbetet

med att undvika att köra ner i ett hål eller fastna mot en vägg.

De enkla beteenden som vi just diskuterat kan visas grafiskt med hjälp av ett diagram. Ett exempel på vårt enkla kör/stötbeteende visas i figur 22. Ovalen till vänster är ingång, rektanglarna i mitten är beteenden och ovalen till höger är utgång.

När en linje från ett beteende med en högre prioritet skär en linje från ett beteende med lägre prioritet på väg till utgångsaktivatorn (i detta fall Motors) kan man säga att den högre prioriteten har inordnat den lägre prioriteten och tagit kontroll över denna. Detta visas i diagrammet ovan där linjen *Get away* (gå bort från hindret) skär linjen *Wander* (kör framåt) med en cirkel i skärningspunkten. Det kan finnas flera olika skärningspunkter på flera olika ställen i riktninglinjerna som kan visa subsumptionslogiken mycket tydligt. Försök att själv rita upp ett nätverk för din egen robot. Det finns många sätt att bygga upp ett diagram över ett subsumptionsnätverk.

#### HUR GÅR VI VIDARE HÄRIFRÅN?

Lär dig diagrammen och tankesättet. Framöver kommer vi att använda dessa idéer och bygga ut de enkla diagram vi hittills visat. I varje sektion kommer vi att dela upp arbetet i bitar som är enkla

## Lista 10.

```
'Servo routine cons and vars
LEFT con 15 'port 15, left motor
RIGHT con 3 'port 3, right motor
SACT con 5 'times through the
loop
drive var word 'both sides in var
ldrive var drive.byte1 'left side is here
rdrive var drive.byte0 'right side here

aDur var byte 'duration counter
```

```
act:'servo controller subroutine
  if aDur > 0 then aDec
    aDur = SACT 'do state 1
    pulsout LEFT,lmotor * 10
    pulsout RIGHT,rmotor * 10
    goto aDone 'state 1 done
aDec:
  aDur = aDur - 1 'do state 2
aDone:
return
```

att följa, såsom de variabler som behövs, I/O-portar som används och nya instruktioner. Vi kommer också att visa den process med vilken statusmaskiner definieras och hur subsumptionsdiagram kan användas för att förstå våra beteendeprioriteringar.

### FAS 1: ATT FÅ X-BOT ATT RÖRA SIG OCH FINITE STATE MACHINE

En robot som inte gör någonting är inte speciellt intressant. När du nu har modifierat dina servon, anslutit dem till I/O-portarna på Stamp II och kontrollerat att allting fungerar, skall vi få vår robot att röra sig slumpvis i sin miljö. För att implementera denna enkla funktion behöver vi två moduler, en som bestämmer en riktning och en tid att gå i denna riktning, samt en annan som sänder kommandon till hjulen.

Vi kommer att kalla dessa moduler *wander* (kör) respektive *act* (agera). *Act* påverkar helt enkelt motorerna. Detta är egentligen inte ett beteende, det är en utgång, och i vårt schema får det beteckningen *motors*. *Wander* är ett beteende och visas också i vårt subsumptionsdiagram som ett beteende.

#### State Machine och Stamp II kod för agera

Vi vet att för att få våra servon att vrida hjulen behöver vi skicka ut en puls vars bredd är ungefär 1 ms till 2 ms lång och där 1,5 ms är stoppläget. Vi vet också att denna puls måste upprepas var 20 ms-30 ms för att våra servon skall fortsätta att röra sig. Detta innebär att vi behöver programmera två operationer. En kommer att vara 'skicka ut pulsen' och den andra 'vänta 20 ms', varefter vi går tillbaka till status 1 (tillstånd 1). Vi har två servon och kan därför säga att vi har tre operationer, skicka vänster servopuls, skicka höger servopuls samt vänta. För att göra denna modul enkel kommer vi att begränsa den till två tillstånd, ett som skickar ut pulserna och ett för fördröjningen innan nästa puls skickas ut. Eftersom vi vet att vi måste upprepa pulsen var 20-30 ms och vi vet att vi inte vill spendera all vår tid i *act*-

modulen i väntan på att denna tid skall förflyta, så vet vi också att vi kommer att gå in och ur detta kodblock flera gånger innan en full uppsättning statusöverföringar har skett. Vi vet att varje instruktion tar ca 250  $\mu$ s att utföra i Stamp II och detta talar om för oss hur länge som varje modul vi anropar kommer att köras, vi räknar antalet instruktioner och multiplicerar med 250  $\mu$ s. Vi gör en gissning och antar att det tar 5 passeringar genom tillstånd 2 för att 20 ms skall förflyta. När vi skriver andra moduler kommer vi att revidera denna uppskattning, men för ögonblicket är det en bra startpunkt. Låt oss nu rita upp vår statusmaskin så att vi förstår vad det är vi vill och kanske upptäcka fel i logiken.

Innan vi kan bygga vår statusmaskin är det bra att definiera alla de aktioner som vi vill att vår beteende eller utgångsfunktion skall utföra. Var och en av dessa funktioner kan grupperas efter sina aktiviteter till en individuell status. Här är ett sätt att definiera en motorstyrningsfunktion, vilket vi såg ovan som subrutinen *act*:

```
State 0
  Output left servo pulse value
  Output right servo pulse value
  Set the number of iterations
  (aDur) = 5
State 1
  Decrement aDur
  If aDur = 0 then go to state 0
```

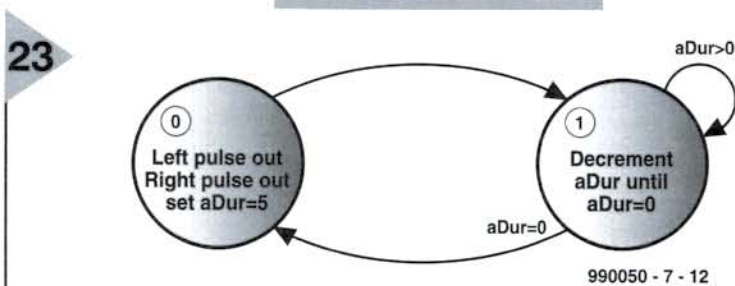
Vi har nu en detaljerad lista av aktioner som måste ske i vår statusmaskin för *act*. Vi kan nu rita ett statusdiagram med de önskade detaljerna, som i figur 23.

Detta statusdiagram talar om för oss när varje övergång sker och exakt vad varje tillstånd gör. Övergången från tillstånd 0 (till vänster) och tillstånd 1 sker alltid, det finns inget villkor. Övergången från tillstånd 1 (till höger) till tillstånd 0 sker emellertid när  $aDur=0$ . Notera att tillstånd 1 har en slinga som går ur och i sig själv. Detta indikerar att tillståndet *upprepar* (itererar) sig själv och i vårt fall betyder det att vi går in i denna del av modulen flera gånger innan den är fullbordad. Titta noga på detta diagram. Det visar mycket tydligt vad vår *act*-modul måste göra och i vilken ordning den skall göra det. Tänk på hur det skulle se ut om vi skulle skicka ut den vänstra motorpulsen i ett annat tillstånd än den högra motorpulsen. Hur skulle detta se ut? För våra ändamål kan vi säga att varje cirkel (tillstånd) är en plats där vi kan lägga in modulen i vårt program och bestämma vad som skall göras sedan. Nu borde du kunna se hur kraftfullt detta koncept är för oss!

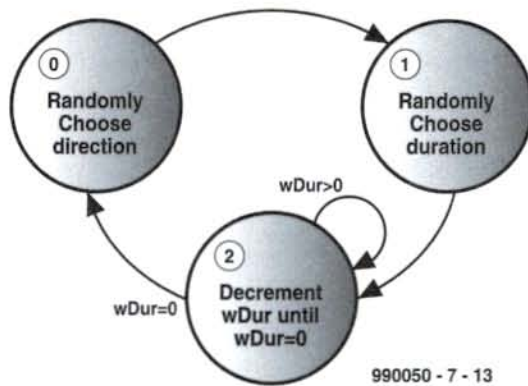
Den kod som implementerar vår *act*-modul visas i Lista 9. Vi kommer att visa variabeldeklarationerna till vänster och den egentliga koden till höger från och med nu. För att göra ett program lättare att läsa och modifiera så skall du använda *con*-uttryck flitigt och inte 'magiska tal' inbäddade i koden. Kom ihåg att allting som kommer efter en apostrof (') är en kommentar och inte en instruktion. Vi kommer att förklara varför vi multiplicerar *lmotor* och *rmotor* värdena med 10 i nästa fas.

Om vi skall skriva ett stort program måste vi spara variabelutrymme och PBASIC-språket har

Figur 23. Statusdiagram för proceduren 'act'.



990050 - 7 - 12



Figur 23. Statusdiagram för proceduren 'Wander'.

använder en variabel med word-storlek, eftersom vi bryter ner drive-variabeln i

en byte0 och en byte1 variabel kan vi få både vänstra och högra motorvärdena från en enda lookup! I lista 11 kan du se hur detta görs och du ser också hur detta gör det mycket enkelt för oss att ändra vad roboten skall göra. Till vänster är alla användbara motorstyrningskommandon listade med en \$ framför dem. Det betyder att dessa tal är hexadecimala och vi har använt HEX-tal eftersom det är enkelt att se vänstra och högra halvorna av dem (*byte1* och *byte0*). Det betyder naturligtvis att vi måste kunna HEX för att kunna fatta vad som sker. I HEX representerar varje tecken talen 0-15 och där A, B, C, D, E och F representerar talen 10, 11, 12, 13, 14, 15 respektive 16. Siffran längst till höger än entalssiffran precis som i vårt decimala talsystem. Den kan dock ha värdet 1 till 15. Nästa siffra åt vänster är '16-talet' och det tal som du ser här skall du därför multiplicera med 16 och sedan lägga till 'entalssiffran' för att få det decimala talet. Våra tal ovan är 4-siffriga, men kom ihåg att det är bara för att vi anger både vänstra och högra motorvärdena samtidigt, så vi kan betrakta de två vänstra siffrorna som separata tal i förhållande till de två siffrorna till höger. Detta betyder att HEX-talet '\$1C = 16+12 = 28. Som du kommer ihåg så nämnde vi tidigare att *act* multiplicerar *lmotor* och *rmotor* talet med 10. Nu är det dags för en förklaring. Ett byte kan endast hålla värden från 0 till 255 och våra servon behöver tal från 500 till 1000. För att få detta område använder vi 50-100 (vilket är mindre än 255) för varje motors hastighetsinställning och sedan multiplicerar vi detta med 10 för att få värden från 500 till 1000. Detta representerar inte alla tillgängliga värden men det spelar ingen roll när vi driver modifierade servomotorer som

några mycket intelligenta sätt att spara detta utrymme). Instruktionen *pulsout* skickar ut en puls till våra servomotorer med den bredd för hastighet och riktning som vi önskar. Variabeln *drive* är ett ord som är två byte, *byte0* och *byte1*. Du kommer att se varför detta är ett användbart sätt att representera motordrivningsvariablerna när vi designar koden för *wander*-modulen i nästa fas!

## FAS 2: ATT FÅ X-BOT ATT RÖRA SIG

Allt om vi vill att *wander* skall göra är att välja en riktning och en tid att röra sig i denna riktning. Denna modul är lika lätt att designa som *act*-modulen och demonstrerar hur enkelt det är att lägga till nya beteenden och hur dessa beteenden kan komma ihåg vad de håller på med. Det första steget när det gäller att designa vår finite state machine är att skriva ner alla steg som skall tas och alla de övergångsfunktioner som skall testas.

Här är vår aktionslista för *wander*:

```
State 0
  Choose a direction to go by
  using the random function (wDir)
  Go to State 1
State 1
```

```
  Choose a duration to go by
  using the random function (wDur)
  Add some minimum time to the
  duration so its not too short
  Go to state 2
State 2
  Decrement wDur
  If wDur = 0 then go to state 0
```

Diagrammet för vårt *wander*-beteende visas i figur 24.

Några av pilarna har inga övergångsetiketter angivna. När en statusmaskin automatiskt går över från ett tillstånd till ett annat, och inte behöver ta något beslut, så behövs ingen övergångsfunktion.

Ändamålet med *wander*-modulen är att slumpvis välja en riktning för vår robot att röra sig i och slumpmässigt välja en tid för hur länge roboten skall röra sig. Instruktionen *random()* spelar en viktig roll i *wander*-logiken. *Random* använder en *word*-variabel och behöver en 'seed' (ett frö) för att sätta upp sitt returvärde. Vi kommer att använda det senast returnerade värdet vi fick som 'seed' för nästa. Vi kommer också att använda en *mask* för att endast använda tal inom ett visst område. Ett litet område för riktningförändringarna och ett större område för *wander*-tiden. En annan intressant egenskap är lookup-instruktionen. Denna instruktion

### Lista 11

```
'Servo drive commands
fd      con      $6432    'forward
rv      con      $3264    'reverse
st      con      $4b4b    'stop
tr      con      $644b    'turn right
tl      con      $4b32    'turn left
rr      con      $6464    'rotate right
rl      con      $3232    'rotate left
'wander values
wstate  var  byt  e      'FSM status
wDir    var  word  'wander value
wDur    var  byte  'wander duration
```

```
wander:
branch wstate, [wDir, wDur]
'This is state 2
wDur = wDur - 1
```

```
if wDur > 0 then wDone1
  drive = wDir      'get direction
  wstate = 0       'reset state
wDone1: 'completed
  return
wDir:
  random seed      'choose direction
  random direction
  i = seed & %111  'mask off for 0-7
  lookup i, [tr, fd, fd, fd, rr, fd, fd, tl], wDir
  'choose direction
  wstate = 1       'next state
  return
wDur: 'choose duration
  random seed      'randomize
  wDur = (seed & %111111) + 20
  'mask for 64 and add 20 for more time
  wstate = 2       'next state
  return
```

hjul. Vi har nu nästan allting vi behöver för att få vår robot att utan mål vandra omkring i ett rum. Nästan. Våra *wander* och *act* funktioner är subrutiner och detta betyder att någonting måste anropa dem. Den loop som definieras som *main* och *goto main* är 'hjärnan' för vår robot. Alla beteenden anropas därifrån. Lista 12 är det kompletta programmet som har alla variabler och subrutiner såväl som *setup* och

huvudslingan och som visar hur allt passar tillsammans. Knappa in detta i din Stamp II programmerare och kolla hur din robot vandrar omkring i rummet. Se till att den inte kör in i någonting! Nästa månad kommer vi att visa hur du undviker att köra in i en vägg. Notera sektionen *set up for running* i koden. Vi måste se till att *wander*-beteendet startar med rätt tillstånd och vi sätter tillståndet här.

Nu måste vi se till att *wander* och *act*

anropas regelbundet och dessa två subrutinsanrop placeras i loopen som börjar vid *main*. Denna loop kommer att vara en mycket viktig del av vår kod när vi börjar addera fler beteenden till vår robot.

(990050-7)

## Lista 12

```
'Generic values
I      var      byte      'loop counter, whatever
Tmp    var      word      'temporary holder
Seed   var      word      'random number seed

'These are for the servo routines
LEFT   con      15       'left wheel port
RIGHT  con      3        'right wheel port
SACT   con      5        'times through act routine
Drive  var      word      'wheel command combo
Ldrive var      drive.byte1 'left wheel command
Rdrive var      drive.byte0 'right wheel command
ADur   var      byte      'duration of pulse left

'Servo drive commands
fd     con      $6432    'forward
rv     con      $3264    'reverse
st     con      $4b4b    'stop
tr     con      $644b    'turn right
tl     con      $4b32    'turn left
rr     con      $6464    'rotate right
rl     con      $3232    'rotate left

'wander values
wstate var      byte      'shared byte
wDir    var      word      'wander value
wDur    var      byte      'wander duration

'set up for running
wstate =0      'initial wander state
main:
    gosub wander
    gosub act
goto main
'=====
'Behaviours follow
'=====
wander:
    branch      wstate,{wDir,wcDur}      'randomly wander around
    wDur = wDur - 1
    if wDur > 0 then wDone1
        drive = wDir
        wstate = 0
        'state 2 immed. follows
        'get direction
        'reset state
wDone1:
    return
        'completed

wcDir: 'choose direction
    random seed
    i = seed & %111
    lookup i, {tr,fd,fd,fd,rr,fd,fd,tl},wDir      'random direction
    wstate = 1
    return
        'mask off for 0-7 only
        'choose direction
        'next state

wcDur: 'choose duration
    random seed
    wDur = (seed & %111111) + 20
    wstate = 2
    return
        'random direction and duration
        'mask for 64 choices
        'next state

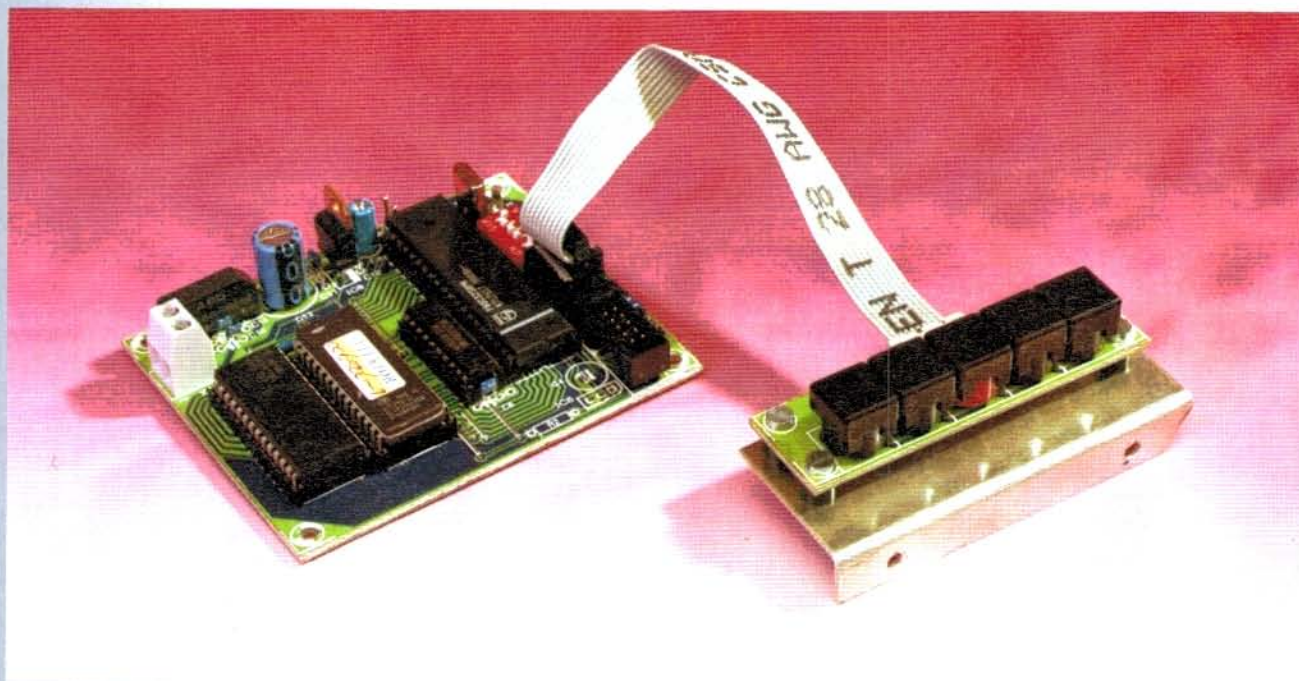
act:
    if aDur > 0 then aDec
        aDur = SACT
        pulsout LEFT,ldrive * 10
        pulsout RIGHT,rdrive * 10
        'moves servo motors
        'already doing one, got here
        'times through this one

aDec:
    aDur = aDur - 1
        'decrement stuff

aDone:
return
```

# Digital volymkontroll

## för vår Audio DAC 2000



Denna krets är speciellt framtagen för er som inte är nöjd med den enkla, fixerade volymnivån hos Audio DAC 2000. Minimala modifieringar av DAC-kortet och en enkel processorkrets är allt som behövs för lägga till en tryckknappsstyrd volym och balanskontroll *de luxe*. Det finns även en justerbar grundinställning!

I del 2 av beskrivningen av Audio DAC 2000 nämnde vi kortfattat att det digitala interpolationsfiltret DF1704 inkluderar en digital dämpare som kan justeras mjukvarumässigt. Vi nämnde också att denna egenskap skulle kunna användas i kombination med en extern processor för att bygga en digital volymkontroll.

Nu var det kanske inte så klokt att nämnda detta, eftersom vi borde begripit att detta omedelbart skulle väcka intresset hos många av våra läsare. Vi var sålunda tvungna att omedelbart påbörja arbetet med att ta fram en sådan krets för Audio DAC 2000.

Resultatet beskrivs i denna artikel. Detta är en processorkrets som inkluderar mjukvara som utvecklats speciellt för denna applikation. Med dessa kan volymen och balansen regleras mycket precist med hjälp av tryckknappar för upp/ned och vänster/höger. Stegstorleken är så liten (0,5 dB) att justeringen i stort sett är kontinuerlig över hela

området. Med en femte knapp går det att snabbt välja en förinställd volymnivå och en 'rak' frekvenskurva. Om knapparna 1 och 5 trycks in samtidigt aktiveras en mute-funktion.

Kretsen är enkel att bygga och det är också mycket enkelt att ansluta den till Audio DAC-kretsen.

### LITE HÅRDVARA

För att kunna styra de digitala dämparna i DF1704 är det nödvändigt att switcha denna digitala filter-IC till sitt mjukvarumod. Som vi noterade i del 2 av Audio DAC 2000 projektet väljs detta mod genom att lämna ben 10 på DF1704 öppet. Benen 13, 12 och 11 är då ML, MC respektive MD-ingångar hos den tretrådiga mjukvarustyrda kontrollporten. Allt som behövs för att göra dessa ingångar åtkomliga på DAC-kortet är att ta bort DIP-omkopplaren S2 och ersätta denna med en 8-polig flatkabelkontakt i DIL-format. De relevanta delarna av Audio DAC-

ens kretsschema återges på nytt i figur 1 för tydlighets skull.

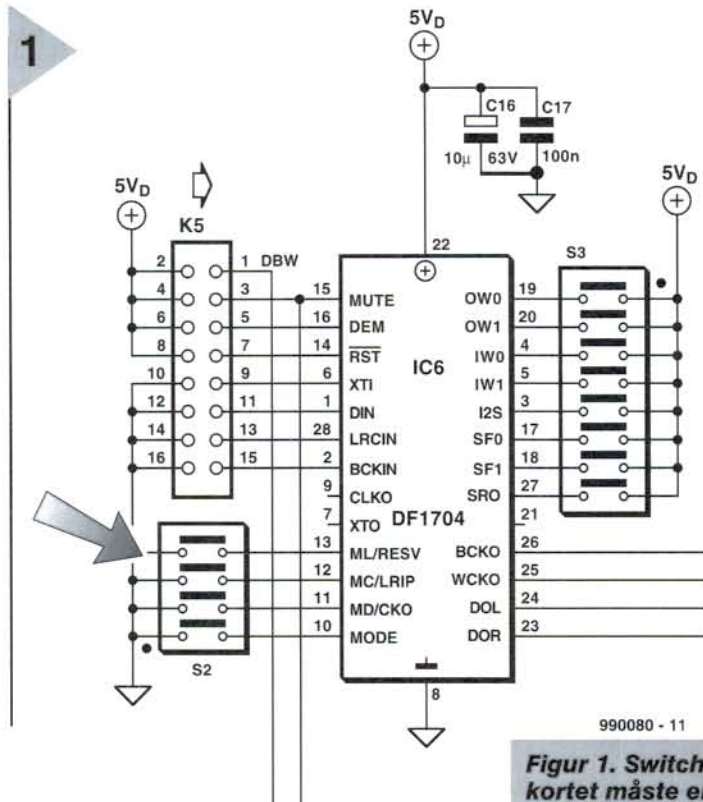
Den kompletta hårdvaran för volymkontrollen visas i figur 2. Som ni kan se är det en riktig grundkrets, baserad på en 80C32-processor (IC1), en adresslatch (IC2), ett EPROM (IC3) med den nödvändiga mjukvaran, samt ett RAM (IC4). Flatkabeln från DF1704 ansluts till kontakten K1. De fem kontrollswitcharna S1 till S5 är placerade på en separat del av kretskortet och kan skiljas från detta. En kort bit flatkabel mellan K2 och K3 kopplar ihop de båda delarna av kretskortet.

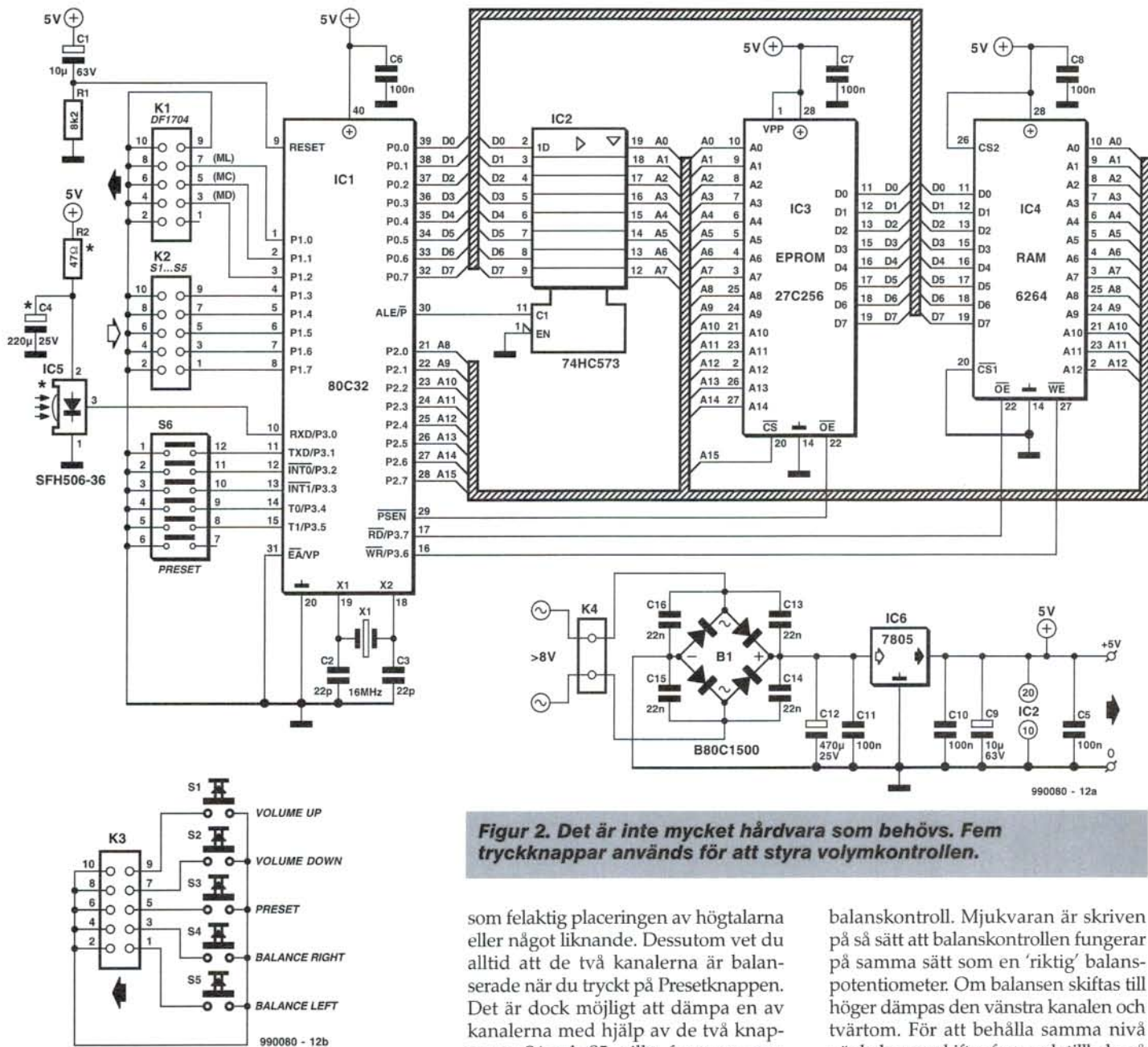
Kretsen inkluderar också en 5-polig DIP-switch (S6) för grundinställning av volymen för switchen S3 och det finns också plats reserverad för en optisk mottagare (IC5). Även om det för närvarande inte finns behov av denna komponent så innebär den dock en möjlighet för den som är intresserad att kunna bygga en fjärrkontroll för volymen. Naturligtvis behövs det då 'hemsnickrad' mjukvara för att stöda denna möjlighet.

Vad som är kvar i figur 2 är strömförsörjningen. Som du kan se är den byggd på konventionellt sätt med ett likriktare, filter samt en spänningsregulator (IC6). Med tanke på eventuella framtida applikationer så går det också att nå den stabiliserade spänningen på +5 V externt.

## FYRA REGISTER

DF1704 har fyra interna register som kallas MODE0 till MODE3 och som bestämmer inställningen av filtret. Dämparna kan endast styras i mjukvarumod, vilket betyder att filtren måste förses med fasta inställningar av processorn efter det att kretsen slagits på. DIP-switcharna på DAC-kortet har ingen funktion i mjukvarumod. Den detaljerade information som behövs för att programmera de olika inställningarna kan du hitta på sida 10 i databladet för DF1704. Om du är intresserad kan du hitta detta datablad på Internetadressen [www.burr-brown.com](http://www.burr-brown.com). Förutom de inställningar som också kan göras med hårdvara så finns det tre inställningar som relaterar till hur dämparna styrs. I registret MODE3 är bit 2 ATC-biten (attenuator control). Denna bit bestämmer om de vänstra och högra dämparna arbetar med samma data eller separata data. Om ATC är satt till 0 kan de två dämparna ställas in oberoende av varandra. Om ATC är satt till 1 är båda dämparna hopkopplade och ställs in med data för den vänstra dämparen, som finns i MODE0-registret. I vår volymkontroll valde vi att använda separata kontroll-data för att kunna styra balansen, vilket betyder att ATC alltid är satt till 0. Register MODE0 innehåller således 8-bitarsdata för den vänstra kanalen och MODE1 innehåller 8-bitarsdata för den





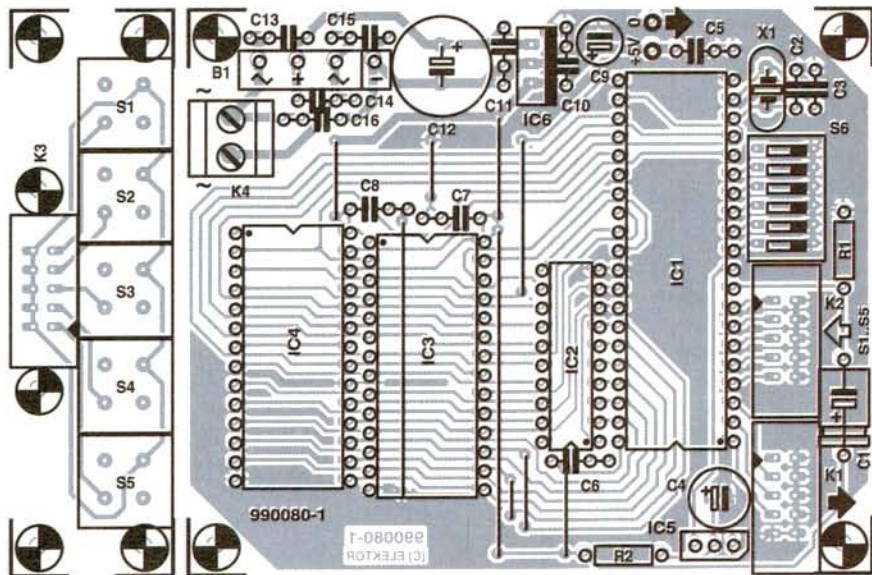
**Figur 2. Det är inte mycket hårdvara som behövs. Fem tryckknappar används för att styra volymkontrollen.**

som felaktig placeringen av högtalarna eller något liknande. Dessutom vet du alltid att de två kanalerna är balanserade när du tryckt på Presetknappen. Det är dock möjligt att dämpa en av kanalerna med hjälp av de två knapparna S4 och S5, vilka fungerar som

balanskontroll. Mjukvaran är skriven på så sätt att balanskontrollen fungerar på samma sätt som en 'riktig' balanspotentiometer. Om balansen skiftas till höger dämpas den vänstra kanalen och tvärtom. För att behålla samma nivå när balansen skiftas fram och tillbaka så

stegstorleken hos S6 är 4 dB. I praktiken är detta en fullt tillräcklig upplösning för att ställa in den genomsnittliga lyssningsnivån trots att den maximala avvikelser i nivån aldrig är mer än  $\pm 2$  dB när en slutförstärkare drivs direkt. Den maximala nivån, som uppnås när alla switcharna hos S6 är ON, motsvarar således nivån i hårdvarumodet. Vi har valt att använda samma nivåer för vänstra och högra kanalen (ingen offset för balansen) eftersom det i praktiken visat sig att balanskontrollerna alltid står i mitten. I annat fall är det något fel med resten av installationen,

4



## KOMPONENTLISTA

### Motstånd:

R1 = 8kΩ  
R2 = 47Ω\*

### Kondensatorer:

C1 = 10μF 63V axial  
C2, C3 = 22pF  
C4 = 220μF 25V radial\*  
C5-C8, C10, C11 = 100nF keramik  
C9 = 10μF 63V radial  
C12 = 470μF 25V radial  
C13-C16 = 22nF keramik

### Halvledare:

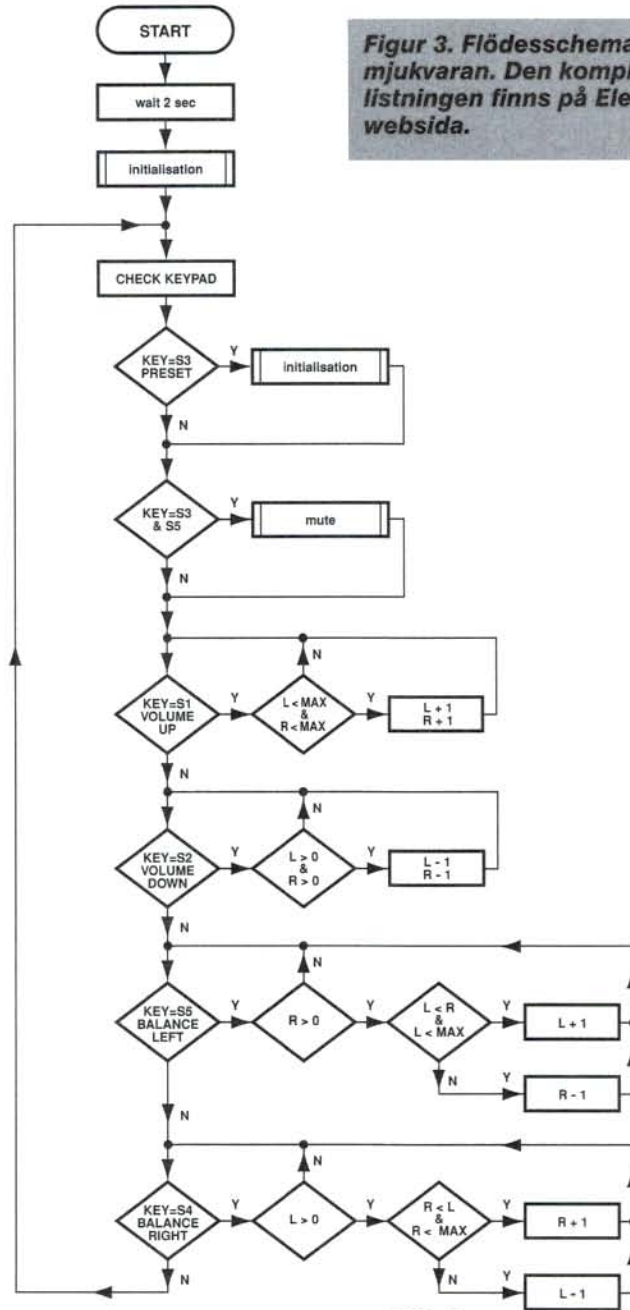
B1 = B80C1500 (rektangulärt hölje)  
IC1 = 80C32-16  
IC2 = 74HC573  
IC3 = 27C256 (programmerad, ordernr 006506-1)  
IC4 = 6264  
IC5 = SFH506-36\*  
IC6 = 7805

### Övrigt:

K1, K2 = 10-pol boxheader  
K3 = 10-pol krets-kortkontakt för flatkabel  
K4 = 2-pol kopplingplint för krets-kort  
S1-S5 = tryckknapp, 1 slutande kontakt, t.ex., ITT/Schadow D6-Q-BK + D6Q-BK-CAP  
S6 = 5- eller 6-pol DIP-switch  
X1 = 16MHz kristall  
Krets-kort 990080-1 (se Läsa-service)  
Krets-kort och programmerad EPROM: ordernr 990080-C

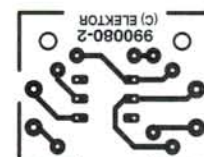
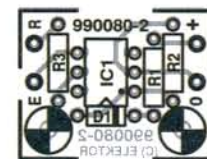
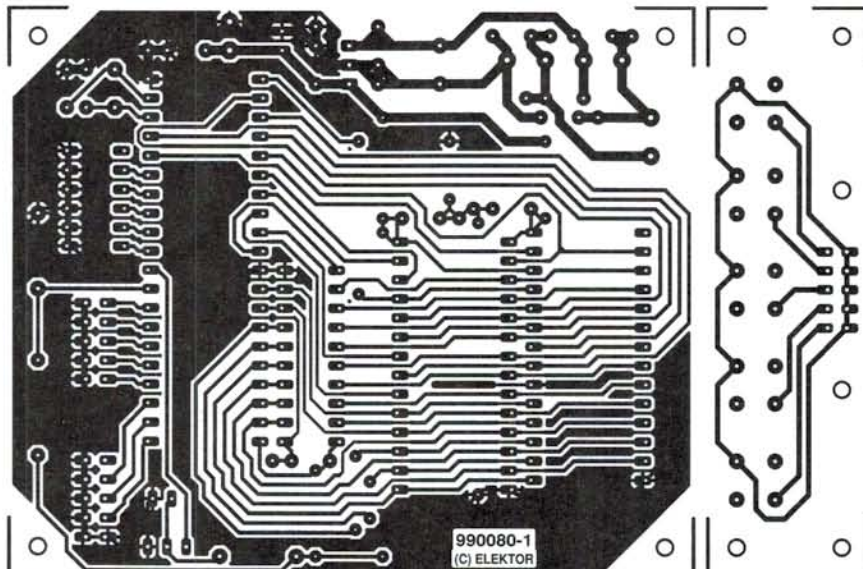
\*) om så behövs

3



Figur 3. Flödesschema för mjukvaran. Den kompletta listningen finns på Elektors websida.

Figur 4. 'Tangentbordet' kan separeras från resten av kortet.



återgår den redan dämpade kanalen till den andra kanalens nivå innan ny dämpning sker.

När det gäller volymkontrollen är det nödvändigt att kolla att kontrollvärdet inte överskrider den maximala nivån när volymen ökas eller den minsta nivån när volymen sänks. Om tryckknapparna i fråga hålls intryckta ökar eller minskar en repeterande programslina dämparvärdet med 1. Naturligtvis måste du kolla maximum och minimumvärdet även för balansjusteringen.

Före initialiseringen är de tre LSB-erna hos port 1 definierade som utgångar och resten som ingångar. Efter initialiseringen övervakar en ändlös slinga port 1. Varje tryckknapp har sin egen funktion och det finns bara en giltig kombination av två knappar, vilket är 1 + 5 för mute-funktionen. Programmet kollar i tur och ordning om S3, S1+S5, S1, S2, S5 eller S4 är intryckt. Att trycka på någon annan knappkombination har ingen effekt.

### KRETSKORTET

Kretskortet för den digitala volymkontrollen visas i figur 4. Som vi redan sagt kan delen med de fem tryckknapparna sågas loss så att detta kort kan monteras på lådans front (till exempel).

Vilka andra praktiska tips kan vi ge när det gäller kortet? Förutom de fyra IC-kretsarna, spänningsregulatorn och de två kontakterna så finns det inte mycket på kortet. Vi kan alltså begränsa oss till att föreslå att du använder hållare av bra kvalitet, vara uppmärksam på hur du vänder de polariserade komponenterna och se till att du inte glöm-

	MODE0	MODE1	MODE2	MODE3
B0	1	1	1	1
B1	1	1	0	0
B2	1	1	0 (res)	0
B3	S6-1	S6-1	1	0
B4	S6-2	S6-2	0	0 (res)
B5	S6-3	S6-3	1	0
B6	S6-4	S6-4	1	0
B7	S6-5	S6-5	0 (res)	0
B8	1	1	0 (res)	0 (res)
B9	0	1	0	1
B10	0	0	1	1
B11...B15 för alla register	0 (res)			

(res = reserverad)

mer trådbyglarna. Detta är det absolut vanligaste felet.

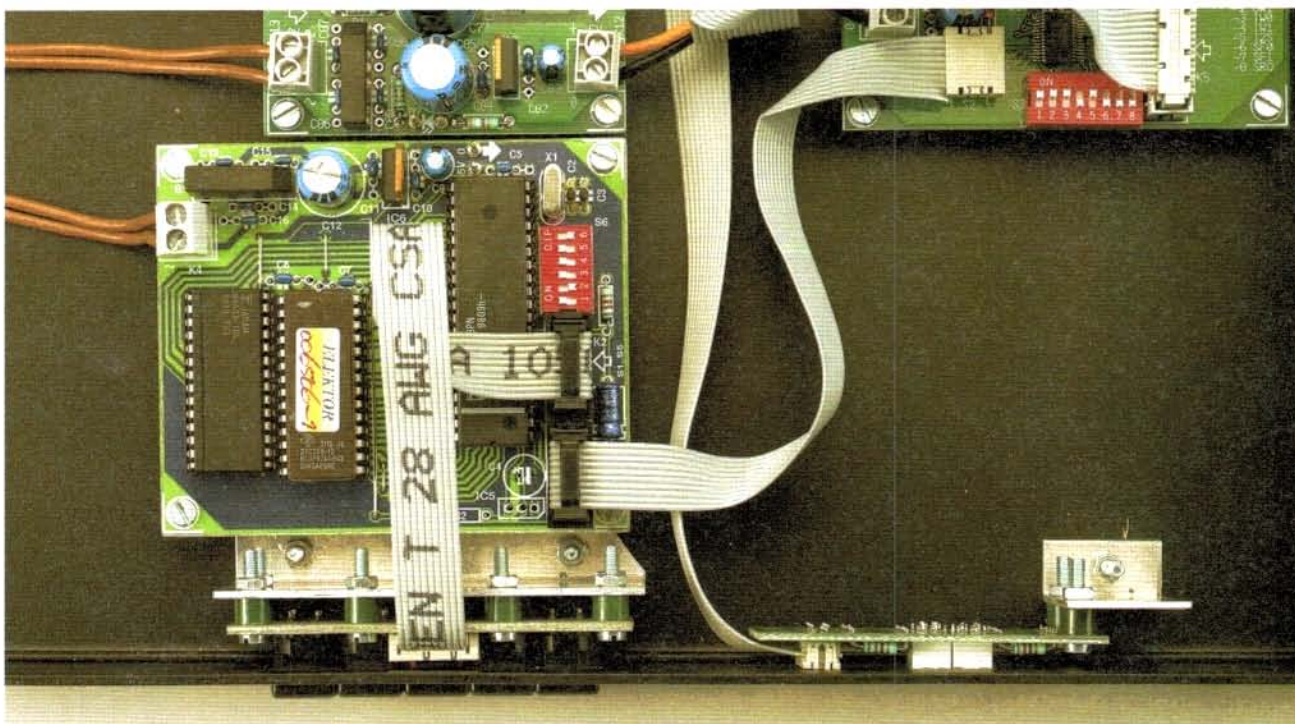
Som du kan se av bilden i figur 5 är det lätt att bygga in volymkontrollen i lådan för Audio DAC 2000.

Montera bara det stora kortet i botten på lådan med hjälp av distanser och tryckknappskortet på fronten, eventuellt med hjälp av en vinkelplåt. Plugga in flatkabeln mellan K2 och K3 på volymkontrollen och sedan flatkabeln mellan K1 och 'S2 kontakten' på DAC-kortet och det är klart. Inte helt förresten. Du måste naturligtvis också ansluta en matningsspänning på 8 V (AC eller DC) till K4. Om du har använt det transformatorkort som vi rekommenderade för Audio DAC-en och beskrev i nummer 1/2000 så behöver du bara dra två kablar till kontakt K4 på transformatorkortet.

(990080-1)

**Tabell 1. Summering av inställningarna för registren MODE0 till MODE3.**

**Figur 5. Det är lätt att montera volymkontrollen i lådan för Audio DAC 2000.**



# elektronik on-line

## Bygg din egen plotter

*rita, skära och fräsa med stegmotorer*

En plotter är en typ av ritapparat som de flesta privatpersoner inte har råd med, speciellt inte när det gäller större pappersformat. Men om du har lämpliga stegmotorer, och den mekaniska kunskapen, så varför inte bygga en egen plotter? Resultatet av detta kan bli bättre än du kanske förväntat dig.

De av våra läsare som varit med oss en längre tid kommer säkert ihåg den plotterdesign som vi publicerade under våren 1988. Denna följdes av en uppgradering, Mark II, våren 1990 och förbättrad mjukvara året därefter. På den tiden var dessa artiklar revolutionerande, men utvecklingen har gått framåt sedan dess.

Vi var faktiskt en smula överraskade av att det fortfarande finns hobbyister som inte bara designar och bygger sina egna plottar, utan som också är villiga att dela med sig av sina erfarenheter till andra intresserade personer. Internet innehåller en stor mängd information och denna är tillgänglig för alla.

Den förste entusiasten vi skall nämna är Luberth Dijman som, en stor plotterentusiast själv, var den som kom med artikel förslaget. Hans plottersida på

<http://home.wxs.nl/~luberth/plotter/plotter.htm>

innehåller en mängd information om plottar. Där finns, bland annat, Luberths egen gör-det-själv plotter och ett stort antal program som är drivna en plotter från en dator. Det finns mjukvara för olika plattformar och med olika programspråk (QBASIC, Pascal, C), det finns något för alla. Förutom styrprogrammen (anpassade för olika typer av drivar-IC) finns det olika omvandlingsprogram, till exempel ett som omvandlar true type fonter till vektorformat.

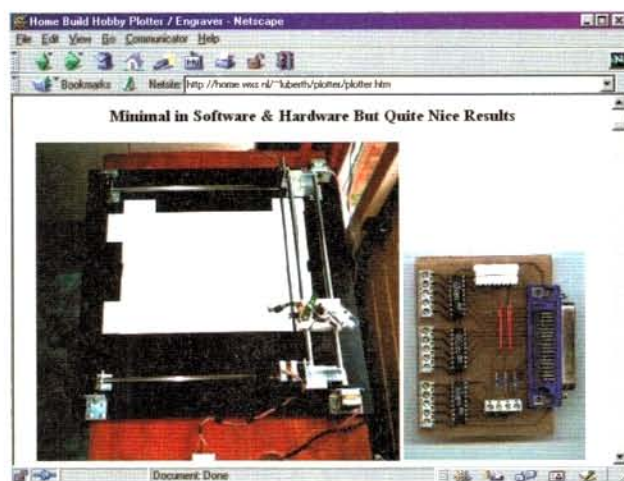
Sidan har också en värdefull samling länkar till andra Internetplatser som handlar om plottar, från kompletta gör-det-självbeskrivningar av fräsmaskiner till leverantörer av delar för hemmabyggare.

Luberth är också moderator i gruppen **Homebuilt Plotter Discussion Forum** som är öppen för alla som har frågor om plottar.

En annan intressant plottersida är **Scumari Technohobbies** som sköts av Rick Schuitemaker vid

<http://www.scumari.demon.nl/plotter/plotnl.htm>

En mycket vacker hemgjord CNC fräsmaskin visas på webben av **BCS Computersysteme** från Tyskland. Det skall noteras att detta är ett privat bygge och att BCS inte säljer



några delar för projektet. Adressen är:

<http://www.bcs-computersysteme.com/software/cnc/index.htm>

Även i Brasilien har man designat och byggt plottar, vilket du kan se på

<http://www.geocities.com/ResearchTriangle/3215/plotter.htm>

Denna sida har dessutom en länk tillbaka till Luberth Dijkmans sida! Internet binder verkligen människor samman.

Om du vill veta mer om hur du bygger en riktigt stor plotter skall du titta på **Dave's E-Size Plotter** vid

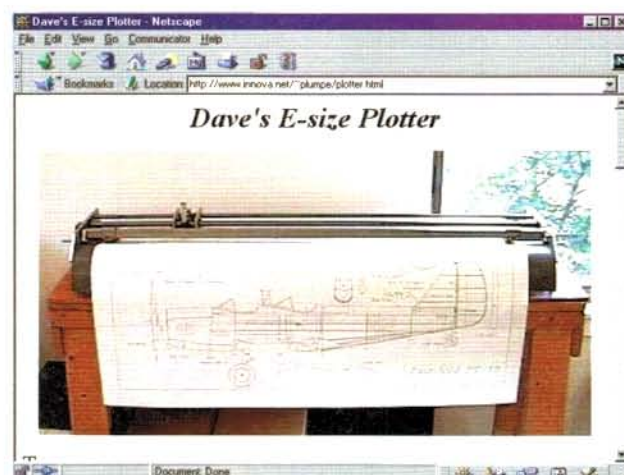
<http://www.innova.net/~plumpe/plotter.html>

**Dave Rigotti** är en av många privatpersoner som erbjuder komponenter för plotterbygge, som stegmotorer och drivar-IC:

<http://hometown.aol.com/drigotti/HobbyCNC.htm>

Om du inte har något emot en hel del mekaniskt arbete så kan du hitta en hel del inspiration från de webbsidor som vi har listat här.

(005021-1)



# Hastighetsmätare

## för modellbilar

En pålitlig hastighetsmätare är utan tvekan mycket viktigt för en modellracerbil. Men de kostar en slant. I denna artikel beskriver vi hur en vanlig cykeldator kan användas som en hastighetsmätare, som är både bekväm, noggrann och billig.



Modellbilsentusiaster är naturligtvis alltid intresserade av tekniken och prestandan hos sina bilar. De vill, till exempel, gärna veta exakt hur fort bilarna går, så att de, till exempel, kan välja rätt utväxling för bästa prestanda. Andra faktorer kan också vara intressanta, som den totala sträcka som en bil tillryggalagt, eftersom det kan vara värt att veta hur långt en bil kan gå på en batteriladdning eller på en full tank.

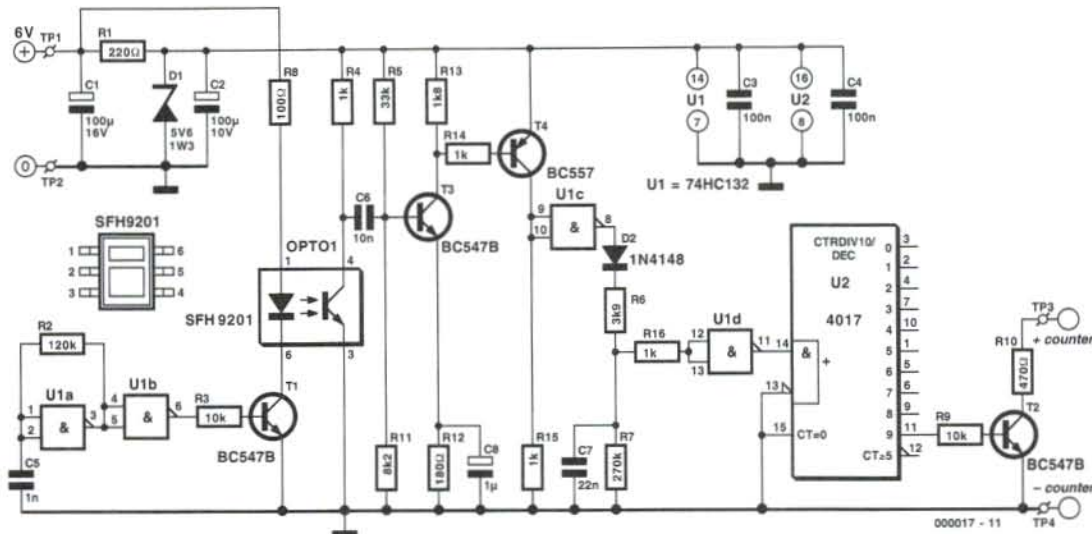
Det finns imponerande instrument att köpa för att utföra dessa mätningar, ända upp till kompletta telemetriska system. Dessa varierar dock i pris, från mycket dyra till fruktansvärt dyra. Detta är tillräckligt skäl för modellbyggare med en begränsad budget att leta efter billigare alternativ.

Konstruktören av den hastighetsmätare som visas här har tagit fram ett sådant alternativ och det är lika enkelt som billigt. Det handlar om en adapterkrets som gör att du kan använda en billig cykeldator som hastighets-

mätare. Dessa kostar från en hundralapp och har den fördelen att de inte bara visar hastigheten utan också den totala körtiden, genomsnittlig hastighet och den totala körsträckan. Det är svårt att föreställa sig något bättre.

### EN ANNAN SENSOR

De flesta av våra läsare vet förmodligen hur en cykeldator får sina pulser. De genereras av en pickup som registrerar varven på framhjulen. Pickupen består av två komponenter. En av dessa är en magnet som sitter på cykelhjulet och den andra är en magnetisk reedkontakt som sitter fast på framgaffeln. Reedswitchen är ansluten till cykeldatorn via en tunn tråd. Varje gång magneten passerar reedswitchen sluts kontakten i denna och datorn får en räknepuls. Denna typ av pickup kan inte användas för en modellbil. Även om du kan fästa magneten på hjulet skulle det bli så obalanserat att du inte kan köra bilen. Alltså behövs någon annan form av pickup.



**Figur 1. Sensor IC OPTO1 omvandlar variationer i det reflekterade ljuset till elektriska pulser. Dessa förstärks och repetitions hastigheten divideras sedan med 10.**

En optisk sensor är ett ganska uppenbart val. Det är en friktionsfri sensor utan kontakt, på samma sätt som en magnet/reedswitchkombination, men med den extra fördelen att den inte har någon rörlig massa.

Magneten ersätts i detta fall med ett högreflekterande band på hjulsidan och reedswitchen ersätts med en infraröd reflektiv sensor.

Den bästa lösningen för det reflekterande bandet visade sig vara vit eller silverfärg. Praktisk erfarenhet visar att detta skall vara ca 1 cm brett, men det får inte vara bredare än en tiodel av bredden hos den icke-målade delen av hjulet. Den reflektiva sensorn skall naturligtvis monteras på så sätt att den kan den kan detektera den målade och den icke-målade delen av hjulet.

### ADAPTERKRETSEN

Det enda som den nya sensorn behöver ytterligare är en krets som omvandlar signalen från den reflektiva sensorn till pulser som kan användas av cykeldatorn. Det är två saker som måste göras. Först att omvandla de optiska pulserna till elektriska pulser och dessutom anpassa pulsfrekvensen. Det förstnämnda behöver ingen ytterligare förklaring. Det andra har att göra med skillnaden mellan omkretsen hos ett cykelhjul och omkretsen på hjulet hos en modellbil. Små hjul roterar snabbare för samma hastighet hos fordonet och producerar pulser med en högre hastighet.

Även om cykelhjulets omkrets kan ställas in på cykeldatorn så finns det naturligtvis en begränsning när det gäller detta. Det är inte möjligt att ställa in en hjuldiameter som bara är en tiodel, eller mindre, enbart med hjälp av omkretsinställningen. Alltså måste antalet pulser minskas med en lämplig faktor.

### DEN PRAKTISKA KRETSEN

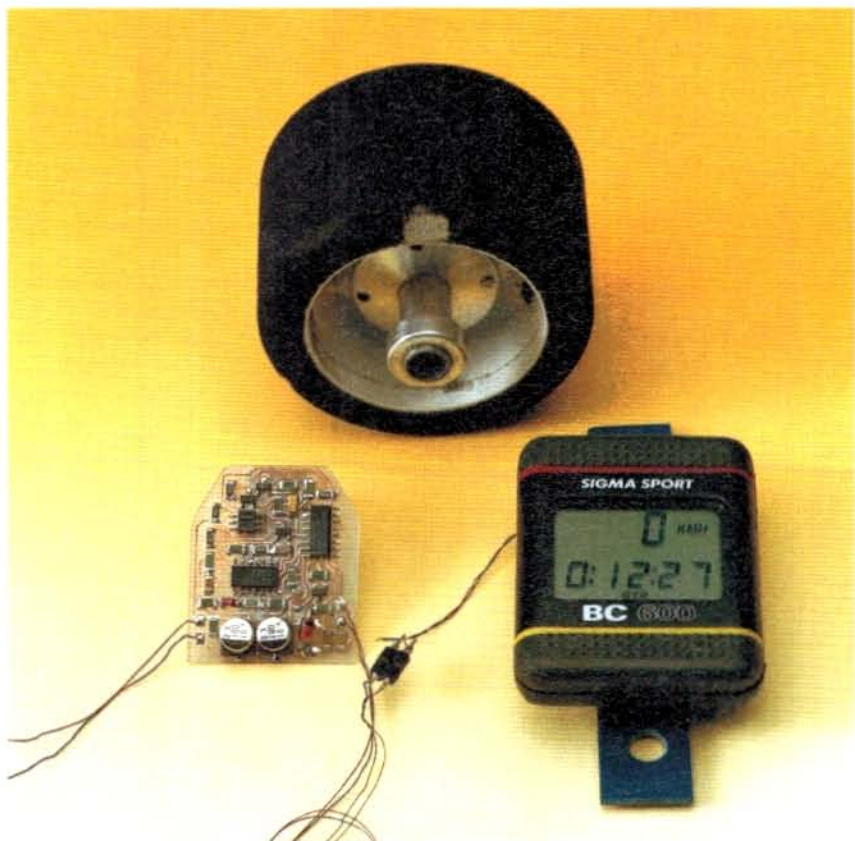
Som man kan se av figur 1 kan vi klara

dessa krav med relativt enkel elektronik.

Hjärtat i kretsen är den reflektiva sensorn (OPTO1). För detta används en SFH9201 IC från Siemens. I den första versionen av kretsen drevs LED-en helt enkelt av en likström. Detta visade sig vara mindre lämpligt eftersom sensorn också reagerade för det omgivande ljuset. Detta gav så många felpulser att hastighetsmätaren inte fungerade korrekt. Vi bytte därför till att driva sensorn med en 10-kHz växelström. Detta har den fördelen att en AC-förstärkare kan användas för detektor kretsen, vilket i stort sett eliminerar effekterna av det omgivande ljuset.

Signalen på 10 kHz för LED-en produceras av en oscillator byggd runt IC1a. Grunden IC1b fungerar som en buffert som driver sensor-LEDen via transistor T1.

När det vita bandet på hjulet passerar framför sensorn kommer fototransistorn i sensorn att kortvarigt leda med en hastighet på 10 kHz. En pulståg med en frekvens på 10 kHz produceras således över motståndet R4. Denna sig-



## KOMPONENTLISTA

### Motstånd:

R1 = 220k $\Omega$   
R2 = 120k $\Omega$   
R3,R9 = 10k $\Omega$   
R4,R14,R15,R16 = 1k $\Omega$   
R5 = 33k $\Omega$   
R6 = 3k $\Omega$   
R7 = 270k $\Omega$   
R8 = 100 $\Omega$   
R10 = 470 $\Omega$   
R11 = 8k $\Omega$   
R12 = 180 $\Omega$   
R13 = 1k $\Omega$

### Kondensatorer:

C1 = 100 $\mu$ F 16V  
C2 = 100 $\mu$ F 10V  
C3,C4 = 100nF  
C5 = 1nF  
C6 = 10nF  
C7 = 22nF  
C8 = 1 $\mu$ F 10V

### Halvledare:

D1 = zenerdiod 5V6 1W3  
D2 = 1N4148  
T1,T2,T3 = BC547B  
T4 = BC557B  
IC1 = 74HC132SO  
IC2 = 4017SO  
OPTO1 = SFH9201 (Siemens)

nal kopplas ut av kondensator C6 och förstärks sedan av en AC-förstärkare som bildas av transistorerna T3 och T4. Detta resulterar i en 10 kHz pulsvåg över motståndet R15. Denna buffras av grinden IC1c och läggs sedan till en detektor krets som består av dioden D2, motstånden R6 och R7, samt kondensator C7. Uppgiften för detektor kretsen är att omvandla den korta serien pulser till en '1'. Komponentvärdena är ganska kritiska eftersom kondensator C7 skall laddas innan bandet har passerat sensorn helt och hållet, men den skall också laddas ur helt innan ban-

det kommer fram till sensorn igen och ett nytt pulståg skapas.

Utgångssignalen från detektorn buffras av grinden IC1d och kommer slutligen fram till den sista delen av kretsen, dela-med-tio räknaren IC2. Från denna skickas endast var tionde puls vidare till transistor T2. Den öppna kollektorkretsen hos denna transistor ansluts till ingången på cykeldatorn.

## STRÖMFÖRSÖRJNING

Kretsen drivs med en matningsspänning på 5 V. Denna kan man normalt ta från mottagarmodulen i bilen. I prototypen fanns en 6 V matningsspänning tillgänglig för mottagaren. Kondensator C1 ger en extra filtrering för denna spänning, som sedan används för att försörja lysdioden i optokopplaren (U+). Matningsspänningen för resten av kretsen stabiliseras runt 5 V av motståndet R1 och zenerdioden D1. Kondensator C2 fungerar som glättningskondensator medan C3 och C4 ger lokal avkoppling för IC1 och IC2.

## SJÄLVA BYGGET

Kretsen är inte speciellt kritisk och med tanke på det lilla antalet komponenter inte heller svår att bygga. Hur den skall byggas beror till stor del på formen hos bilen i fråga. Den viktigaste faktorn är naturligtvis att sensorn OPTO1 måste ha en obegränsad sikt till det reflektiva bandet på hjulet.

Eftersom det alltid är ont om utrymme i en modellbil så har författaren designat ett kretskort som nästan bara använder ytmonterade komponenter. Figur 2 visar layouten och komponentplaceringen. Kortet finns inte att köpa från oss.

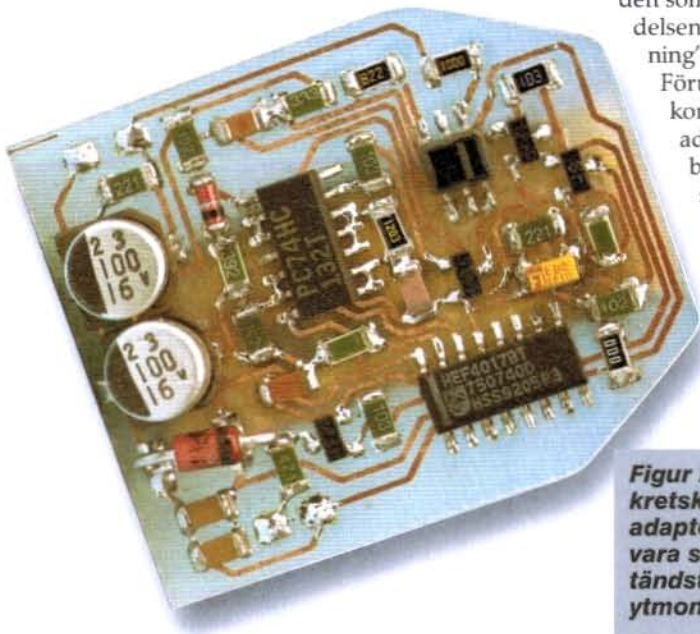
Även om detta kort fungerade mycket bra i den prototyp vi testade så skall vi understryka att kretsen inte har labbttestats av oss. Du skall således betrakta den som ett förslag i betydelsen 'en möjlig lösning'.

Förutom den exakta konstruktionen av adapterkretsen så beror naturligtvis sättet hur ni monterar cykeldatorn på hur bilen ser ut där den skall sitta. Med tan-

ke på hur många olika modeller det finns måste vi överlåta detta åt de av våra läsare som tänker bygga hastighetsmätaren.

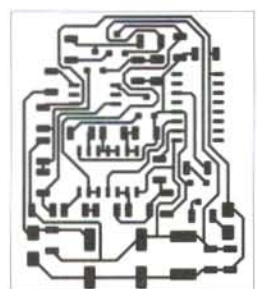
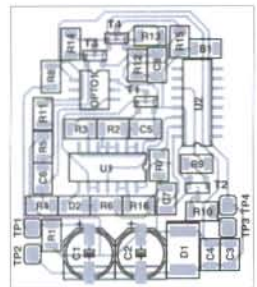
Att ansluta kretsen är mycket enkelt. Dra 6-V matningen till elektrolytkondensatorn C1 (med rätt polaritet) och anslut de två ledarna från cykeldatorn till motståndet R10 respektive jord. På komponentplaceringen kan man se anslutningen av spänningsmatningen in till TP1 och TP2 medan utgångsanslutningarna är märkta TP3 och TP4. Slutligen en sista anmärkning när det gäller inställningen av hjulomkretsen i cykeldatorn. Glöm inte faktorn 10 som ges av delarkretsen i adapterkretsen. Om, till exempel, omkretsen på hjulet i modellbilen är 21 cm så skall du ställa in omkretsen på 210 cm i cykeldatorn.

(000017-1)



Figur 2. Ett förslag till kretskortslayout för adapterkretsen. Kortet kan vara så litet som en tändstickask om du använder ytmonterade komponenter.

2

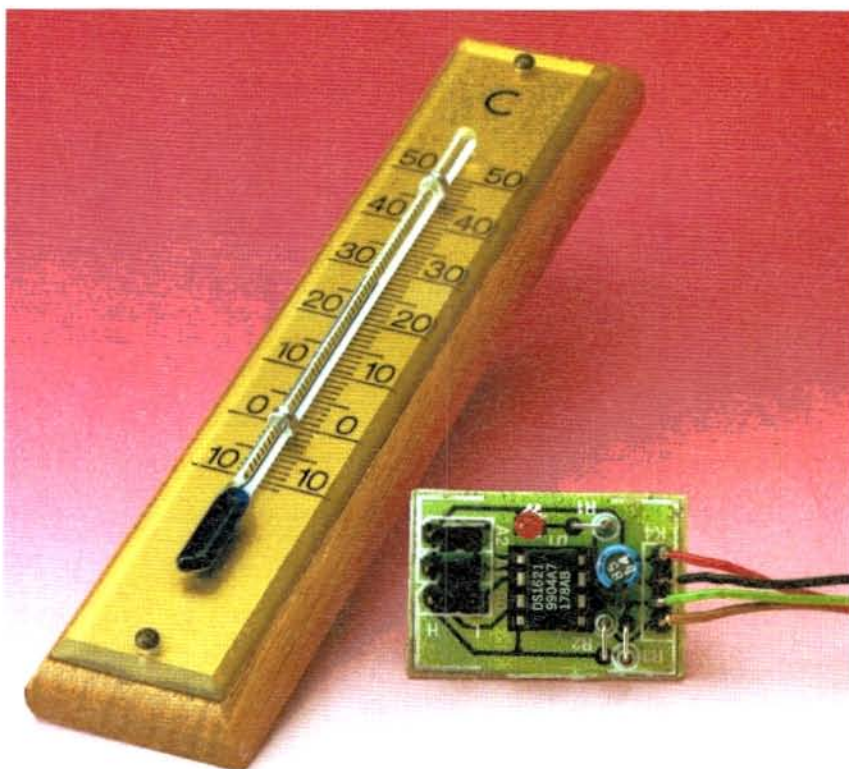


# Temperaturmätningar med en DS1621

*en smartare sensor med ett I<sup>2</sup>C-bussinterface*

Inom många områden av det dagliga och tekniska livet är temperaturmätningar den viktigaste instrumentella uppgiften.

'Intelligenta' digitala sensorer, som kan göra mer än att bara skicka ut temperaturavläsningar, vinner mer och mer mark som komplement till den klassiska analoga tekniken. Ett exempel på en sådan sensor är DS1621 från Dallas Semiconductors.



## Tekniska specifikationer

*Inga externa komponenter behövs*

*Temperaturmättningsområde -55 °C till +125 °C*

*Upplösning 0.5 °C*

*9-bitars temperaturmättningsvärde med tecken och storhet*

*Omvandlingstid 1 sekund*

*Användarprogrammerbar termostatfunktion*

*Seriellt tvåtråders interface (I<sup>2</sup>C) med öppen-drain anslutningar*

Naturligtvis är huvuduppgiften för en modern digital temperatursensor att mäta den aktuella temperaturen. Den gör detta genom att mäta sin egen temperatur, som i stort sett är densamma som den omgivande temperaturen, och sedan omvandla detta analoga temperaturvärde till ett digitalt ord som den sedan sänder till en mikrokontroller via ett serieinterface. Användaren behöver inte oroa sig för

den analoga teknik som behövs för kalibrering, förstärkningen av den termiska spänningen med opampar, linjäriseringen av egenskapskurvorna osv. eftersom allt detta sköts om av chipet själv eller också kan det enkelt ställas in med mjukvara. Externa komponenter är därför i stort sett obehövligen. Men en intelligent sensor har dessutom ytterligare funktioner, som temperaturloggning med DS1615 (Allt om Elekt-

ronik 8/99) eller termostatfunktioner, som den DS1621 som beskrivs i denna artikel. Huvudegenskaperna hos denna IC summeras i rutan för tekniska specifikationer.

## TEMPERATURMÄTNINGAR

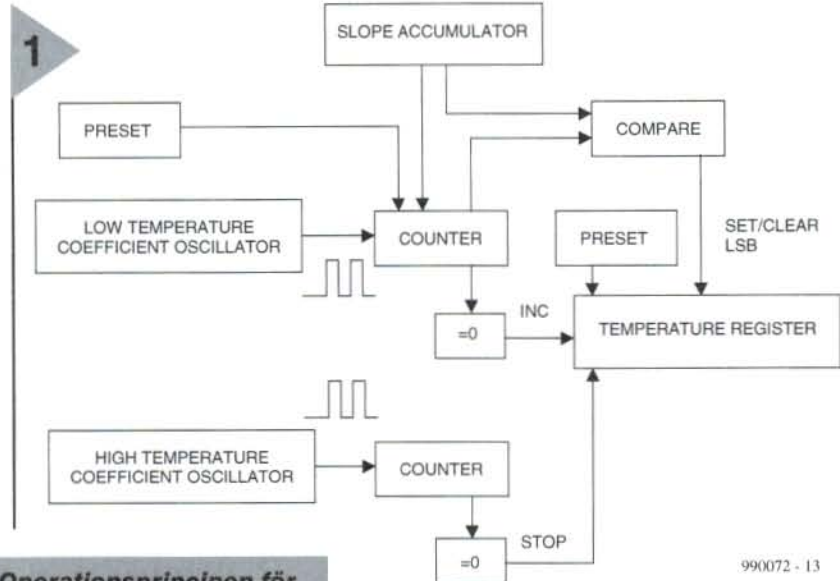
Den primära funktionen för DS1621 är naturligtvis att mäta temperaturen. Figur 1 visar de moduler som finns i IC-kretsen för detta ändamål. DS1621 mäter temperaturen genom att mäta det antal klockcykler som produceras av en oscillator med en låg termisk koefficient inom ett givet tidsfönster. Bredden på detta fönster bestäms av en andra oscillator som har en större temperaturkoefficient. Räknares för-laddas med ett värde som motsvarar en temperatur på -55 °C. Om räknares när värdet noll inom tidsfönstret inkrementeras (ökas) tem-

peraturregistret (som också är inställt på det värde som representerar -55 °C) för att indikera att temperaturen är högre än -55 °C. Räknares klockas tills den når noll och återstartas sedan så länge som tidsfönstret är öppet.

Räknares ackumulerar inte alltid samma antal klockpulser inom tidsfönstret för varje gradskillnad i temperaturen. Detta beror på att det finns en variabel offset adderad till räknares av Slope Accumulator för varje grad av temperaturskillnad, för att kompensera det olinjära beteendet hos oscillatoren över temperaturområdet. Med detta kan man också uppnå den mycket höga temperaturupplösningen på 0,5 °C.

Den uppmätta temperaturen lagras i temperaturregistret som ett 9-bitars värde. Som visas i tabell 1 går mätområdet för DS1621 från -55 °C till +125 °C i steg om 0,5 °C. Den aktuella temperaturen skickas ut via ett två-tråders seriellt interface som svar på kommandot READ TEMPERATURE, antingen som ett enda byte med en upplösning på 1 °C eller som två byte med en upplösning på 0,5. I det senare fallet skickas mest signifikanta bit (MSB) först, följt av minst signifikanta bit (LSB) som endast specificerar den minst signifikanta biten hos temperaturvärdet (0,5 °C). De återstående sju bitarna hos LSB är satta till noll.

Man kan använda ett trick för att uppnå en ännu högre upplösning. Först läses temperaturen och 0,5 °C biten (LSB) klipps bort. Det resulterande värdet kallas TEMP\_READ. Det värde som finns kvar i räknares när tidsfönstret stängs kan läsas med kommandot READ\_COUNTER och sedan lagras i variabeln COUNT\_REMAIN. Värdet i Slope Accumulator kan sedan läsas med hjälp av kommandot READ\_SLOPE. Detta värde (COUNT\_PER\_C) är antalet räknade flanker per grad Celsius vid den aktuella temperaturen.



Figur 1. Operationsprincipen för temperaturmätningar.

Tabell 1. Temperatur vs. datavärde

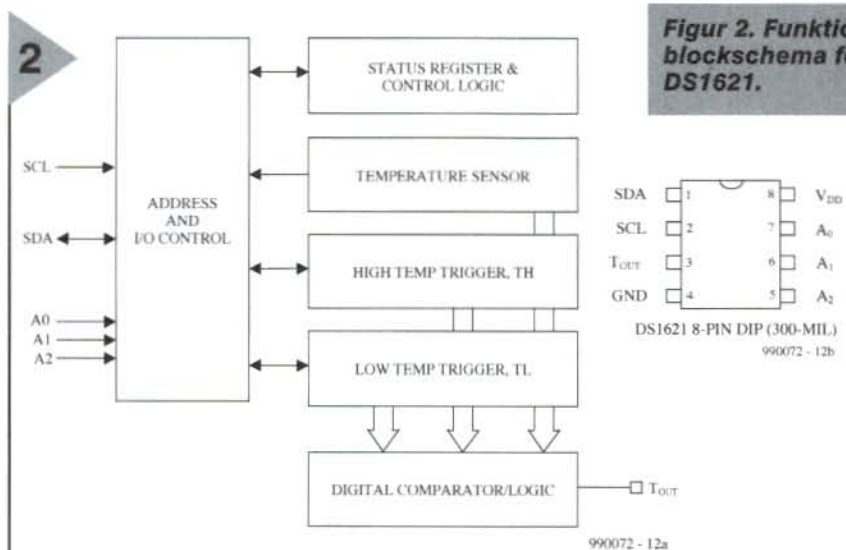
Temperatur i °C	Binära utgångsdata	hex
+125	01111101 00000000	7B00
+25	00011001 00000000	1900
+0.5	00000000 10000000	0080
0	00000000 00000000	0000
-0.5	11111111 10000000	FF80
-25	11100111 00000000	E700
-55	11001001 00000000	C900

Temperaturen kan sedan avgöras med hjälp av formeln:

$$\text{Temperature} = \text{TEMP\_READ} - 0.25 + \frac{(\text{COUNT\_PER\_C} - \text{COUNT\_REMAIN})}{\text{COUNT\_PER\_C}}$$

Som visas i det funktionsblockschema som ses i figur 2 inkluderar den intelligenta sensorn också en termostatfunktion. Två tröskelvärden kan programmeras för att skapa en hysteresfunktion. Om temperaturen når ett av de inställda värdena, stiger över det övre tröskelvärdet TH eller sjunker under det nedre tröskelvärdet TL, blir utgången

## TERMOSTAT



Figur 2. Funktionsblockschema för DS1621.

T<sub>OUT</sub> aktiv (hög).

Dessutom noterar DS1621 när en temperaturtröskel har överskridits och sätter motsvarande flaggor i Configuration/Status registret. Detta register inkluderar också annan information som styr arbetssättet för en speciell applikation. Registret konfigureras på följande sätt:

DONE	THF	TLF	NVB	1	0	POL	1SHOT
------	-----	-----	-----	---	---	-----	-------

Meningen med detta är följande:

**DONE**

High = omvandling färdig

Low = omvandling aktiv

**THF** (Temperature High Flag)

Sätt till 1 när temperaturen är lika med eller högre än tröskeln TH. Förblir aktiv tills återställning (reset).

**TLF** (Temperature Low Flag)

Sätt till 1 när temperaturen är lika med eller lägre än tröskeln TL. Förblir aktiv tills återställning (reset).

**NVB** (Non-volatile memory flag)

Aktiv (1) så fort en EEPROM-cell skrivs till. Detta kan ske varje 10 ms.

**POL** (Output Polarity bit)

Bestämmer om utgången är aktivt hög (1) eller aktivt låg (0). Icke-flyktig.

**1SHOT**

Om 1SHOT biten är hög utför DS1621 en temperaturomvandling när den tagit emot ett CONVERT T kommando. Annars arbetar den på normalt sätt där den kontinuerligt utför mätningar och omvandlingar. Denna bit är icke-flyktig.

## KOMMUNIKATION: FORM OCH INNEHÅLL

DS1621 arbetar som slav i ett tvåtråders bussystem med open-drain SDA och SCL linjer och med en klockhastighet på 100 eller 400 kHz. Kommunikationen med en mikrokontroller eller en PC sker med hjälp av det välkända I<sup>2</sup>C protokollet, vilket inte behöver förklaras närmare. Du kan hitta mer information angående timingen och andra tekniska specifikationer i databladet för DS1621 på Internetadressen:

[www.dalsemi.com/doccontrol/pdfs/pdfindex.html](http://www.dalsemi.com/doccontrol/pdfs/pdfindex.html).

Varje kommunikation behöver en startbit och avslutas med en stoppbit. DS1621 svarar på varje mottaget byte med en Acknowledge.

Startbiten följs alltid av ett adressbyte, vilket består av fyra bitar som är satta till värdet 1001 och tre bitar som bestäms av de nivåer som läggs till adressingångarna A<sub>2</sub>, A<sub>1</sub> och A<sub>0</sub>. Den

## KOMPONENTLISTA

### Motstånd:

R1 = 1kΩ

R2,R3 = 10kΩ

### Kondensator:

C1 = 100µF 16 V radial

### Halvledare:

D1 = LED, lågström

IC1 = DS1621 (Dallas Semiconductor)

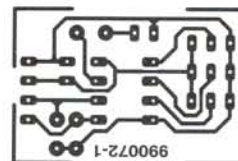
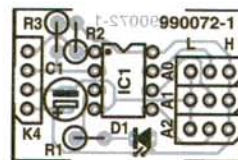
### Övrigt:

K1,K2,K3 = 3-pol stiftlist

K4 = 4-pol stiftlist

Diskett med mjukvara 996027-1

3



**Figur 3. Kretskortet får plats i en tändsticksask.**

sista biten bestämmer kommunikationsriktningen/R/W).

Adressbytet följs av ett kommando-byte. DS1621 känner igen de skriv och läskommandon som listas i tabell 2, inklusive deras koder. De två omvandlingskommandona är avslutade när adress och kommandobyten har sänts, men en eller två databyte följer kommandobytet för skrivkommandon. Med läskommandona är strukturen något mer komplicerad eftersom IC-n måste föras med ett andra adressbyte för att sätta R/W-biten till läs. Det ett eller två databyten kan läsas endast när detta har skett.

Dallas Semiconductors har en gratis demonstrationssats (Demonstration Kit) tillgänglig för DS1621. Denna övervakar sensor-IC:n i en temperaturloggningsapplikation. De tre temperaturvärdena (mätning och hysteres) visas med en upplösning på 0,1 °C eller 0,5 °C. Användaren kan ställa in tröskelvärdena för termostatutgången och inhämtningsintervallet i sekunder, aktivera direkt utgång av de mätta värdena till en fil och välja temperaturvisning i Fahrenheit eller Celsius eller som 2-D eller 3-D grafik. Mjukvaran stöder tre parallellportar och åtta hård-

varuadresser för DS1621.

Mjukvaran för denna sats, som kan köras under Windows 3.xx eller högre, kan laddas ner från denna Internetadress:

[ftp://ftp.dalsemi.com/pub/thermal/ds1621k.zip](http://ftp.dalsemi.com/pub/thermal/ds1621k.zip)

Om du inte har tillgång till Internet kan du köpa mjukvaran på diskett från oss. Ordernr 996027-1. Se Läsaerservice.

## HÅRDVARA

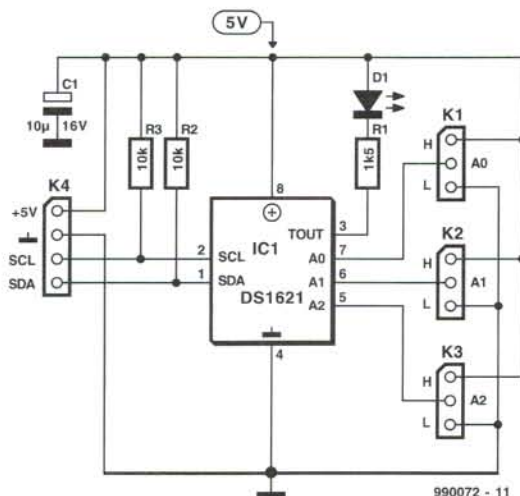
Som namnet anger inkluderar satsen ett kretskort med sensorn och alla övriga komponenter som behövs för anslutning till parallellporten på en PC. Elektronikhobbyister föredrar förmodligen att bygga den minimala hårdvaran själva i stället för att beställa satsen. Allt du behöver är det kretskort som visas i figur 3, de listade komponenterna och en varm lödkolv.

Kretsen på detta kort, som du bygger ihop på några minuter, motsvarar kretsschemat i figur 4, som endast behöver en kort förklaring.

Adressen för DS1621 ställs in med jumprarna K1-K3. Kortet kan enkelt anslutas till I<sup>2</sup>C-interface hos ett mikrokontrollersystem via K4.

Att ansluta kortet till parallellporten på en PC, för att kunna köra Dallaspro-

4



**Figur 4. Inga överraskningar här: LED-en indikerar att ett tröskelvärde överskridits.**

**Tabell 2. DS1621 kommandoset**

Kommando	Beskrivning	Värde (hex)	Följande byte
<i>Temperaturmätningsskommandon</i>			
Read Temperature	Läser det senaste temperaturvärdet från temperaturregistret	AA	läser två databyte
Read Counter	Läser resten i tidsfönsterräknaren	A8	läser ett databyte
Read Slope	Läser värdet i Slope Accumulator	A9	läser ett databyte
Start Convert T	Startar en temperaturomvandling	EE	inga data
Stop Convert T	Stoppar en temperaturomvandling	22	inga data
<i>Kommandon för termostatfunktion</i>			
Access TH	Läser/skriver övre tröskelvärdet från/till TH-registret	A1	läser/skriver två databyte
Access TL	Läser/skriver nedre tröskelvärdet från/till TL-registret	A2	läser/skriver två databyte
Access Config	Läser/skriver data från/till Configuration-registret	A1	läser/skriver ett databyte

grammet, är dock en helt annan sak. För detta behöver du ett parallellt I2C-interface, som det vi publicerade i nummer 2/96, med tillhörande mjukvara (ordernr 950063-C)

(990072-1)

## RÄTTELSE & uppdateringar

### I2C interface för skrivarporten

3/1999,

Om mjukvaran producerar felmeddelanden trots att hårdvaran fungerar korrekt skall stiften 10 (ACK) och 13 (SLCT) hos K1 anslutas till jord.

### EEDTS Pro

Serien, 6/1999 – 9/1999, 1/2000 (980055)

För byggare och användare har konstruktören, S. de Vries, satt upp en speciell hemsida på:

[www.gironet.nl/home/editspro](http://www.gironet.nl/home/editspro)  
Mr. De Vries kan också nås via email på adressen: [editspro@gironet.nl](mailto:editspro@gironet.nl)

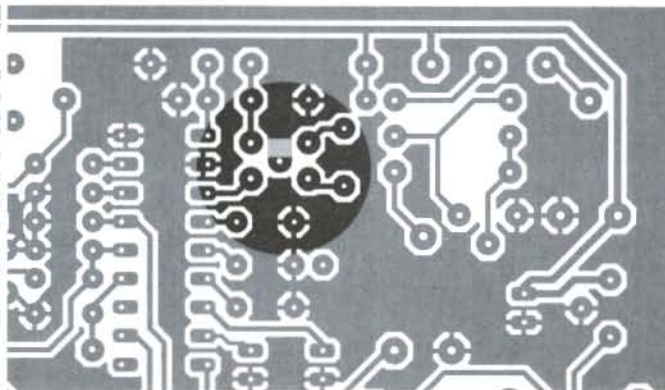
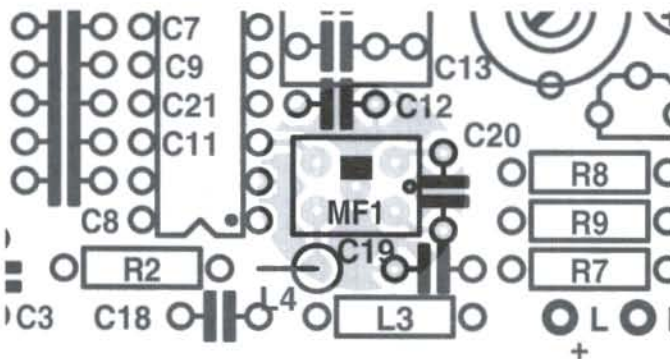
### Fattigmans radio

9/1999, (990068-1)

På kretskortet har mittstiften på filtret MF11 ingen jordanslutning. Detta orsakar otillräcklig känslighet. Felet kan korrigeras genom att löda fast en lite bit kopplingstråd såsom visas nedan.

Om du använder en konden-

sator med raster på 7,5 mm i position C13 måste jordanslutningen göras med en bit kopplingstråd.



# Nya audioformat

*hög upplösning och många egenskaper*

Digital audio har blivit vardagsmat. Mycket har skett sedan digital audio introducerades på konsumentmarknaden under 1970-talet. Tekniken har förbättrats och media-kapaciteten har ökat betydligt, delvis beroende på utvecklingen av nya generationer halvledarlarar.

Det finns många människor idag som har svårt att föreställa sig hur en skivaffär såg ut för bara 20 år sedan. På anmärkningsvärt kort tid har vinylskivorna fått ge plats åt ett allt större sortiment med CD-skivor, och på senare tid också DVD. Vinylskivorna (EP och LP) har nu i stort sett försvunnit från jordens yta tillsammans med den utrustning som användes för att spela dem.

## EN VÄRLD FULL AV FÖRÄNDRINGAR

Efter de första initiativen att digitalisera audioinformation för konsumentmarknaden under 1970-talet, och introduktionen av den första audio-CD:n för över femton år sedan, har mycket arbete lagts ner på att utveckla efterföljare till CD:n, som kan lyfta kvalitén på audioinformationen till ännu högre höjder. Även om en audio-CD representerar den ultimata musikupplevelsen för många av oss, så finns det vissa tekniska svagheter i dess design. Med den nya utvecklingen kommer denna situation definitivt att ändras.

## LITE HISTORIA

Utvecklingen av audio-CD under 70 och 80-talen i förra århundradet var på många sätt ett enormt tekniskt genombrott. Traditionell analog inspelningsteknik, som då bestod av graverade spår i vinylskivor, eller magnetspår på band,



övergavs helt och hållet.

Utvecklingen av A/D och D/A omvandlare var en svår uppgift under den första tiden för digital audio. Ett multibit PCM kodningssystem med en relativt låg samplingshastighet valdes för digitaliseringen av audiosignaler. Baserad på en audiobandbredd på 20 kHz sattes samplingshastigheten till 44.1 kHz, med en upplösning på 16 bitar. På den tiden var detta specifikationer av allra högsta klass och tillät inspelningar av hög kvalitet.

## ÄNDRINGAR PÅ FLERA FRONTER

Den digitala inspelningen av audioformat var inte den enda uppfinningen. Användningen av CD:n som ett lagringsmedium för data var också ett tekniskt genombrott. För första gången användes en halvledarlaser i en konsumentprodukt för att läsa ett digitalt optiskt lagringsmedium med hög hastighet. Numera kan man knappast tänka sig ett liv utan audio-CD eller CD-ROM.

Tillverkarna av elektroniska konsumentprodukter har nu funnit att tiden är mogen för att höja kvalitén på ljudåtergivningen till nya nivåer. Tyvärr har man inte kunnat enas om en gemensam standard. Två nya audiomedier har introducerats på konsumentmarknaden där de kommer att konkurrera med varandra under de kommande åren, Super Audio CD och DVD-Audio.

## MEDIA MED HÖG DENSITET

Alla förbättringar av kvalitén hos digital audio går hand i hand med ökningen av den mängd data som måste lagras. Med tanke på att dimensionerna på en CD-skiva, 120 mm, mer eller mindre blivit industristandard var det otänkbart att en fysiskt större skiva (och därmed ett nytt format) skulle användas. Dessutom är kompatibilitet med redan befintliga apparater och/eller mjukvara ett krav om ett nytt system överhuvudtaget skall accepteras. Detta betyder att lagringskapaciteten hos det optiska mediet (CD-skivan) måste förbättras. Det tog lång tid innan industrin kunde komma överens när det gällde en ny generation CD, men den kom till slut i form av DVD (Digital Versatile Disc).

## DVD: ETT KONCEPT, FYRA FORMAT

Även om en DVD ser ut på samma sätt som en normal CD vid första anblicken så är den betydligt mer sofistikerad. För det första så används en annan typ av laser för att läsa en DVD. Denna laser använder ljus med en kortare våglängd (635 nm till 650 nm i stället för 780 nm). Detta gör att den ljuspunkt som används för att läsa en DVD är betydligt mindre. Detta betyder i sin tur att längden på de digitala data 'groparna' kan minskas från 0,83 till 0,4 mikron. Samtidigt kan spåren göras smalare och spåren kan också läggas närmare varandra. Avståndet mellan spåren är således bara 0,74 mikron jämfört med 1,6 mikron hos en normal CD. Sammantaget ökar detta lagringskapaciteten från 640 MB till 4,7 GB. Även detta är emellertid otillräckligt för vissa applikationer. Följaktligen finns det fyra format definierade för DVD: DVD-5, DVD-9, DVD-10 och DVD-18. Förutom dessa format finns det också övernskomelse angående (åter)inspelningsbara DVD, men detta faller utanför denna artikel.

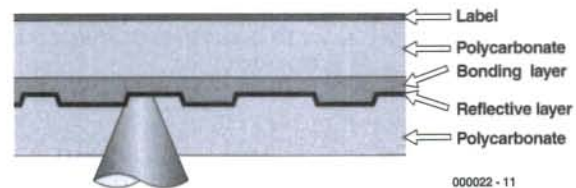
Den minsta DVD:n har en lagringskapacitet på 4,7 GB, medan den största är i det närmaste fyra gånger så stor (17,1 GB). Alla DVD använder två substrat (transparenta plastlager som bär den digitala informationen) som är 0,6 mm tjocka. I en traditionell CD finns bara ett substrat som är 1,2 mm tjockt. Eftersom den totala kapaciteten hos ett substratlager är 4,7 GB för alla DVD, betyder detta att den maximala kapaciteten endast kan

uppnås genom att använda två informationslager ovanpå varandra på varje sida av skivan, och sedan använda båda sidorna av skivan. Låt oss titta lite närmare på alla fyra formaten.

### DVD-5 (4.7 GB)

#### Enkelsidigt, ett lager

Detta format är den mest elementära medlemmen av DVD-familjen. Skivan har en lagringskapacitet på 4,7 GB. Endast ett av de två substraten har ett informationslager. De två substraten är dock kombinerade (hoplimmade) till ett enda substrat som är 1,2 mm tjockt.

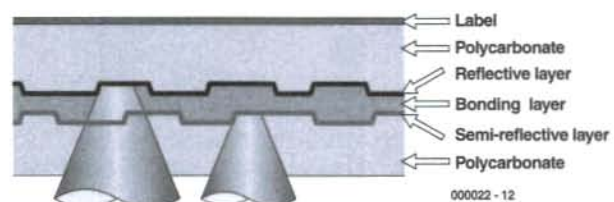


Figur 1. Konstruktion av en DVD-skiva med ett optiskt datalager (DVD-5).

### DVD-9 (8.5 GB)

#### Enkelsidigt, två lager

Denna DVD-variant är enkelsidig, men den använder två lager på den ena sidan. Detta ger en kapacitet på 8,5 GB, vilket är något mindre än två gånger 4,7 GB. Skillnaden ligger i de åtgärder som tillverkarna måste göra för att förbättra läsbarheten hos det andra lagret. Det resulterande bitmönstret är således 10% större än det hos DVD-5 och DVD-10 formaten. Varje lager är placerat på ett 0,6 mm tjockt substrat, där ett av lagren är halvtransparent (guldfärgat). De två lagren är hoplimmade med ett transparent (optiskt neutralt) lim så att det blir ett homogent lager. Lagringskapaciteten är mer än tillräcklig för en långfilm av hög kvalitet (MPEG2-format) och har också utrymme för ytterligare data, inklusive åtta spår med audioinformation.



Figur 2. Konstruktion av en DVD-skiva med två optiska datalager på en sida (DVD-9).

### DVD-10 (9.4 GB)

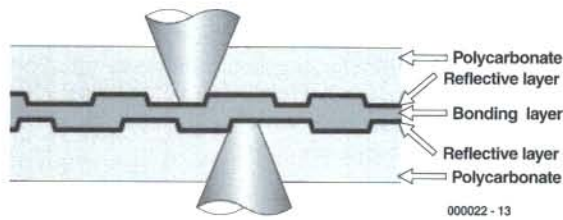
#### Tvåsidigt, ett lager per sida

Denna DVD-variant använder båda sidorna på skivan, med

Table 1. Comparative physical specifications of DVDs and standard CDs.

Specification	CD	DVD	DVD
Number of layers	single	single	double
Substrate thickness	1.2 mm	0.6 mm	0.6 mm
Number of substrates per disc	1	2	2
Track separation	1.6 micron	0.74 micron	0.74 micron
Minimum pit length	0.83 micron	0.4 micron	0.44 micron
Reading speed (m/s)	1.3	3.49	3.84
Laser wavelength (nm)	780	635/650	635/650
Numerical aperture (NA)	0.45	0.6	0.6
Modulation	EFM	8-to-16	8-to-16

ett digitalt lager med en kapacitet på 4,7 GB på varje sida. För att kunna spela båda sidorna måste CD:n vändas. För närvarande måste man göra detta manuellt, men det är teoretiskt möjligt att tillverka CD-spelare som kan läsa båda sidorna på skivan utan att den behöver vändas för hand.



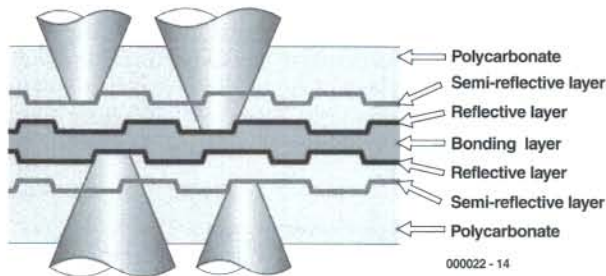
**Figur 3. Konstruktion av en DVD-skiva med ett optiskt datalager på varje sida (DVD-10).**

#### DVD-18 (17.1 GB)

##### Tvåsidig, två lager per sida

DVD-18 formatet är den variant som har den största lagringskapaciteten. Det finns ett dubbelt datalager på varje sida av skivan och varje skiva innehåller således totalt fyra datalager.

Med detta format måste de två lagren på varje sida placeras på ett enda substrat av polycarbonat. En fotoprocess används för att producera det lager som ligger i mitten av substratet. När det gäller tillverkningsprocessen betyder detta att det också måste finnas ett produktionssystem som för över bitmönstret till det fotokänsliga lagret, förutom den vanliga DVD-pressen. Tillverkningen av DVD enligt denna specifikation är mycket svår, och för närvarande mycket dyr.

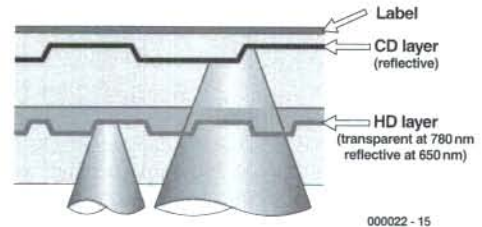


**Figur 4. Konstruktion av en DVD-skiva med två optiska datalager på varje sida. (DVD-18). Tillverkningen av detta format är för närvarande mycket svårt och därmed också mycket dyrt.**

### SUPER AUDIO CD: BÄSTA FORMATET?

Som vi redan nämnt har musikindustrin också introducerat Super Audio CD förutom DVD. Som ett data-medium är detta en blandning av en DVD och en normal CD. Utvecklarna av detta koncept, Philips och Sony, behövde ett lagringsmedia med DVD-kapacitet för att uppnå de önskade förbättringarna i kvaliteten på audioinformationen, men de ville absolut också behålla kompatibiliteten med traditionella CD. Super Audio CD (SACD) kan således användas tillsammans med alla befintliga CD-spelare såväl som med nya SACD-spelare. Det är dock bara i de senare fallen som den förbättrade kvaliteten och de nya egenskaperna finns tillgängliga. Här används en enkelsidig

skiva med två datalager. Det informationslager som en vanlig CD-spelare läser ligger på standarddjupet 1,2 mm, medan ett speciellt lager med hög lagringskapacitet är placerat på en djup av 0,6 mm. Lasern hos en standard CD-spelare, som arbetar med en våglängd på 780 nm, fokuseras på det optiska lager som ligger på 1,2 mm djup och ser således igenom högdensitetslagret på 0,6 mm. Lasern hos SACD-spelaren, med en våglängd på 650 nm, fokuserar på högdensitetslagret och ser inte 'standard-CD' lagret.

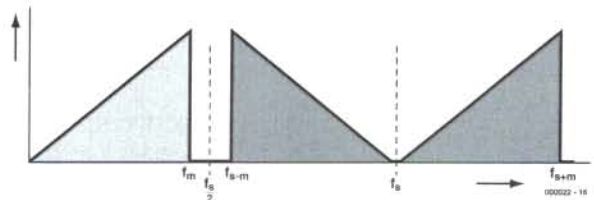


**Figur 5. Konstruktionen av en Super Audio CD. Det finns två olika optiska datalager på skivans ena sida. Vilket lager som läses beror på vilken typ av laser som används.**

### MER INFORMATION PÅ SKIVAN

Upp till nu har vi beskrivit hur industrin har ökat den mängd data som kan lagras på en skiva med en diameter på 120 mm från 640 MB till maximalt 17,1 GB. För att nu kunna utnyttja denna lagringskapacitet optimalt, och också låta användaren få fördelar av detta, så har även inspelnings och kodningstekniken förbättrats.

Som vi redan noterat är en original-CD baserad på multibit



**Figur 6. När en audiosignal samplas så skapas det en spegelbild av det samplade frekvensspektrat ( $f_m$ ) på varje sida om samplingsfrekvensen ( $f_s$ ).**



**En kompakt DVD-spelare för återgivning av audio-CD, video-DVD och speciella video-CD som innehåller audiospår.**

PCM-kodning med en upplösning på 16 bitar. Numera har den digitala tekniken avancerat så mycket att hastigheten i sig själv inte är något problem. Philips och Sony, som tillsammans uppfann både standard-CD och Super Audio CD, har under de senaste åren utvecklat ett nytt inspelnings och kodningssystem som kallas Direct Stream Digital (DSD). Detta arbetar med en upplösning på 1 bit och en samplingshastighet på 2.8224 MHz ( $64 \cdot f_s$ ). Figur 6 visar hur audiosignalen digitaliseras. När en audiosignal ( $f_m$ ) samplas med samplingsfrekvensen  $f_s$ , genereras det 'spegelbilder' av audiosignalen, en på varje sida om samplingsfrekvensen. För att förhindra att dessa två spegelbilder överlappar varandra så måste samlingsfrekvensen vara så hög som möjligt. Med en standard audio-CD är bandbredden på audiospektrat 20 kHz och samplingshastigheten är 44.1 kHz. Originalsignalen och den nedre delen av spegelsignalen ligger således mycket nära varandra. Kraftiga filter som skarpt klipper audiosignalen precis ovanför 20 kHz är därför absolut nödvändiga. Sådana filter kan också påverka signalen inom audiospektrat. Om en hög samlingsfrekvens ( $64 \cdot f_s$ ) används i stället, så försvinner filterproblemen eftersom de resulterande spegelfrekvenserna ligger i en helt annat del av spektrat än audiofrekvenserna. Dessutom kan upplösningen hos A/D-omvandlaren minskas drastiskt, i detta fall till 1 bit. För varje sampligt kollar omvandlaren om amplituden hos den nya samplingen är större eller mindre än den hos den föregående samplingen. Denna typ av digitalisering, som kallas för delta-sigmaomvandling, bildar hjärtat i en DSD-omvandlare. Om omvandlingsfelet (kvantiseringsfelet) hos föregående omvandling tas med i beräkningen vid varje ny omvandling (vilket kallas noise shaping) kan kvaliteten hos samlingsprocessen förbättras ytterligare. Figur 7 visar denna process. Resultatet blir en digitaliserad audiosignal med en bandbredd på 100 kHz (fem gånger större än en normal audio-CD) och med ett dynamiskt område på 120 dB, jämfört med 96 dB hos en standard audio-CD. För att uppnå detta resultat behövs fyra gånger så mycket data jämfört med en standard audio-CD. En normal CD-inspelning som tar upp 600 MB behöver således ungefär 2,4 GB i DSD-formatet.

Audiospecifikationerna för DSD är så bra att det är bättre att försöka göra bättre inspelningar. Philips och Sony har rekommenderat alla inspelningsstudior att göra sina masterinspelningar i DSD-formatet (även om det faktum att DSD-formatet introduceras först i mitten av 1999 betyder att det är mycket få studior som har lämplig utrustning, men det är en helt annan sak).

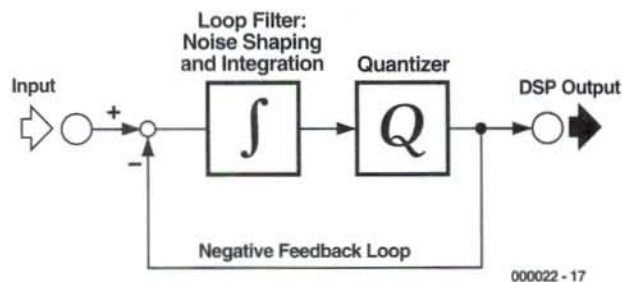
För att garantera kompatibilitet med befintlig PCM-teknik har Sony utvecklat vad som kallas Super Bit Mapping Direct. Figur 8 visar hur alla standard PCM-format kan tas från en DSD-signal med denna teknik. Det är möjligt att omvandla signalen till PCM-formatet med olika upplösning (16, 18, 20 eller till och med 24 bitar). Om andra format skulle dyka upp i framtiden kan dessa också tas från DSD-formatet.

Det finns en allmän uppfattning att audioindustrin skall kunna använda sig av DSD-formatet under de närmaste tjugo åren minst.

DSD är den bästa möjliga inspelningstekniken och data som lagras i högdensitetslagret på en Super Audio CD är i DSD-format. Eftersom en inspelning med maximal längd (74 minuter) inte utbytjar den fulla lagringskapaciteten så finns det utrymme kvar för ytterligare information. Här kan man tänka sig flerkanalsinspelning, grafisk information och text. Även om Super Audio CD-spelare för närvarande ligger utanför vad den genomsnittlige konsumenten harråd med, så räknar man med att priset på DSD-spelare kommer att falla så mycket under de närmaste åren att de kommer att byggas in i fler och fler apparater.

## DVD - AUDIO

Förutom Super Audio CD så har man också under de senaste

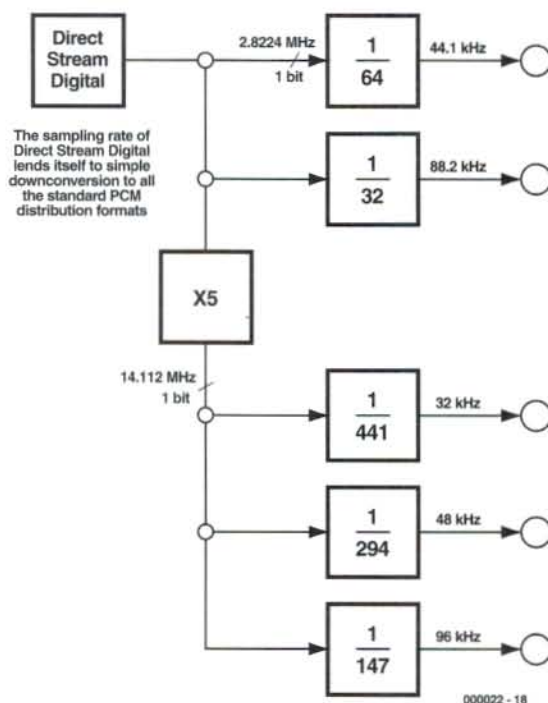


Figur 7. En DSD-omvandlare använder en 1-bits delta-sigmaomvandlare. Om korrekt återkoppling (feedback) används faller bruset och signalkvaliteten förbättras.

åren arbetat på en audiostandard som baseras på DVD. De resulterade formaten definierades slutligen i en standard i april 1999.

Denna standard baseras på två DVD-format, vilka kallas DVD-Audio och DVD-AudioV. Dessa två format kompletterar DVD-Video, vilket är det format som utvecklats för distribution av film. Framtida DVD-spelare skall kunna klara av alla DVD-formaten och de skall också kunna användas i datorer.

DVD-Audio erbjuder användaren nya möjligheter, inklusive förbättrad ljudkvalitet (tack vare ökad samplingsfrekvens och högre upplösning), surround sound (flerkanalsljud), längre speltid samt kompletterande funktioner. Lagringskapaciteten hos DVD-Audio med två lager räcker till för minst två timmar högkvalitets surroundljud eller fyra timmar stereoljud. Specifikationerna för en skiva med bara ett lager är ungefär hälften av detta. För att kunna öka lagringskapaciteten har en förlustfri komprimeringsmetod som kallas Meridian Lossless Packing utvecklats för DVD-Audio. Detta ökar lagringskapaciteten så mycket att minst 74 minuter audio kan spelas in på en enda DVD-sida. Hårdvaran i DVD-spelaren kan hantera denna komprimering utan att belasta processorn



Figur 8. Super Bit Mapping Direct tekniken kan användas för att ta vilket önskat PCM-format som helst från DSD-formatet, både när det gäller sampling och upplösning.



**Än så länge är det bara Sony som har SACD-spelare. De är fortfarande ganska dyra, den på bilden kostar runt 45000:-**

alltför mycket.

Förutom dessa attraktiva specifikationer har man vidtagit en hel del åtgärder både när det gäller DVD-Audio och Super Audio CD för att försvåra kopiering. Den genomsnittlige brukaren kan inte (för närvarande) göra kopior med samma kvalitet som originalet. Piratkopierare kommer att få svårt att tillverka kopior som ligger så nära originalet i kvalitet att köparen inte märker någon skillnad. De speciella vattenstämplar som kan läggas på datasidan är mycket svåra att kopiera.

Med tanke på det faktum att kopieringsskyddet för DVD-videoskivor med CSS (Content Scrambling System) nyligen knäckts och att crackerprogrammet DeCSS distribuerades fritt över Internet, så är det dock förmodligen bara en tidsfråga hur länge dessa säkerhetsåtgärder kommer att vara effektiva. En variation av CSS används också för DVD-Audio.

## DVD-AUDIO I PRAKTIKEN

Den maximala datahastigheten för DVD-Audio är 9,6 Mbit/s. Detta betyder att den maximala samplingshastigheten för flerkanals audio är begränsad till 96 kHz. För att kunna använda den tillgängliga bandbredden på bästa möjliga sätt är det möjligt att ha olika specifikationer för de olika kanalerna. Till exempel kan vänster, höger och mittkanalen samplas med 96 kHz med 24-bitars PCM-upplösning, medan de två bakre kanalerna får 'nöja sig' med en samplingshastighet på 48 kHz och 16-bitars upplösning. Andra kodningar kan användas förutom PCM, som Dolby Digital (AC3), Digital Theatre Sound (DTS), MPEG1 stereo eller MPEG2 flerkanals audio (även om det idag verkar som om MPEG audioformatet framstår som en förlorare). DVD-AudioV är en kompletterande standard som kombinerar audio med video så att audiodelen kan spelas av på en speciell DVD-audiospelare och

videodelen på en vanlig DVD video-spelare.

## STANDARD DVD-VIDEO SOM ETT AUDIOMEDIUM

Som ett komplement till dessa format är det intressant att notera att en DVD-Videoskiva också kan ha åtta ljudkanaler. På detta sätt kan skivtillverkarna välja mellan flera monokanaler, till exempel för tal, eller multikanaler för musik. Minst tre flerkanals audioinspelningar kan kombineras med en videosignal.

Under tiden har flera musikproducenter (som Denon och Chesky) tillverkat video-DVD med bara två audiospår, med 96 kHz, 24-bitars audio av mycket hög kvalitet. Så länge som det inte finns någon hårdvara eller mjukvara tillgänglig för DVD-Audio kan dessa speciella inspelningar betraktas som ett excellent alternativ.

Efter denna summering av de nya audioformaten så måste vi fråga oss om den genomsnittlige konsumenten egentligen är intresserad av detta. DVD-Video kan man betrakta som ett framgångsrikt nytt format, men detta ger också betydligt bättre bild och ljudkvalitet än videoband. Det är däremot tveksamt om de nya audioformaten kommer att väcka någon större entusiasm. Å andra sidan trodde man inte att datorer skulle ha någon framtid heller..

(000022-1)

### Primära specifikationer för DVD-Audio och DVD-Video

	DVD-Audio	DVD-Video
Kodning	linjär eller komprimerad PCM	linjär PCM, Dolby AC-3 eller DTS
Samplingsfrekvens	44.1/48/88.2/96/176.4/192 kHz	48/96 kHz
Upplösning	16/20/24 bitar	16/20/24 bits
Antal kanaler	6 (96 kHz maximum) 2 (176.4/192 kHz maximum)	8
Maximal bithastighet	9.6 Mb/s (linjär PCM)	6.144 Mb/s

Vad som skiljer detta kort från alla andra DAQ (Mät & Styr) och kontrollkort för PC-ns serieport, är sättet på vilket det kommunicerar med datorn. Många av de kort som du kanske sett inkluderar en mikro-kontroller, eller ett UART-chip för seriekommunikationen. Det här kortet har inget sådant chip eftersom det utnyttjar direkt access till UART-registren för aktivering av seriell till parallell omvandling

Design av George Vastianos

# 32-kanals digitalt ingångskort för PC-ns serieport

## länkar din PC till världen

### Huvudspecifikationer

Antal ingångar:	32
Typ av ingångar:	TTL-kompatibla ("0" = 0 V, "1" = 5 V)
Anslutning:	Serieport
Kommunikation:	Direkt access till UART-registren
Strömförsörjning:	9-15 V DC

Området datainhämtning och styrning är inte någonting nytt inom den elektroniska vetenskapen. Detta började direkt efter det att halvledartekniken lyckades tillverka den första mikroprocessorn i en IC-kapsel. Det var på den tiden som datorerna gjorde sitt intåg i industrin och, efter det, i våra hem.

Mät & Styrteknik handlar om behandling av data och styrning via datorer. Målet för ett Mät & Styrsystem är att tillåta att datorn kommunicerar med, och i viss mån styr, världen 'utanför'. Applikationer för Mät & Styrsystem har att göra med industrin där man genom att använda lämpliga ingångs/utgångskort

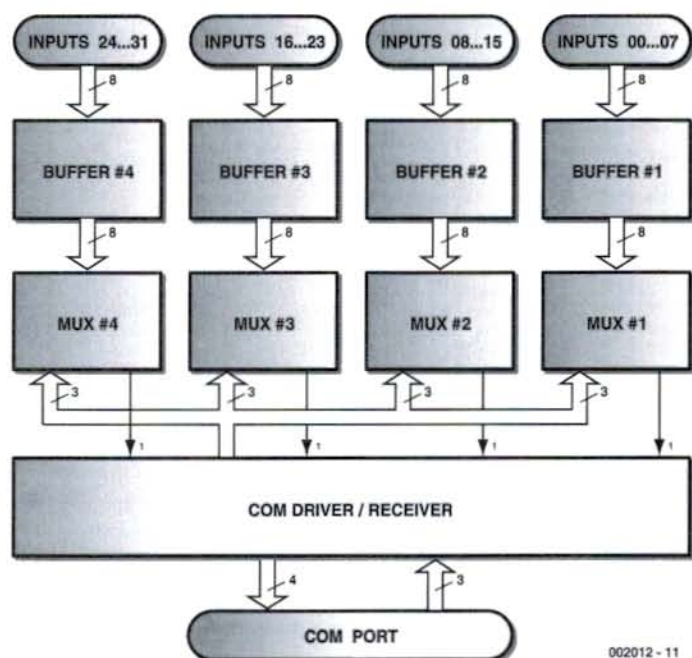
kan automatisera och styra system och på så sätt lösa komplicerade uppgifter. Ett Styr & Mätkort kan ha enbart ingångar, enbart utgångar eller en kombination av dessa. En annan egenskap hos dessa kort är sättet på vilket de är anslutna till datorn. Många kort har designats för att installeras på en av de lokala bussarna på huvudkortet (ISA, EISA, PCI etc.) eller för anslutning till de tillgängliga portarna på datorn (parallell, seriell, game, tangentbordsport).

Det projekt vi beskriver i denna artikel är ett kort med 32 digitala ingångar (32 Channel D/I Card) för extern anslutning till den seriella (RS232) porten på din PC.

### Om serieporten

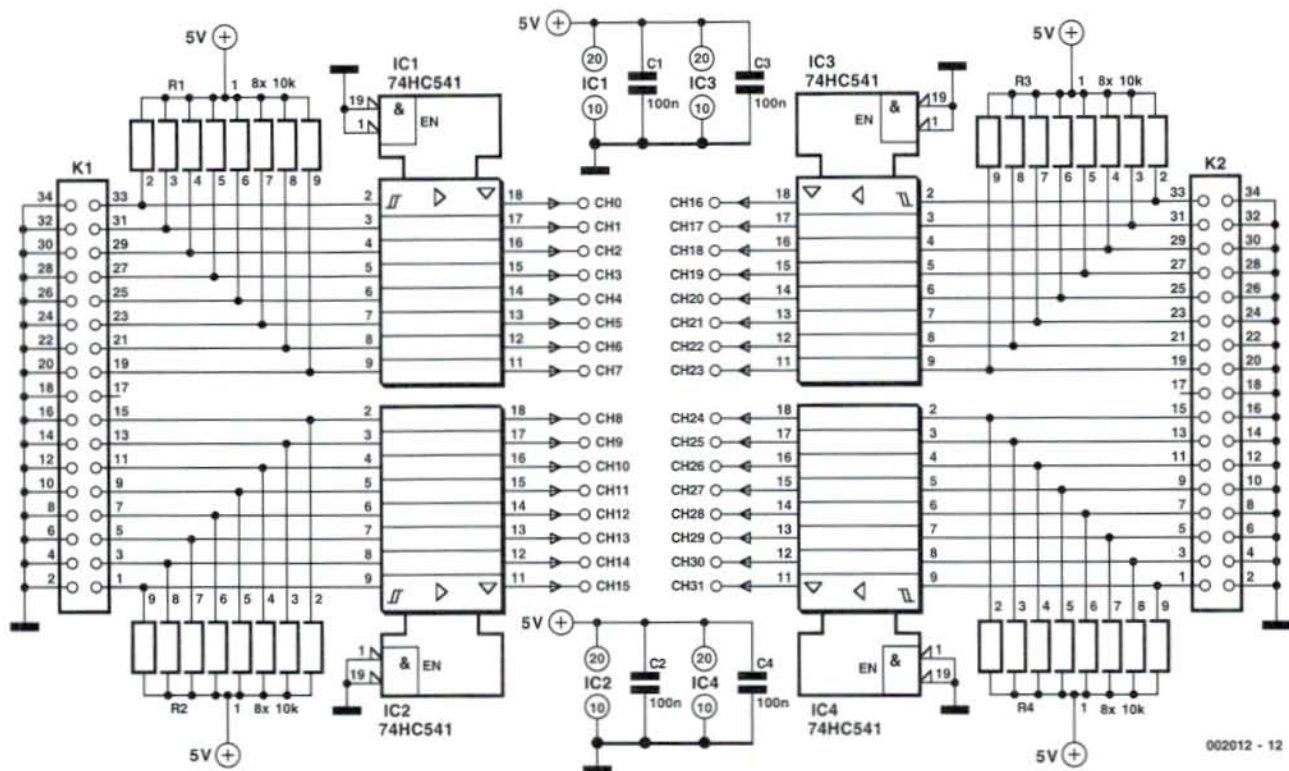
Seriella portar används i huvudsak för kommunikation mellan datorer eller för kommunikation mellan en dator och yttre utrustning som ett modem eller en mus. Kontrollen som sitter i hjärtat på denna port är i stort sett alltid ett UART-chip (Universal Asynchronous Receiver Transmitter) som sitter på moderkortet. Detta chip arbetar som en seriell-till-parallell och parallell-till-seriell adapter. En dator kan ha en till fyra serieportar (COM1-COM4) där varje port tar upp åtta platser i sitt I/O-område. Se tabell 1 för de relevanta detaljerna.

De baslinjer som används av UART i seriell kommunikation för sändning och



002012 - 11

Figur 1. Blockschemat för 32-kanals D/I-kort.



Figur 2. Kretsschema för Bufferenheten.

mottagning kallas för TxD och RxD. Det finns också en grupp med extra linjer (DCD, DSR, RTS, CTS, DTR, RI) som används för att etablera olika typer av seriell kommunikation. Även om vissa av dessa extra linjer fungerar som ingångar och andra som utgångar så kan var och en (med undantag för RxD) styras med hjälp av en bit i UART-registret. I tabell 2 summeras interfacets stiftanslutningar och systemets I/O-adresser. De spänningsnivåer som används på serieporten skiljer sig från TTL-nivåerna. På RS232-språk representeras en logisk '1' av en spänning på -12 V och en logisk '0' av en spänning på +12 V.

## Hårdvaran

I kretsens blockschema, figur 1, har de tillgängliga ingångarna delats i fyra grupper om åtta ingångar (00-07, 08-15, 16-

23, 24-31) och de går alla in i Buffer. Därefter går linjerna in i MUX (Multiplexers Unit), där endast en linje i varje grupp väljs. De fyra valda ingångarna skickas vidare till COM Driver/Receiver (där de omvandlas från TTL-kompatibla till RS232-kompatibla) och kommer fram till fyra seriella portingångar (CTS, DSR, RI, DCD). För att välja ingångarna använder vi de tre utgångarna på serieporten (TxD, DTR, RTS). När de har

passerat COM Driver/Receiver och anpassats från RS232-kompatibla till TTL-kompatibla) kommer utgångssignalerna fram till adressingångarna på MUX.

## Buffer

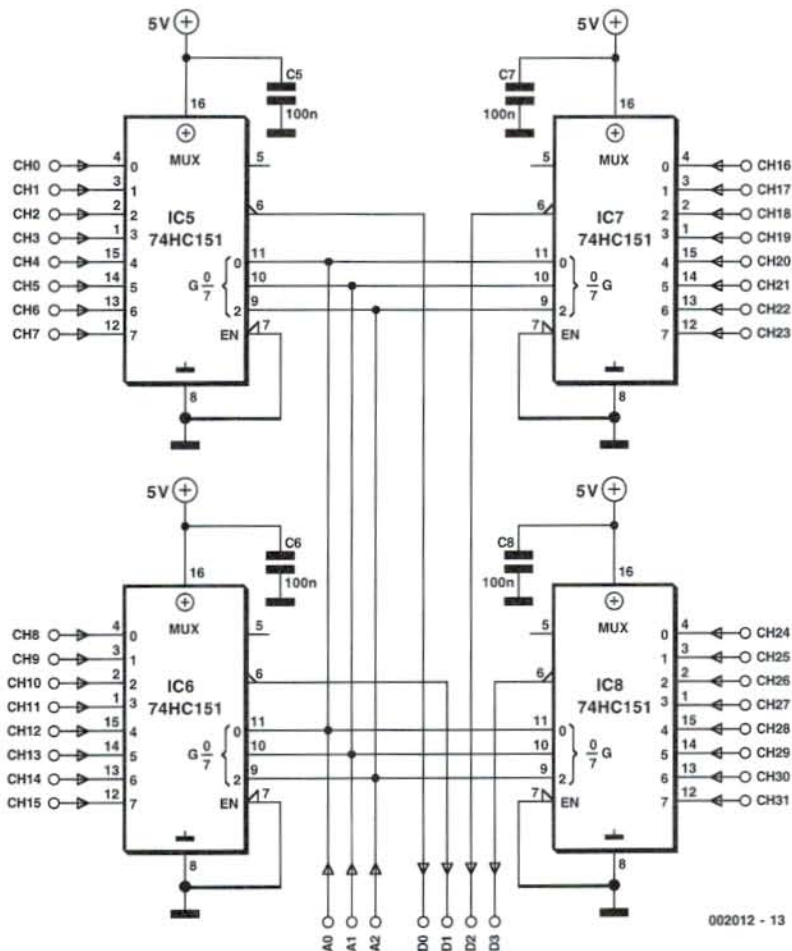
I schemat, figur 2, har de 32 ingångarna delats i två grupper om 16 (för enkel kretskortsdesign) och kommer in i kretsen via kontaktarna K1 och K2. Hur in-

Tabell 1.

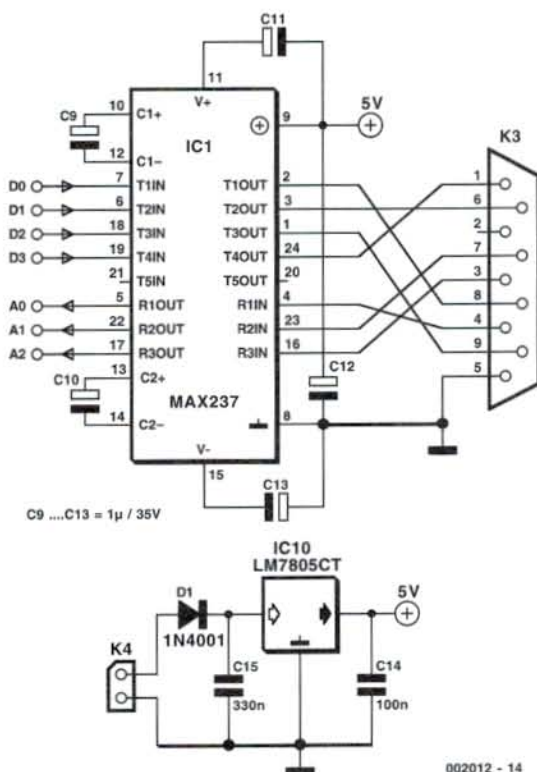
Transmit/Receive Buffer	3F8h	2F8h	3E8h	2E8h
Interrupt Enable Register	3F9h	2F9h	3E9h	2E9h
Interrupt Identification Register	3FAh	2FAh	3EAh	2EAh
Line Control Register	3FBh	2FBh	3EBh	2EBh
Modem Control Register	3FCh	2FCh	3ECh	2ECh
Line Status Register	3FDh	2FDh	3EDh	2EDh

Tabell 2.

Stift	Stiftnr på 25-pol kontakt	Stiftnr på 9-pol kontakt	COM1	COM2	COM3	COM4	Bit	I/O
TxD	2	3	3FBh	2FBh	3EBh	2EBh	6	O
DTR	20	4	3FCh	2FCh	3ECh	2ECh	0	O
RTS	4	7	3FCh	2FCh	3ECh	2ECh	1	O
CTS	5	8	3FEh	2FEh	3EEh	2EEh	4	I
DSR	6	6	3FEh	2FEh	3EEh	2EEh	5	I
RI	22	9	3FEh	2FEh	3EEh	2EEh	6	I
DCD	8	1	3FEh	2FEh	3EEh	2EEh	7	I



Figur 3. Krettschema för MUX-enheten.



Figur 4. Krettschema för COM Driver/Receiver och strömförsörjningen.

gångarna ligger på kontakterna kan ses i tabell 3.

Alla ingångar är försedda med pull-upmotstånd R1-R4 för att ge terminering i den händelse en eller flera ingångar inte är anslutna. Buffrarna är IC1-IC4 (74HC541). De fyra kondensatorerna (100 nF) används för att förbättra kretsens stabilitet. Kontrollgångarna Output Enable (benen 1 och 19) hos 74HC541 är kopplade till jord så att buffrarna arbetar kontinuerligt.

### MUX

Som visas i figur 3 är multiplexrarna i verkligheten fyra 74HC151 (IC5-IC8). Fyra kondensatorer, C5-C8 (100 nF) har lagts till för att avkoppla matningsspänningen. Kontrollgången Output Enable (ben 7 på varje 74HC151) är kopplad till jord så att varje multiplexer arbetar kontinuerligt. Adressgångarna A, B och C hos alla 74HC151 är hopkopplade för att implementera multiplexing av alla 32 ingångarna.

### COM Driver/Receiver

Den sista enheten inkluderar en spänningsregulator så att kortet inte behöver någon stabiliserad spänningskälla för att fungera. I krettschemat över COM Driver/Receiver enheten (figur 4) fungerar IC9 (MAX237) som en RS232 drivare/mottagare. Den har tre kanaler som omvandlar från RS232 till TTL och fem kanaler som omvandlar från TTL till RS232. Fem kondensatorer (C9-C13, 1  $\mu$ F, 35V max. arbetsspänning) gör att MAX237 kan utföra en spänningsdubbling (så att drivsektionen kan lämna en spänning på 10 V för de fem RS232-utgångarna från en matningsspänning på 5 V). Inuti MAX237 har varje kanal en inverterare. För att komma förbi detta problem använder vi de fyra inverterade utgångarna hos de fyra 74HC151-orna (ben 6). Efter invertering två gånger finns det alltså ingen invertering alls. IC10, en 7805 tillsammans med kondensatorerna C14 och C15 tar ner matningsspänningen till 5 V. Dioden D1 skyddar kretsen mot omvänd polaritet.

### Styrjukvaran

Mjukvaran för kommunikationen med kortet utvecklades i QBasic. Kommunikationsrutinen kallas CARD32DI och dess källkod kan hittas i Lista 1. Om du anropar denna rutin (från något program skrivet i QBasic) så följ denna syntax:

```
CALL CARD32DI (COMADDRESS, CHANNEL (), DATA0, DATA1, DATA2, DATA3)
```



Figur 5. Skärmdump av demo-programmet i funktion.

Där:

**COMADDRESS:** Heltalsvariabel som (före anrop) måste innehålla basadressen för serieporten. Giltiga värden för denna variabel är: &H3F8 (för COM1), &H2F8 (för COM2), &H3E8 (för COM3), &H2E8 (för COM4).

**CHANNEL ():** Matris heltalsvariabel (med pekare från 0 till 31) som (efter anrop) innehåller det logiska tillståndet hos varje kanal (värde 1 eller 0).

**DATA0, DATA1, DATA2 & DATA3:** Heltalsvariabel som (efter anrop) innehåller det matematiska värdet hos varje grupp om åtta kanaler (00-07, 08-15, 16-23 och 24-31). De logiska tillstånden hos alla 32 kanalerna gör ett Double Word (32 bitar) med Ch0 som LS-bit (minst signifikanta) och Ch31 som MS-bit (mest signifikanta). Detta Double Word kan uttryckas genom de fyra byten DATA0, DATA1, DATA2 & DATA3 där LS-byte är DATA0 och MS-byte är DATA3. Använd dessa variabler i de fall där du vill spara de logiska tillstånden hos varje kanal i en fil. 'Komprimeringen' gör att du kan spara endast fyra byte i stället för 32.

Ett demoprogram har utvecklats för att testa 32-kanalskortet. En skärmdump av detta program visas i figur 5.

För att ändra serieportadressen använder du tangenterna <1> till <4>. Om du vill sluta trycker du på <Esc>.

## Hur du får mjukvaran

Källkoden för kommunikationsrutinen (CARD32DI.SUB) och demonstrationsprogrammet (32DICARD.BAS), med en körbar version av demonstrationsprogrammet (32DICARD.EXE) kan hämtas på denna webbadress: <http://members.xoom.com/robofreak/download/32dicard.htm>

som, för den egentliga nedladdningen tar dig till konstruktörernas hemsida på Internetadressen: <http://www.robofreak.xs3.com>

Tabell 3.

K1		K2	
Kanal	Stiftnummer	Kanal	Stiftnummer
00	33	16	33
01	31	17	31
02	29	18	29
03	27	19	27
04	25	20	25
05	23	21	23
06	21	22	21
07	19	23	19
08	15	24	15
09	13	25	13
10	11	26	11
11	9	27	9
12	7	28	7
13	5	29	5
14	3	30	3
15	1	31	1

Slutligen, du kan nå konstruktören via email till [sebastian@mail.kapatel.gr](mailto:sebastian@mail.kapatel.gr)

(002012-1)

```

REM *****
REM * 32 Channel D/I Card *
REM * CARD32DI Communication Routine *
REM * Copyright (c) April 1999 *
REM * by George Vastianos *
REM * email:robofreak@technologist.com *
REM * http://members.xoom.com/robofreak/ *
REM *****
SUB CARD32DI (COMADDRESS, CHANNEL(), DATA0, DATA1, DATA2, DATA3)

DATA0 = 0: DATA1 = 0: DATA2 = 0: DATA3 = 0

FOR BIT = 0 TO 7

IF (BIT AND 1) = (INP(COMADDRESS + 4) AND 1) THEN
OUT (COMADDRESS + 4), INP(COMADDRESS + 4) XOR 1
END IF
IF (BIT AND 2) = (INP(COMADDRESS + 4) AND 2) THEN
OUT (COMADDRESS + 4), INP(COMADDRESS + 4) XOR 2
END IF
IF (BIT AND 4) = (INP(COMADDRESS + 3) AND 64) / 16 THEN
OUT (COMADDRESS + 3), INP(COMADDRESS + 3) XOR 64
END IF

OUT COMADDRESS + 1, 0
OUT COMADDRESS + 2, 0

INDATA = INP(COMADDRESS + 6) AND 240

CHANNEL(BIT) = (INDATA AND 16) / 16
CHANNEL(BIT + 8) = (INDATA AND 32) / 32
CHANNEL(BIT + 16) = (INDATA AND 64) / 64
CHANNEL(BIT + 24) = (INDATA AND 128) / 128

DATA0 = DATA0 + CHANNEL(BIT) * 2 ^ BIT
DATA1 = DATA1 + CHANNEL(BIT + 8) * 2 ^ BIT
DATA2 = DATA2 + CHANNEL(BIT + 16) * 2 ^ BIT
DATA3 = DATA3 + CHANNEL(BIT + 24) * 2 ^ BIT

NEXT BIT
END SUB

```

Med en sats A/D och D/A omvandlare från Maxim och en handfull andra komponenter kan du tillverka en kurvföljare som kan drivas av PC-ns skrivarport. Ett enkelt BASIC-program tar hand om kommunikationen mellan PC-n och mätkretsen och omvandlar också de uppmätta värdena till grafisk form som kan visas på skärmen.

Källa: Maxim Integrated Products

# I/U kurvföljare

## som drivs från PC-ns skrivarport

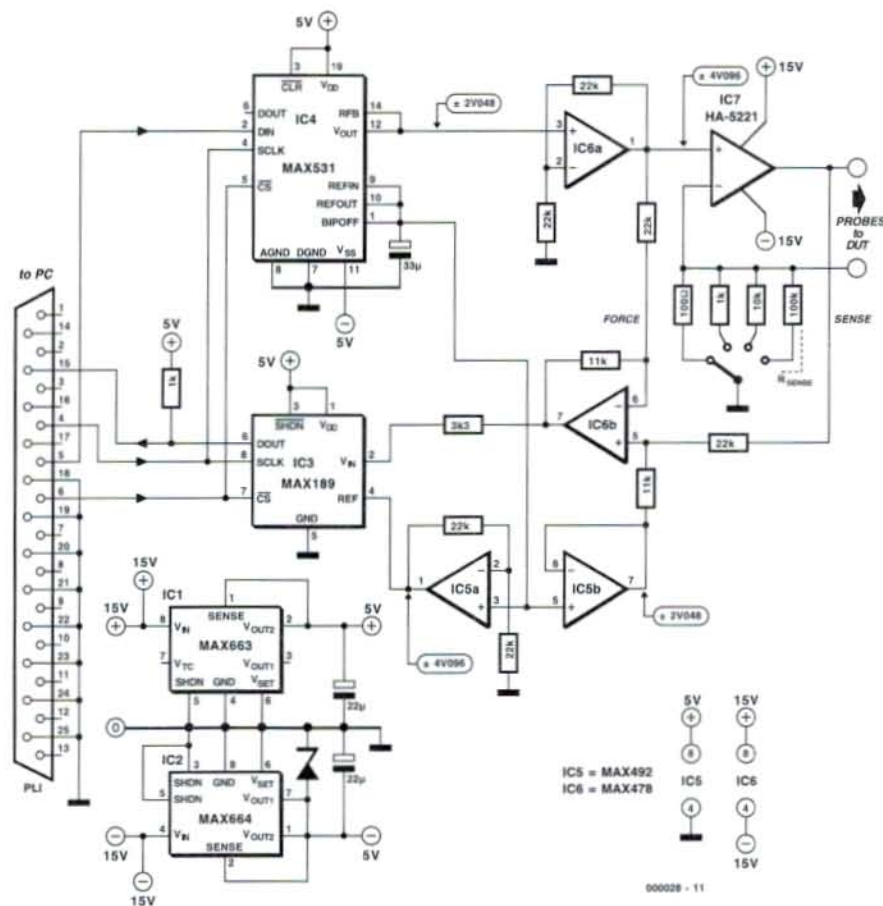
För inte så länge sedan var det vanligt att man mätte I/U-egenskaperna (ström/spänning) hos en halvledare med ett X-Y oscilloskop och en lämplig mätkrets. En tekniker kunde få fram en hel del information angående funktionen hos en halvledare från dessa kurvor.

I den nuvarande dataåldern så har I/U-kurvorna försvunnit in i glömskan, men de är fortfarande användbara när man vill utvärdera ett antal egenskaper hos en halvledare. Enda skillnaden är att vi inte längre behöver ett oscilloskop, utan vi kan i stället använda den mycket mångsidiga PC-n.

Med den mätkrets som presenteras här, och som kommer från Maxim, kan I/U-egenskaperna hos en halvledare, eller IC, mätas och sedan visas på skärmen. Detta är möjligt genom att använda två serieinterface-IC: En 12-bitars DAC och en 12-bitars ADC. Ett kort BASIC-program kontrollerar kretsen och visar data på skärmen. Eftersom det är meningen att använda kretsen på en standard skrivarport på PC-n är samspillet mellan kretsen och datorn mycket enkelt.

### Hårdvaran

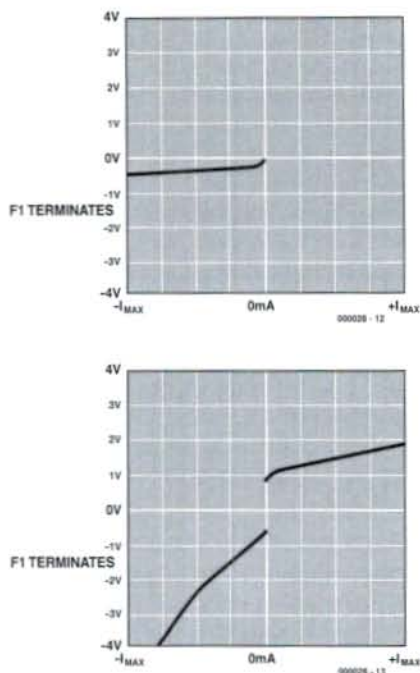
Figur 1 visar kretsschemat för hårdvaran. IC4 är en 12-bitars D/A-omvandlare som är konfigurerad för en bipolär utgångsspänning med område på plus och minus 2,048 V. Opampen IC6a, som är ansluten till Vout-benet, förstärker denna spänning med en faktor 2, så att ett spänningsområde mellan plus och minus 4,096 finns på utgången från IC6a. IC7 omvandlar i sin tur denna spänning till en ström som är proportionell till spänningen. Detta är test-



Figur 1. Huvudkomponenterna i kretsen är en sats A/D och D/A omvandlare med serieinterface från Maxim.

strömmen för den komponent som skall mätas. Strömmen går från  $\pm 40 \mu\text{A}$  till  $\pm 40 \text{mA}$  beroende på det värde som väljs för resistensen  $R_{\text{SENSE}}$  (100  $\Omega$ , 1 k $\Omega$ , 10 k $\Omega$  eller 100 k $\Omega$ ). Den maximala utgångs-

strömmen är i stort sett lika med värdet på utspänningen från IC6a (4,096 V) dividerat med värdet på  $R_{\text{SENSE}}$ . Strömmen genom den komponent som testas producerar ett spenningsfall över komponenten. Denna spänning mäts av



Figur 2. Två exempel på uppmätta I/U-kurvor: (a) visar en Schottky diod, medan (b) visar en mer komplex analog IC.

differentialförstärkaren IC6b. För att undvika ett offset fel, som beror på värdet hos det valda strömkännande motståndet, tas signalen för den invertterande ingången hos denna opamp från den lågimpedanta icke-inverterande ingången hos IC7. Nackdelen med detta är ett fast offsetfel som är lika med ingångsoffset hos IC7.

Förstärkningen i differentialförstärkaren, tillsammans med den adderade offseten, ger ett maximal utgångssving från 0 till 4,096 V. Detta är exakt det rätta området för den unipolära ingången hos 12-bitars A/D-omvandlaren (IC3). Motståndet på 3.3-k $\Omega$  i serie med ingången till ADC begränsar ingångsströmmen i händelse att utgångsspänningen från IC6b skulle vara högre än matningsspänningen för IC3.

När en mätning har skett driver mjukvaran DAC-en på så sätt att den producerar en 'trappstegsström'. Den resulterande spänningen över den komponent som testas mäts av ADC-n och visas sedan på skärmen med en upplösning på 640 x 480 pixlar. De två exempel som visas i figur 2 illustrerar

**BASICprogrammet 'I-V Curve Tracer'** har skrivits av Terry Millward hos Maxim UK, och finns på Maxims **webbsida:**  
<http://www.maxim-ic.com/TechSupport/other.htm>

```
'LISTING 1 - BASIC PROGRAM

'LPT1 OUT @ &H378, IN @ &H379
'LPT2 OUT @ &H278, IN @ &H279

'25 WAY D TYPE

'SIGNALS TO CIRCUIT
'PIN 4 D2 (OUT) SCLK
'PIN 5 D3 (OUT) DATA OUT (FROM PC)
'PIN 6 D4 (OUT) CS\

'SIGNALS FROM CIRCUIT
'PIN 15 D3 (IN) DATA IN (TO PC), ALSO SIGNALS EOC
'PINS 18-25 ARE GROUND

'INTRODUCTION

PRINT "WELCOME TO THE MAXIM CURRENT-VOLTAGE CURVE TRACER"
PRINT

INTRO:
PRINT "WHERE IS THE CIRCUIT CONNECTED? ENTER 1 FOR LPT1 OR 2 FOR LPT2"
INPUT P%
IF P% = 1 THEN PORT% = &H378 'SET LPT1
IF P% = 2 THEN PORT% = &H278 'SET LPT2
IF P% = 1 OR P% = 2 THEN GOTO INTRO1 'TRAP ERROR
PRINT "WRONG!!!!!! PLEASE TRY AGAIN"
GOTO INTRO
INTRO1:

ON KEY(1) GOSUB FINISH 'F1 EXITS
KEY(1) ON

MAIN: 'MAIN BIT OF PROGRAM

'DEFINE SOME VARIABLES
DIM Y(512) 'ARRAY TO HOLD PLOT DATA
DIM DIN(12) AS INTEGER, DOUT(12) AS INTEGER 'DATA IN AND DATA OUT

'SET UP DISPLAY FOR OUTPUT
CLS 0
SCREEN 12
WINDOW (-120, -55)-(520, 435) 'VGA SCREEN
LINE (0, -5)-(0, 410), 1 'DEFINE WINDOW CO-ORDINATES
LINE (-5, 0)-(512, 0), 1 'DRAW A FEW LINES
LINE (512, -5)-(512, 410), 1
LINE (-5, 410)-(512, 410), 1
LINE (-5, 205)-(512, 205), 1 'CENTRAL AXES
LINE (256, -5)-(256, 410), 1
LINE (-5, 51)-(0, 51), 1 'MARKERS
LINE (-5, 102)-(0, 102), 1
LINE (-5, 153)-(0, 153), 1
LINE (-5, 256)-(0, 256), 1
LINE (-5, 308)-(0, 308), 1
LINE (-5, 359)-(0, 359), 1
LINE (64, -5)-(64, 0), 1
LINE (128, -5)-(128, 0), 1
LINE (192, -5)-(192, 0), 1
LINE (320, -5)-(320, 0), 1
LINE (384, -5)-(384, 0), 1
LINE (448, -5)-(448, 0), 1

COLOR 9 'LABELS
LOCATE 1, 1, 0: PRINT "MAXIM CURRENT-VOLTAGE CURVE TRACER"
LOCATE 20, 1, 0: PRINT "F1 TERMINATES"
LOCATE 2, 12, 0: PRINT "+4V"
LOCATE 15, 13, 0: PRINT "0V"
LOCATE 27, 12, 0: PRINT "-4V"
LOCATE 28, 15, 0: PRINT "-Imax"
LOCATE 28, 46, 0: PRINT "0mA"
LOCATE 28, 75, 0: PRINT "+Imax"

LOCATE 10, 1, 0
PRINT "Rs Imax"
PRINT "100R 40mA"
PRINT "1K 4mA"
PRINT "10K 400uA"
PRINT "100K 40uA"

START: 'START OF PLOT ROUTINE
IDATA% = &H10 'INITIALISE PORT, SCLK=0, CS\=1
OUT PORT%, IDATA%

Z% = 8 'INITIALISE MAX531
GOSUB IO
```

```

FOR X% = 1 TO 511 STEP 1
  Z% = 8 * (X% + 1)
  GOSUB IO

  PSET (X%, Y(X%)), 0
  IF Y(X%) = 0 THEN PSET (X%, Y(X%)), 1
  IF Y(X%) = 205 THEN PSET (X%, Y(X%)), 1
  IF X% = 256 THEN PSET (X%, Y(X%)), 1
  Y(X%) = INT(ODATA% / 10)
  IF Y(X%) > 408 THEN Y(X%) = 409
  IF Y(X%) < 1 THEN Y(X%) = 1
  PSET (X%, Y(X%)), 4
NEXT X%
GOTO START

IO:
ODATA% = &H0
IDATA% = IDATA% AND &HEF
OUT PORT%, IDATA%
WAIT (PORT% + &H1), &H8

IDATA% = IDATA% OR &H4
OUT PORT%, IDATA%
IDATA% = IDATA% AND &HF3
OUT PORT%, IDATA%

FOR SHIFT = 11 TO 0 STEP -1
  DOUT(SHIFT) = (INP(PORT% + &H1) AND &H8)
  ODATA% = ODATA% + DOUT(SHIFT) * 2 ^ (SHIFT - 3)
  IDATA% = IDATA% OR ((Z% AND (2 ^ SHIFT)) / (2 ^ SHIFT) * 8)

  OUT PORT%, IDATA%
  IDATA% = IDATA% OR &H4
  OUT PORT%, IDATA%
  IDATA% = IDATA% AND &HF3
  OUT PORT%, IDATA%
NEXT SHIFT
IDATA% = IDATA% OR &H10
OUT PORT%, IDATA%

RETURN
FINISH:
END

```

```

'X% IS X AXIS FOR PLOT
'NEXT VALUE FOR MAX531

'RESET BACKGROUND
'CATCH BOTTOM LINE AT START
'CATCH HORIZ CENTRE LINE
'CATCH VERT CENTRE LINE

'CHECK FOR OVER-RANGES

'SET NEW VALUE
'AROUND AGAIN

'DATA I/O ROUTINE

'CS\ SET TO 0

'WAIT FOR EOC FROM CIRCUIT

'SCLK CYCLE TO SHIFT 189 DATA TO DOUT
'SCLK TO 1

'SCLK TO 0

'GET DATA FROM MAX189
'COMPILE DATA
'DATA TO GO TO MAX531

'SCLK TO 1

'DATA BIT BACK TO 0, SCLK TO

'CS\ SET TO 1

```

(000028-1)

Källa: Maxim Integrated Products

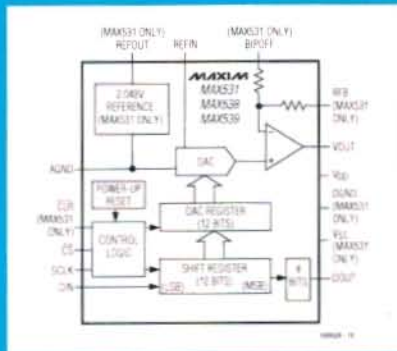
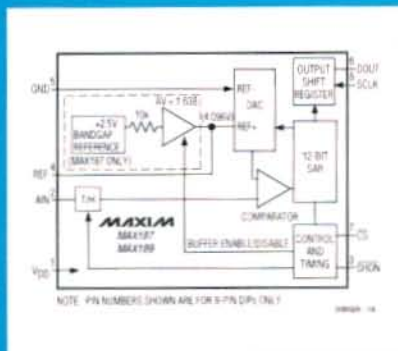
## Serie-interface omvandlare

Enkelheten hos den krets som beskrivs i denna artikel beror i huvudsak på användningen av en sats serie-interfaceomvandlare, nämligen MAX 189 och MAX 1531.

MAX 189 är en seriell A/D-omvandlare med en upplösning på 12 bitar som abretar med en enkel +5 v matning. Dess inspänningsområde ligger också mellan 0 och 5 V. Kärnan i denna IC är en successivt approximerande ADC med en omvandlingstid på 8,5  $\mu$ s. Denna kompletteras med

en snabb sample-and-hold krets (1,5  $\mu$ s), en inbyggd klockgenerator och ett snabbt tretråders interface (se blockschemat i figur A).

Omvandlingshastigheten hos MAX 189 är 25 kilosamples/sekund. Tack vare det inbyggda interfacet kan denna IC enkel anslutas till en PC eller en mikroprocessor, som man kan se i denna applikation. I motsats till dess kompanjon MAX 187 så har MAX189 inte någon intern referens, men detta är inte heller nödvändigt i denna applikation eftersom den DAC som används lämnar referensspänningen. Den mycket låga effektförbrukningen (7 mW) gör denna IC mycket lämplig för batterimatade applikationer. I viloläge (shut-down) sjunker effektförbrukningen till endast 10  $\mu$ W. MAX 189 finns tillgänglig i en 8-bens DIL-kapsel och en 16-bens SOI-kapsel.



MAX 1531 kan man se som ett komplement till MAX 189. Dess blockschema ses i figur B. Denna 12-bitars D/A-omvandlare arbetar också med en enkel +5 V matning. MAX 1531 valdes speciellt för denna applikation, som kräver en bipolär utgångsspänning, eftersom den också kan arbeta med en symmetrisk  $\pm 5$  V matning. Strömförbrukningen hos MAX 1531 är endast 260  $\mu$ A, inklusive den interna 2,048 V referenskällan. IC-n finns i en 16-bens DIP eller SOI kapsel. Offsetspänningen, förstärkningen och linjäriteten justeras under tillverkningen och användaren behöver därför inte bry sig om detta.

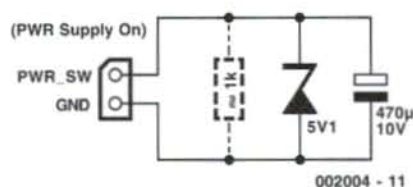
Den inbyggda utgångspumpen hos MAX 1531 kan konfigureras för förstärkningsfaktor 1 eller 2 och för en unipolär eller bipolär utgångs-

Om du installerar ett AT-moderkort med strömanslutningar för både ATX och AT, i kombination med en ATX strömförsörjning, i en AT-låda, kan denna krets bespara dig jobbet med att även installera en extra tryckomkopplare.

Av R. Freitag

# Erättare för en ATX-strömomkopplare

En extra tryckomkopplare behövs normalt för ATX Power Switch/Soft Power Switch signalen, men detta kan du fixa själv med hjälp av denna enkla krets. Det är en konstfull design, men den har testats vid upprepade tillfällen. Zenerdioden är avsedd som skydd mot för höga spänningar och anslutning med omvänd polaritet. I det senare fallet kommer den resulterande kortslutningen (ca 1 A) att överskrida den tillåtna gränsen och ATX-nättdelen kommer att stängas efter ca 5 sekunder. Det kan vara möjligt att använda en mindre



kondensator, men detta måste testas från fall till fall. Om du inte har ordentlig dokumentation för moderkortet skall du verifiera jordstiftet med en signalföljare.

Motståndet behövs bara om du vill kunna slå på datorn igen inom tio sekunder efter det att den stängts av. Det urladdar kondensatorn tillräckligt snabbt för att detta skall kunna ske. Med ett motstånd på 1-k $\Omega$  är tidskonstanten ca 0,5 s. Eftersom kondensatorn också tenderar att stabilisera spänningen kan kretsen också vara till hjälp där ATX-nättdelen stängs av oavsiktligt på grund av spänningsvariationer i nätspänningen.

(002004-1)

## RIKTLINJER FÖR BYGGPROJEKT

Allt om Elektronik tillhandahåller inte andra delar och komponenter än kretskort, folier för frontpaneler och mjukvara på diskett eller IC (inte nödvändigtvis för alla projekt). Komponenter finns normalt att köpa från återförsäljare av elektronik (se annonser i tidningen).

Stora och små värden på komponenter indikeras med hjälp av en av följande prefix:

E (exa) = 10 <sup>18</sup>	a (atto) = 10 <sup>-18</sup>
P (peta) = 10 <sup>15</sup>	f (femto) = 10 <sup>-15</sup>
T (tera) = 10 <sup>12</sup>	p (pico) = 10 <sup>-12</sup>
G (giga) = 10 <sup>9</sup>	n (nano) = 10 <sup>-9</sup>
M (mega) = 10 <sup>6</sup>	$\mu$ (mikro) = 10 <sup>-6</sup>
k (kilo) = 10 <sup>3</sup>	m (milli) = 10 <sup>-3</sup>
h (hekto) = 10 <sup>2</sup>	c (centi) = 10 <sup>-2</sup>
da (deka) = 10 <sup>1</sup>	d (deci) = 10 <sup>-1</sup>

I vissa kretsscheman, för att undvika förvirring, men i motsats till IEC-reglerna, visas komponentvärdet med hjälp av en symbol i stället för den relevanta decimalpunkten. Till exempel,

$$3k9 = 3.9 \text{ k}\Omega$$

$$4\mu7 = 4.7 \mu\text{F}$$

Såvida inget annat indikeras är toleransen hos motstånd  $\pm 5\%$  och effekten 1/3-1/2 watt. Arbetsspänningen hos kondensatorer är  $\approx 50 \text{ V}$ .

Vid bestyckning av ett kretskort skall du alltid börja med de minsta passiva komponenterna, dvs trådbyglar, motstånd och små kondensatorer. Därefter IC-hållare, reläer, elektrolyter och andra stora kondensatorer samt kontakter. Känsliga halvledare och IC-kretsar sist.

**Lödning.** Använd en 15-30 W lödkolv med en fin spets och lödtenn med flussmedel (60/40). Sätt in komponentbenen i kortet, böj dem något och klipp dem kort. Löd. Vänta 1-2 sekunder på att tennet skall flyta ut och avlägsna lödkolven. Inte för mycket värme, speciellt när du löder IC-kretsar och halvledare. Avlödning gör du bäst med en tennsug eller avlödningsfläta.

**Felsökning.** Om kretsen inte fungerar skall du noga jämföra det bestyckade kortet med den publicerade layouten och komponentlistan.

Sitter alla komponenter på rätt ställe? Har du vänt dem åt rätt håll (polaritet)? Har du vänt strömledarna? Är alla lödningar rätt gjorda? Har du glömt någon trådbygga?

Om spänningsnivåer har angetts i kretsschemat så mät på kortet och jämför med dessa - avvikelser på upp till  $\pm 10\%$  från de specificerade värdena är acceptabla.

Eventuella korrigeringar av publicerade projekt publiceras då och då i denna tidning.

Värden på motstånd indikeras av en färgkod på följande sätt.



färg	siffr 1	siffr 2	multipl. faktor	tolerans
svart	-	0	-	-
brun	1	1	$\times 10^1$	$\pm 1\%$
röd	2	2	$\times 10^2$	$\pm 2\%$
orange	3	3	$\times 10^3$	-
gul	4	4	$\times 10^4$	-
grön	5	5	$\times 10^5$	$\pm 0,5\%$
blå	6	6	$\times 10^6$	-
violett	7	7	-	-
grå	8	8	-	-
vit	9	9	-	-
guld	-	-	$\times 10^{-1}$	$\pm 5\%$
silver	-	-	$\times 10^{-2}$	$\pm 10\%$
ingen	-	-	-	$\pm 20\%$

Exempel:

brun-röd-brun-guld = 120  $\Omega$ , 5%

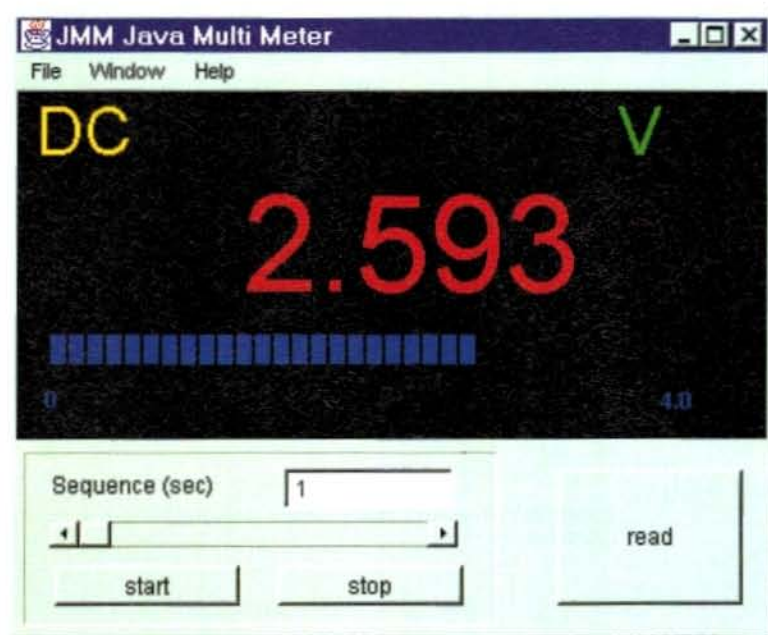
gul-violett-orange-guld = 47 k $\Omega$ , 5%

Java är ett världsspråk. Många tror att detta programmeringsspråk, som utvecklats av Sun Microsystems, enbart kan användas till att lägga ut roliga grejer på Internet. Inget kan vara mer felaktigt. Java är ett fullt utvecklat programmeringsspråk som kan användas för Internetapplikationer och fristående applikationer för att möjliga plattformar. Vad sägs om en Java-applikation i en ring, ett kreditkort, eller i din brödrost? Allt detta är möjligt!

Av G. Polder

# Java MultiMeter (JMM)

## händig mjukvara för RS232 multimeterar



Figur 1. Huvudfönstret i Java MultiMeter shows visar inställningarna för den multimeter som är ansluten till datorn och det uppmätta värdet i form av ett tal och en grafisk stapel. Knapparna kan användas för att göra nya mätningar manuellt, eller låta programmet automatiskt läsa av mätningarna med regelbundna intervaller.

I detta projekt används Java för att läsa data från en multimeter som har en RS232-port (som till exempel Escort 179).

Java MultiMeter (JMM) har en mycket enkel design. I huvudfönstret visas det uppmätta värdet, området och en bargraph (en halv analog mätstapel), plus ett antal knappar. Read-knappen nere till höger används för att kläsa ett värde, medan START och STOPP-knapparna kan användas för att styra upprepande mätningar med ett juster-

bart intervall som kan ställas in med en skjutomkopplare (Sequence).

Fram till hit är ingenting ovanligt. Men det finns mer. Överst i fönstret finns en menyrad som har två alternativ i menyn Window, nämligen 'Plot' och 'Log'. Om du klickar på 'Log' får du upp ett loggningsfönster som visar alla uppmätta värden och tiden när mätningarna skedde. Detta kan vara användbart när man vill titta tillbaka på vad som hänt. Dessutom kan informationen i loggfönstret lagras för senare behandling, till exem-

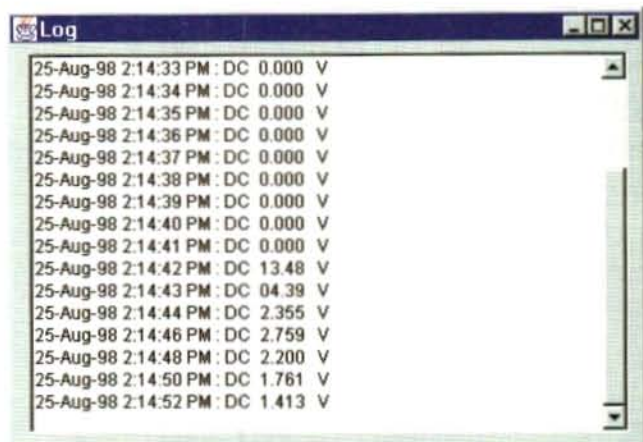
pel i Excel (eller liknande).

Om du klickar på 'Plot' kommer det upp ett plottningsfönster som visar de uppmätta värdena i grafisk form. Denna plottningsfunktion arbetar på ett ganska avancerat sätt. Skalan justeras automatiskt efter varje ny mätpunkt som läggs till. Du kan använda musen och dra en rektangel runt en viss del av plottningen och sedan zooma in genom att dra musen nedåt och åt höger, eller zooma ut genom att dra den uppåt och åt vänster. Om du trycker på knappen 'Fill' visas samtliga data igen. Flera plottningsfönster kan vara öppna samtidigt. Nya mätdata läggs alltid in i det fönster som öppnats sist.

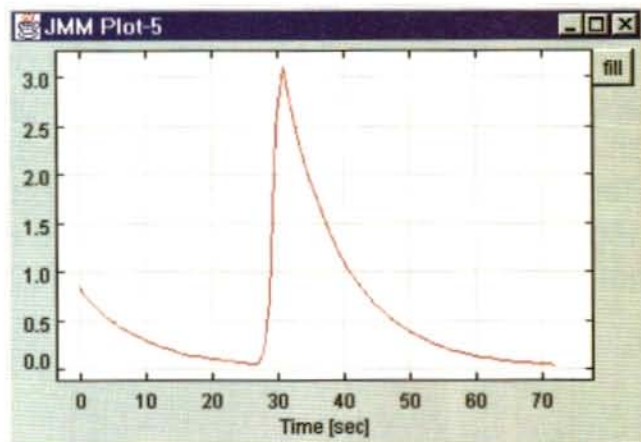
### Arbetsätt

Java är ett objektorienterat språk. Detta betyder att programmeraren gör objekt som kommunicerar med varandra och som är 'semi-samverkande' aktiva. De viktigaste objekten i JMM är följande:

- huvudobjektet, som innehåller mätdata, knapparna och en timer som periodiskt skickar ett 'D' till RS232-porten om denna är aktiv;
- ett RS232 mottagarobjekt, som samlar in RS232-data och skickar hela strängen till huvudobjektet när den tagit emot en <CR> (Carriage Return) sekvens;
- ett loggningsobjekt, som tar emot mätsträngar från huvudobjektet, lägger till datum och tidinformation och organiserar dem i en lista;
- plottningsobjekt, bland vilka det senaste objektet grafiskt visar de data som erhållits från huvudobjektet.



Figur 2. Logfönstret visar en summering av alla mätningar, med tillhörande datum och tid.



Figur 3. Plotmodulen visar ett grafiskt diagram över de senaste uppmätta värdena. Flera plottningsfönster kan vara öppna samtidigt.

## Hårdvara som behövs

Kraven på dator är inte speciellt höga. Det måste naturligtvis vara en PC som körs under Windows 95/98 eller NT. Dessutom behövs en digital multimeter med RS232-port (som Escort 179) för att förse datorn med mätresultat via en seriekana. Slutligen behövs en lämplig RS232-kabel som kopplar ihop multimetern och datorn.

## Mjukvaran

Mjukvaran producerades med hjälp av följande mjukvarupaket:

- ◆ Symantic Visual Cafe: detta är en kraftfull Javaprogrammeringsmiljö;
- ◆ Solutions Consulting serial port software: detta är Javaklasser (rutiner) och Windows DLL-er som gör att Java kan använda sig av datorns RS232-portar.
- ◆ PT-Plot: ett Java-plottningspaket.

En av egenskaperna hos Java är att det är plattformsoberoende. Detta uppnås genom att Javaprogram körs med vad

som kallas en Java Virtual Machine. Med Visual Cafe, är det möjligt att kombinera Javaprogrammen med Virtual Machine till en enda .EXE fil. JMM utnyttjar denna möjlighet. Detta betyder dock att ett antal Semantic DLL-er måste installeras. Ett installationsprogram för dessa, som också inkluderar de resterande komponenterna för JMM, kan du hitta på Elektors CD-ROM 'Software 98-99', som säljs av oss.

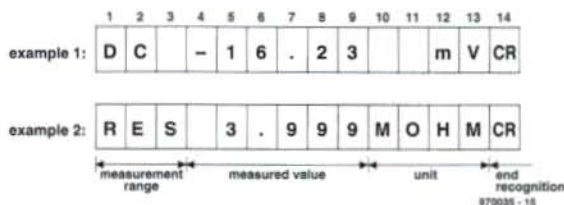
(002010-1)

## Filer som behövs

jmm.exe	Java MultiMeter programmet.
csall	Denna katalog innehåller ett antal av Solutions Consulting serial port DLL-er. Placera dessa DLL-er i katalogen c:\windows\system32 (för Windows 95/98) eller c:\winnt\system32 (för Windows NT).
snjrt20.exe	Installationsprogram för Semantic Java runtime DLL-er.
libwin95	Filer för Windows 95/98. Kopiera dessa filer till katalogen c:\windows\system\lib.
libwinnt	Filer för Windows NT. Kopiera dessa filer till katalogen c:\winnt\system32\lib.

## Det seriella protokollet

Att programmera kommunikationen mellan en PC och en digital multimeter med ett serieinterface är ganska enkelt om du använder ett högnivåspråk som Visual BASIC, Visual C (Java) eller Visual Pascal (Delphi). Till att börja med måste serieinterfacet initialiseras. För de flesta DMM-er skall det konfigureras för 1200 bps, ingen paritet, 7 bitar i ASCII-format, 2 stoppbitar och handskakning via mjukvara. Efter detta är allt klart för att överföra de uppmätta värdena. Allt som datorn, eller programmet, nu måste göra är att fråga efter ett mätvärde. Detta görs genom att sända tecknet 'D' (#68) till DMM-en. Som svar på detta får PC:n en sträng med tecken som innehåller det aktuella mätvärdet, enheten och mätområdet. Den mottagna strängen är 14 byte lång, såsom visas i figuren. Nu är det dock inte alla DMM-er som arbetar på detta sätt. Voltcraft modeller 506 och M-3860M är till exempel undantag. Den senare modellen arbetar med en datahastighet på 9600 baud och skickar alltid mätvärdena för huvuddisplayen och de tre hjälpdisplayerna som ett enda datapaket som är 56 byte långt. Modell 506 arbetar med 1200 baud, men den sänder ett datapaket vars längd varierar mellan 6 och 15 byte. Alla DMM-er är långsamma när det gäller arbetssättet. Nya mätningar skickas inte oftare till PC:n än en gång varje 0,5 sekund.

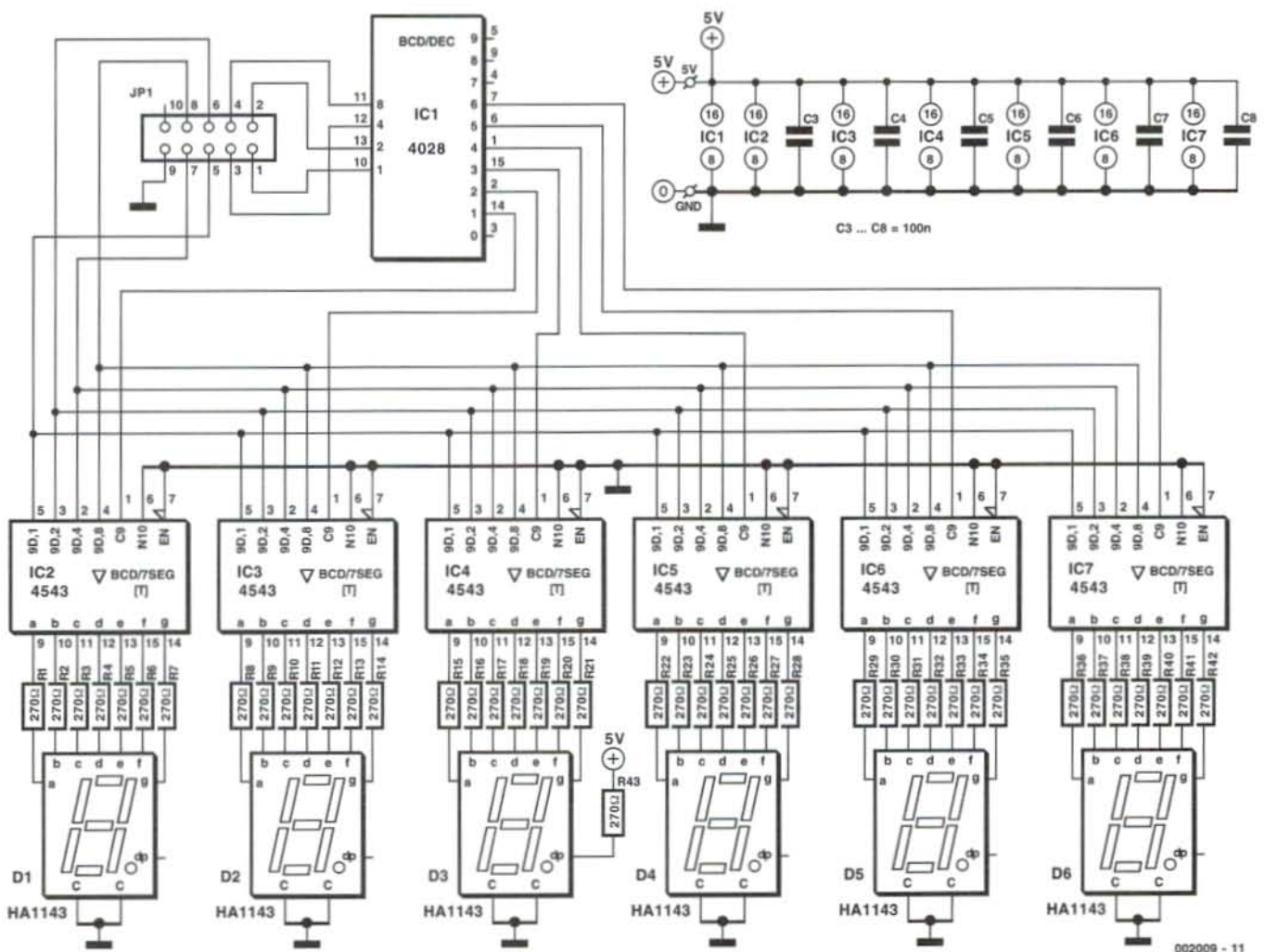


Mikroprocessorer är ena riktiga arbetshästar. I många fall handlar det om att hämta in data, utföra beräkningar och sedan visa resultatet på något sätt. För det sistnämnda, ofta en mycket viktig uppgift, är det nödvändigt med en display. Den krets som presenteras här visar att det endast behövs några få komponenter för att tillverka en billig display som har hög kontrast och således är lätt att läsa.

Design av I. Gerlach

# Mångsidig LED-display

## för många mikroprocessorsystem



Figur 1. Kretsschemat över LED-displayen. Här används endast standardkomponenter. Designen av displayen gör den mycket flexibel, utan att lägga någon större belastning på processorn.

De flesta mikroprocessorsystem är besvärliga att använda utan en bra display. Mikroprocessorer är ju trots allt 'svarta lådor' som mycket snabbt kan utföra massor med beräkningar, men som är totalt gömda för världen utanför. Eftersom beräkningarna inte är ett självändamål, och användaren gärna vill se det färdiga resultatet, så behöver många projekt också en display. Displayer med flytande kristaller (LCD) används allt oftare för detta ändamål. Och skälen till detta är ganska uppenbara:

- ◆ både bokstäver och siffror kan visas på en LC-display;
- ◆ en inbyggd LCD-kontroller gör anslutningen enkelt och minimerar belastningen på processorn;
- ◆ LC-displayer drar lite ström,
- ◆ displayerna är kompakta.

Förutom dessa fördelar har LC-displayer naturligtvis också nackdelar:

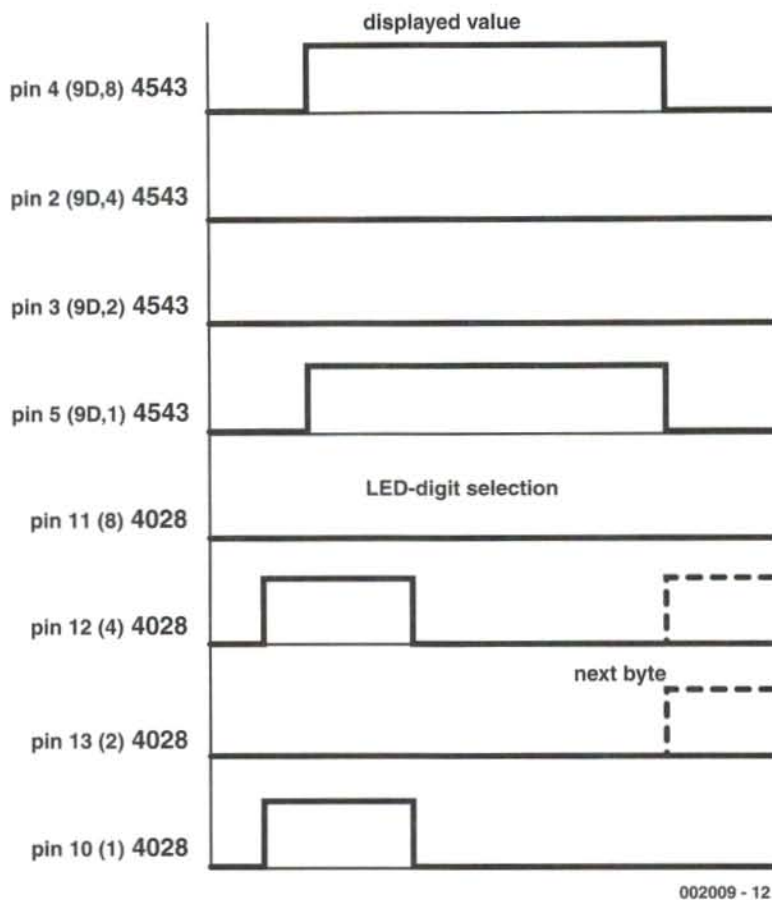
- ◆ de sänder inget ljus och måste således ha en bakgrundsbelysning där ljusförhållandena är dåliga;
- ◆ de har en begränsad betraktningvinkel;
- ◆ de kan endast användas över ett begränsat temperaturområde;
- ◆ de har relativt låg kontrast;
- ◆ de är ganska dyra;
- ◆ olika typer av mjukvara kan behövas, beroende på vilken display som används,
- ◆ dimensionerna på displayen är begränsade,
- ◆ den användbara livstiden är begränsad.

Sammanfattningsvis är en LC-display i många fall ett bra val, men i andra fall kan det vara bättre med en LED-display. I denna artikel visar vi att en 6-teckens LED-display kan byggas till en relativt låg kostnad. Det finns flera primära saker att tänka på vid utvecklingen av denna design. Speciellt måste displayen:

- ◆ använda få I/O-linjer,
- ◆ vara billig att bygga,
- ◆ ta relativt lite beräkningstid från mikroprocessorn,
- ◆ vara flexibel och således utbyggbar,
- ◆ kunna användas med i stort sett alla typer av mikroprocessorer som finns.

## Hårdvaran

Figur 1 visar kretsschemat för displaykretsen. Endast två typer av IC-kretsar används i projektet, och båda dessa är billiga att köpa. IC1 är en 4028, vilket är



Figur 2. Tidsdiagrammet för drivsignalerna. De övre fyra signalerna bestämmer koden för displayen (9 i detta fall) och de undre fyra signaler väljer display-IC (5 i detta fall).

en enkel demultiplexer. De återstående sex IC-kretsarna är 4543-typer. Detta är en BCD-avkodare med inbyggd latch och displaydrivare.

Displaykomponenterna är sju-segments display-IC med gemensam katod. Detta betyder att CC-pinnarna är kopplade till jord. De resterande pinnarna (a, b, ... g) är direkt anslutna till motsvarande utgångar på de tillhörande displaykontrollerna (IC2 till IC7). Slutligen har varje display-IC ytterligare en pinne (ben), som är märkt 'dp'. Denna pinne driver displayens decimalpunkt. I denna design är det endast decimalpunkten i den tredje displayen som är aktiverad, genom att ansluta den till +5 V via ett motstånd på 270 Ω (R43). Om du vill ha decimalpunkten placerad på något annat ställe är det lätt att ändra detta arrangemang.

Hela displayen ansluts till mikroprocessorn via en 'header' med två rader med fem stift vardera. Endast nio av de tio stiften används egentligen. Eftersom ett av stiften ansluts till jord behövs det bara åtta av processorns I/O-linjer för att driva displayen.

En 4-bitars signal som läggs till stiften 1

till 4 väljer den önskade display-IC:n. Bitmönstret på dessa fyra linjer bestämmer vilken utgång från IC:n som skall vara hög. Mönstret '0000' används inte eftersom ingen display-IC är ansluten till pinne 3 (Q0) på IC1. De övriga fyra signallinjerna (stiften 4 till 8) används för att generera de BCD-koder som driver displayen. Så fort som den korrekta BCD-koden har lagts på dessa linjer sätts load-ingången (LD, ben 1) på kontrollern för den önskade display-IC:n hög. På den fallande flanken (från hög till låg) på denna signal lagras 4543-an den kod som finns på dess ingångar 1A till 1D (benen 2 till 5). Det mönster för utgångsbenen (AS till G) som går högt för en speciell ingångskod, bestäms av en matris som är integrerad i avkodar-IC:n. De associerade segmenten i display-IC:n tänds då.

Hela processen illustreras i det tidsdiagram som visas i figur 2. De fyra övre bitarna visar att BCD-koden '1001' läggs till ingångarna på displaykontrollerna. Efter detta aktiveras IC5 genom att 4028-an avkodar väljarkoden '0101' (decimalt 5).

Denna lösning har två viktiga fördelar för

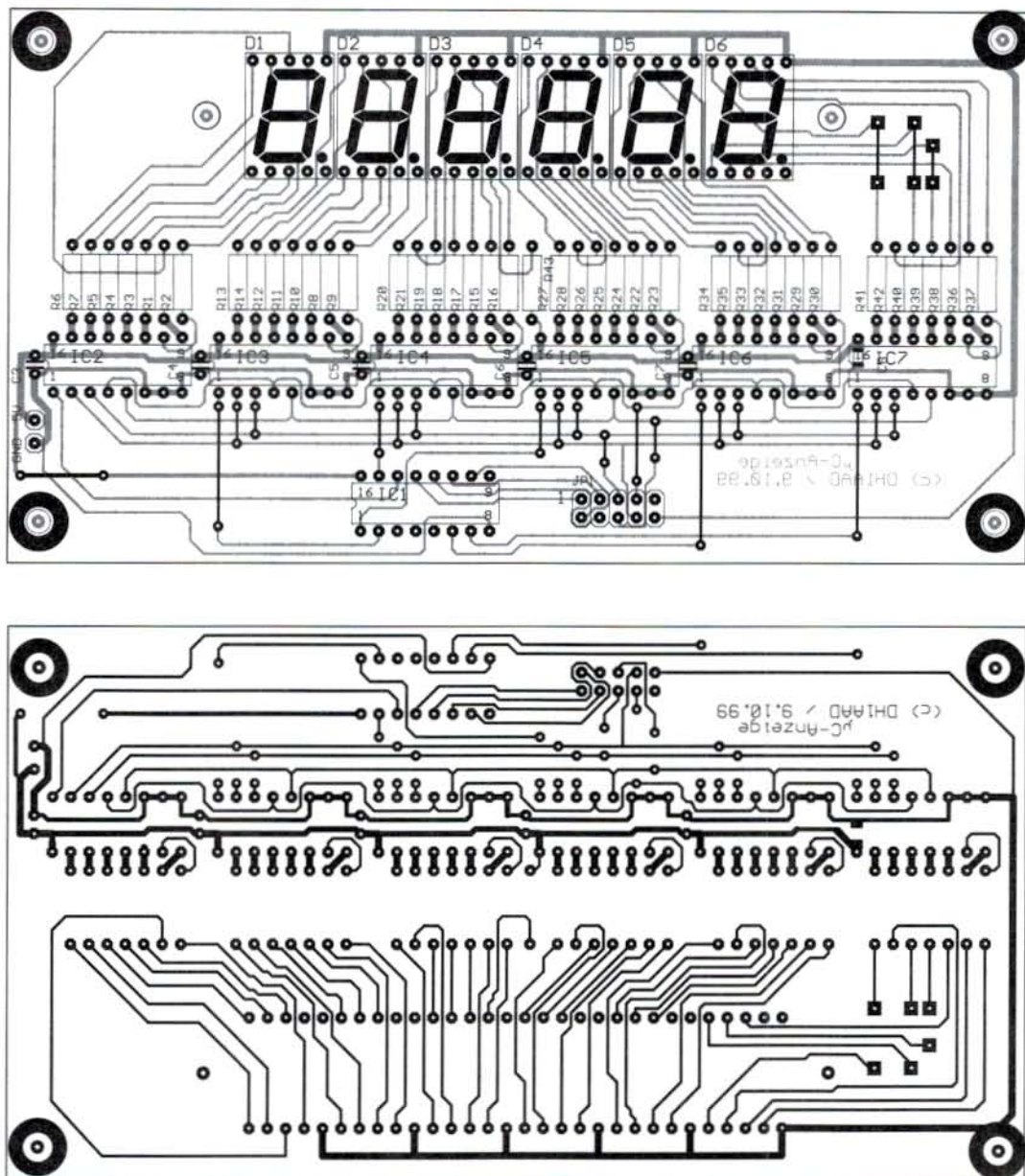


Figure . Kretskortslayouten och komponentplaceringsritningen för LED-displayen.

användaren. För det första, så länge som det visade värdet inte har ändrats så behöver några nya kontrollsignaler inte sändas till displayen. För det andra är det möjligt att modifiera det visade värdet med en enda display-IC. Dessa två egenskaper betyder att belastningen på processorn hålls vid ett minimum. Det finns inte mycket att säga angående resten av kretsen. En enkel matningsspänning på 5 V används. Varje IC är försedd med en avkopplingskondensator. Seriemotstånden för displaysegmenten är utvalda för den matningsspänning som används. Om du vill att displayen skall lysa starkare kan värdet på dessa motstånd (R1 till R43) minskas, till exempel till 180 Ω. Om du i stället vill minska ljusstyrkan hos dis-

playerna så ökar du bara värdet på dessa motstånd.

### Kretskort

För att det skall vara enkelt att bygga displayen har vi tagit fram en kretskortslayout. Vi säljer inte något färdigt kort till detta projekt utan du får etsa det själv. Detta borde inte vara särskilt svårt eftersom layouten är okomplicerad och kortet är enkelsidigt.

När du sedan skall bestycka kortet så börja med trådbyglarna. Efter detta kan du montera resten av komponenterna. Använd helst IC-hållare och var noggrann med lödningen. För IC2 till IC7 har den sista motståndspositionen inte något komponentnummer. Detta är

det motstånd som driver decimalpunkten (R43 i kretsschemat). Först bestämmer du dig för om du vill ha en decimalpunkt och därefter bestämmer du var denna skall finnas. Allt du sedan behöver göra för att aktivera den är att löda in ett seriemotstånd på 270-Ω på den korrekta platsen.

### Mjukvara

För att kunna använda displayen tillsammans med en mikrokontroller behöver du naturligtvis någon form av mjukvara. Följaktligen har vi inkluderat ett litet program i form av en maskinspråkrutin som är lämplig för Atmel AT90S1200 till AT90S8515 processorer. Vi har valt att använda en sepa-

rat drivrutin för att göra applikationen så enkel som möjligt. Detta gör det möjligt med en flexibel lösning och denna drivrutin kan anropas från en applikation. Detta betyder att kontrollmjukvaran bara behöver skrivas en enda gång, och därefter kan den användas i flera olika system. Lista 1 visar designen av denna drivrutin. Den har en mycket rakfram konstruktion, vilket gör att dess funktion är lätt att följa.

Lista 2 innehåller ett kort demonstrationsprogram. Detta program sätter först alla displayer till noll och visar sedan talet '145675'. Om decimal-

punkten ställs in efter det tredje segmentet kommer det värde som visas på displayen således att vara 145.675. Du kan enkelt experimentera med denna flexibla LED-display. Om koden måste översättas för en annan processorfamilj är det lätt att göra detta med hjälp av den information som ges i listorna.

(002009-1)

```

;*****
;* File Name      :LEDDisp.inc
;* Title          :Driver for LED Display
;* Date          :Ingo Gerlach / 10.10.99
;* Version       :1.0 / 11.10.99
;* Version       :
;* uC            :AT90S1200...8515
;*              :
;* Changes       :
;*              :
;*****
;
;
; Main program register variables
;-----
;.def temp      = r16
; Registers / LED
;-----
;.def cnt      = r20 ; counter
;.def dly      = r21 ; delay loop variable
;.def pos      = r23 ; position
;.def byte     = r24 ; byte

; Equates
;-----
;.equ LED_qty  = 6           ; number of LEDs
;.equ LED_Del  = 45         ; delay
;.equ OutPort  = PortB

; Functions
; LED_Blank    : switch display on
; LED_Null     : reset display 0 ( Null)
; LED_Show     : show bytes , transport byte (R24),
;               position (R23)

; **** Switch display off
;*****
LED_Blank:    ldi    cnt,LED_QTY
LedLoop1:    ldi    temp,192
             add    temp,cnt
             out    OutPort,temp
             rcall  Led_Delay
             dec    cnt
             brne  LedLoop1
             Ret

;*****
; **** Reset display
;*****
LED_Null:    ldi    cnt,LED_QTY ; load number of LEDs
LedLoop2:    out    OutPort,cnt
             rcall  Led_Delay
             dec    cnt
             brne  LedLoop2
             out    OutPort,cnt
             Ret

;*****
; **** Show byte
;*****
LED_Show:    mov    temp,pos ; position in register
             out    OutPort,temp ; activate BCD-
to-decimal decoder, LD 4543
             rcall  Led_Delay ; short delay
             mov    temp,byte ; value in
             ; register
             swap   temp ; value high
             ; nibble
             add    temp,pos ; goto position
             out    OutPort,temp ; 4028 / 4543
             ; move value to display
             rcall  Led_Delay ; short delay
             sub    temp,pos
             out    OutPort,temp ; LD signal off.

store value
             rcall  Led_Delay
             Ret

;*****
; * Internal functions !!!
;*****
; *** Delay LED display
;*****
LED_Delay:   ldi    dly,LED_Del
LedLoop:    dec    dly
             brne  LEDLoop
             ret

```

Lista 1. En maskinspråksrutin för LED-displayen. Denna kan integreras i olika applikationer.

```

;*****
;* File Name      :LED.asm
;* Title          :Test program for LED display
;* Date          :Ingo Gerlach / 10.10.99
;* Version       :1.0 / 10.10.99
;* Version       :
;* µC            :AT90S1200...8515
;*
;* Changes :
;*
;*****

;***** Directives

.device AT90S1200           ;device type
.NOLIST

.include "1200def.inc"

.list
.listmac

; Show data
; Structure of data

;      MSB                LSB
;      7 6 5 4           3 2 1 0
;      0 0 0 0           0 0 0 0
;
;      Select position 1 = 1, 2 = 2, etc
;      Number in BCD code
;
;e.g. 10010010b = 146d = pos. 2, value 9

;
; Main program register variables
;-----
.def    temp      = r16

; Registers / LED
;-----
.def    cntr      = r20 ; counter
.def    dly       = r21 ; delay loop variable
.def    pos       = r23 ; position
.def    byte      = r24 ; byte

; Equates
;-----
.equ   LED_qty   = 6           ; number of LEDs
.equ   LED_Del   = 40          ; delay
.equ   OutPort   = PortB

;***** Interrupt vector table

reset:

                                rjmp  main ; main routine
                                reti   ; external interrupt0 handle
                                reti   ; T/C0 overflow interrupt handle
                                reti   ; analogue comparator interrupt handle

;***** Functions
;*****

;***** Main *****

main:
; ldi   temp, LOW(RAMEND)      ; setup StackPointer
                                ; for > 90S1200
; out   SPL, temp             ; initialize SPL
; ldi   temp, HIGH(RAMEND)    ; initialize SPH
; out   SPH, temp

                                ldi   temp,255           ; temp = 255
                                out   ddrb,temp         ; port B output

                                Rcall LED_Null          ; reset display

mainloop:
; Show 145.675

                                ldi   Pos,1
                                ldi   Byte,1
                                Rcall Led_Show

                                ldi   Pos,2
                                ldi   Byte,4
                                Rcall Led_Show

                                ldi   Pos,3
                                ldi   Byte,5
                                Rcall Led_Show

                                ldi   Pos,4
                                ldi   Byte,6
                                Rcall Led_Show

                                ldi   Pos,5
                                ldi   Byte,7
                                Rcall Led_Show

                                ldi   Pos,6
                                ldi   Byte,5
                                Rcall Led_Show

forever:    rjmp forever

; ***** End of main program *****

; *** Include Files ***
.include "LEDDisp.inc"

```

Lista 2. Ett demonstrationsprogram som visar hur drivrutinen kan integreras med en applikation.