

ELEKTOR Allt om **Elektronik**

Sveriges största specialtidning för tillämpad elektronik

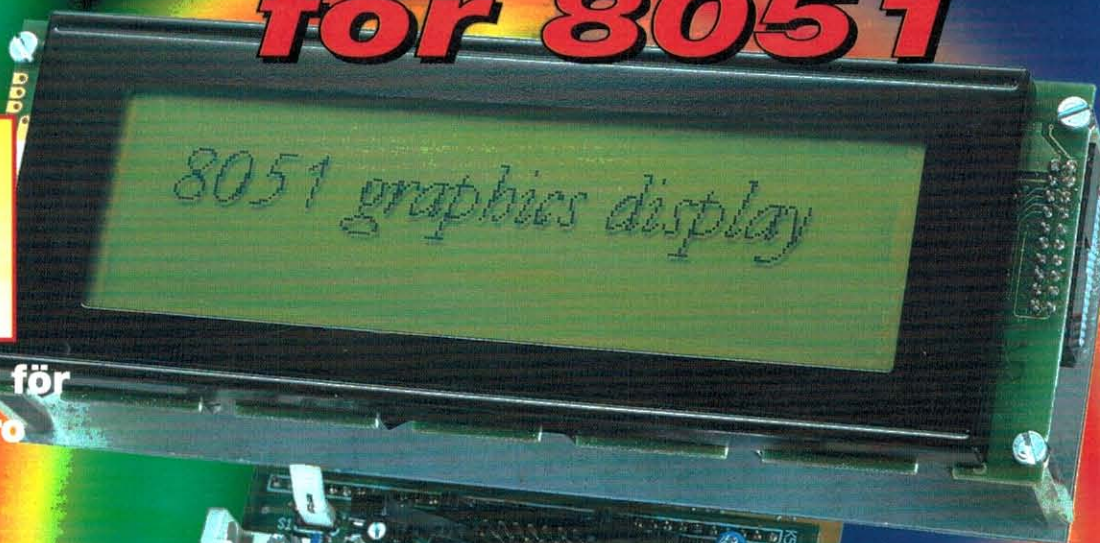


**Personlig Mini
Webserver.**
Styr ditt hem via
Internet eller via
Ethernet

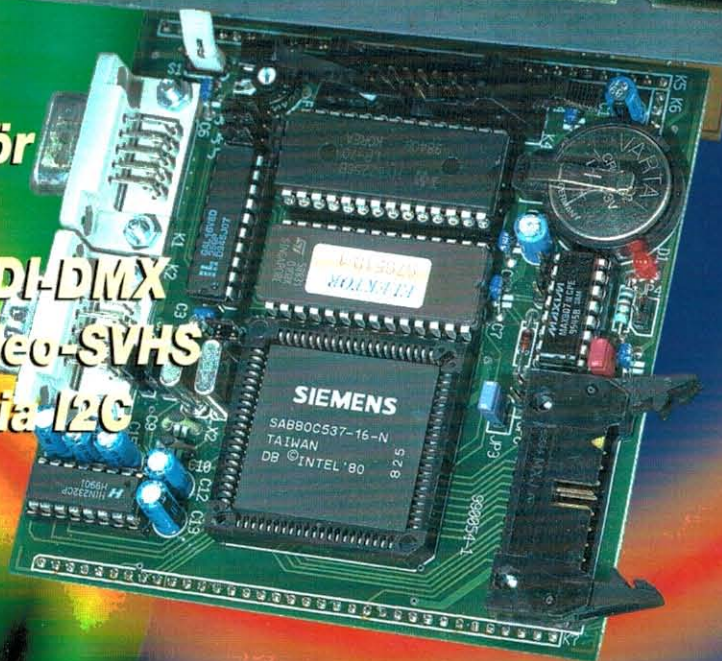


**Programmerare för
Atmel AVR-mikro**

Grafisk LCD-modul för 8051



- Återupplivare för blybatterier
- Omvandlare MIDI-DMX
- Omvandlare Video-SVHS
- Servostyrning via I2C



INTERPRESS 0173-08
SPECIALRETUR



08

Chefredaktör och ansvarig utgivare
Bill Cedrum

Redaktionen

Box 2094
194 02 Upplands Väsby
Besöksadress
Industrivägen 8
194 77 Upplands Väsby

Redaktionssekreterare

Asta Russell

Utgivare

Electronic Press AB
Box 2094
194 02
Upplands Väsby
Tel: 08-710 08 90
Fax: 08-710 08 92
E-mail: red@alltomelektronik.com
http://www.alltomelektronik.com

Prenumerationer 2001

PressData AB
Tel: 08-799 63 25
Fax: 08-98 45 72
Helår (11 nummer) 485:-
Halvår (6 nummer) 285:-
2 år (22 nummer) 875:-
Norge helår 420:- Nkr + porto
Norge halvår 255:- Nkr + porto
Norge 2 år 755:- Nkr + porto

Annonser

Annonssavdelningen
Box 2094
194 02 Upplands Väsby
Tel: 08-710 08 90
Fax: 08-710 08 92
E-mail: annons@alltomelektronik.com

Tryck

NDB-Printing
Zoeterwoude, Nederland 2001

ISSN 0280-8765

Samtliga artiklar, fotografier, kretskortslayouter och annat är copyrightskyddat och får endast återges efter utgivarens skriftliga tillstånd. Detta gäller även delar av materialet. Alt om Elektronik har, så långt som är möjligt, kontrollerat samtliga byggprojekt. Dock tar vi inget ansvar för fel som ligger utanför vår kontroll. Ej heller tar vi ansvar för byggen som våra läsare utför.

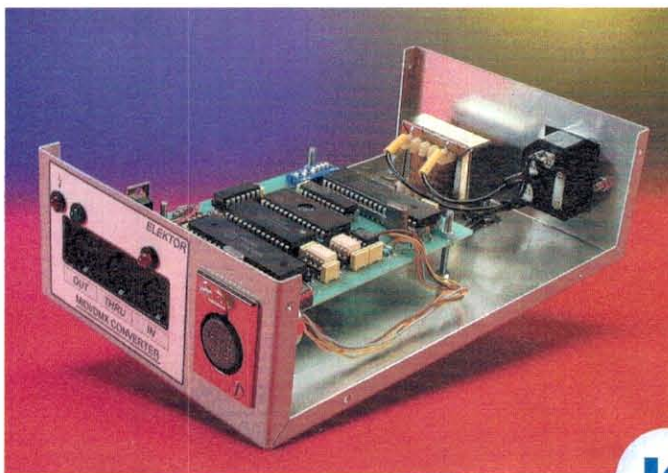
Copyright: Segment B.V.
(Beek, the Netherlands) 2001
P.O. Box 75
6190 AB Beek (L), The Netherlands

BYGGPROJEKT

MIDI-DMX interface

Här visar vi hur du styr ditt DMX-nät från en sequencer! Med DMX512-baserad fjärrstyrning kan upp till projektorer eller strålkastare kontrolleras på upp till 512 kanaler via en RS485-länk. Då varje DMX-system omfattar en master och flera slavar som styrs med DMX-kommandon är det ingen dum idé

att koppla
koppla
dem till
MIDI.
Här kan
du läsa hur
du gör.

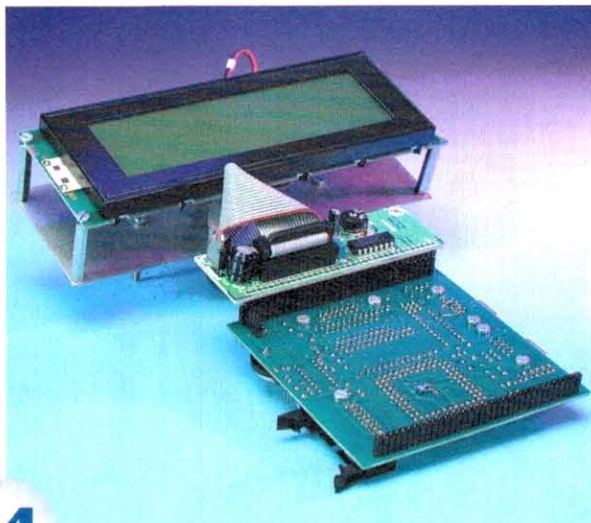


16

Grafisk LCD-modul för 8051

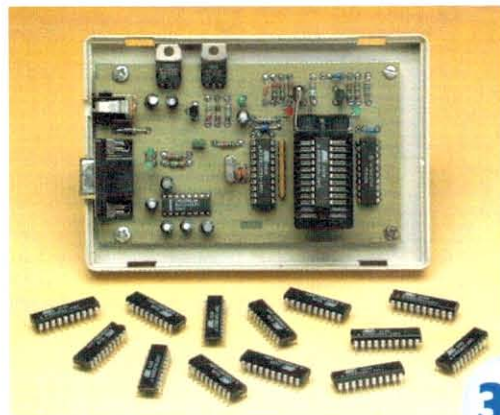
I denna artikel beskriver vi hur en display med toshibas populära kontroller T6963 kan drivas av en 8051 (eller kompatibel) mikrokontroller.

Läs här hur du kan utnyttja hela displayen för en grafisk bild och hur du omvandlar en bild i din PC till de pixlar som displayen vill ha.



24

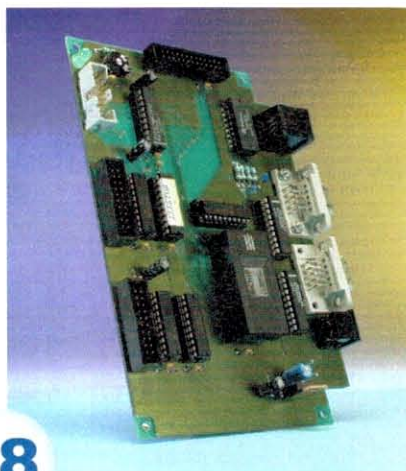
Programmerare för Atmel AVR



Denna programmerare är avsedd för att programmera de tre populära 20-bens processorer 89C1051, 89C2051 och den nya 89C4051 med 4 k programminne.

Personlig Mini Webserver (I)

En 'inbäddad' PC i en DIL-kapsel bildar kärnan i denna Mini Webserver. I första artikeln visar vi ett utvecklingskort för denna kontrollern och visar hur du använder det. Efter detta kan du styra vilken apparat du vill via Internet eller Ethernet.



Byggprojekt

- 6 Återupplivare för gamla blybatterier
Spara pengar genom att ge ditt batteri nytt liv
- 10 I2C servointerface
Styr 8 servon via I2C-bussen
- 14 Video till S-VHS omvandlare
Från komposit (CVBS) till C/Y
- 16 MIDI-DMX interface
Styr ditt DMX-nät från en sequencer!
- 24 Grafisk LCD-Modul för 8051 mikrokontroller
- 30 Programmerare för Atmel AVR mikrokontroller
- 36 Personlig Mini Webserver (I)
*Årets 'killer' projekt.
Ett utvecklingskort med en inbäddad PC.*

Småprojekt

- 44 Integrerat röstminne
- 45 Växlande blinker
- 46 Indikator för trasig säkring

Ovrigt

- 48 Läsarservice
- 49 Elektronik On-Line
Trimma din CD-brännare
- 50 Nästa nummer

VIKTIG INFORMATION!

Vi flyttar!

Den 15 oktober flyttar Allt om Elektronik från Upplands Väsby. Vårt flyttlass drar västerut, till Stenungsund, som ligger ungefär fyra mil norr om Göteborg.

Ny adress och nytt telefonnummer i samband med detta blir:

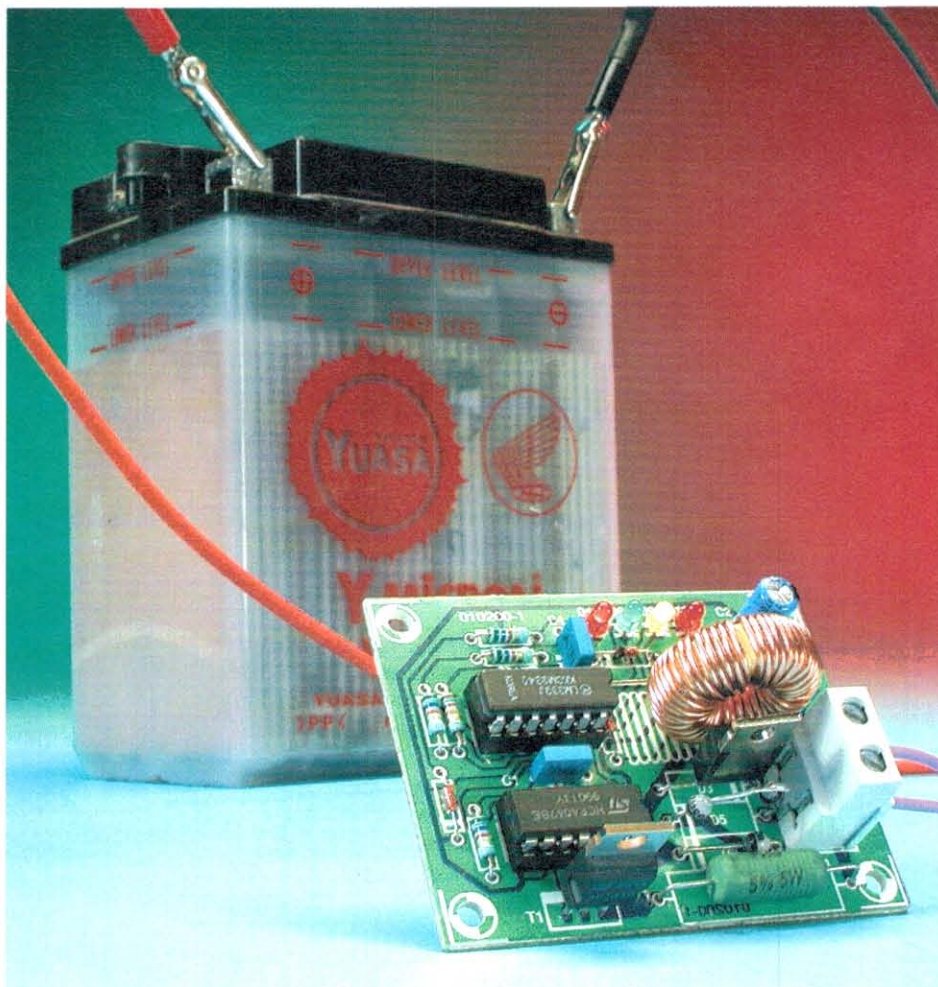
Allt om Elektronik
Box 178
444 22 Stenungsund
Tel: 0303-770 490

Återupplivare för gamla blybatterier

en avsulfateringsenhet för utslitna batterier

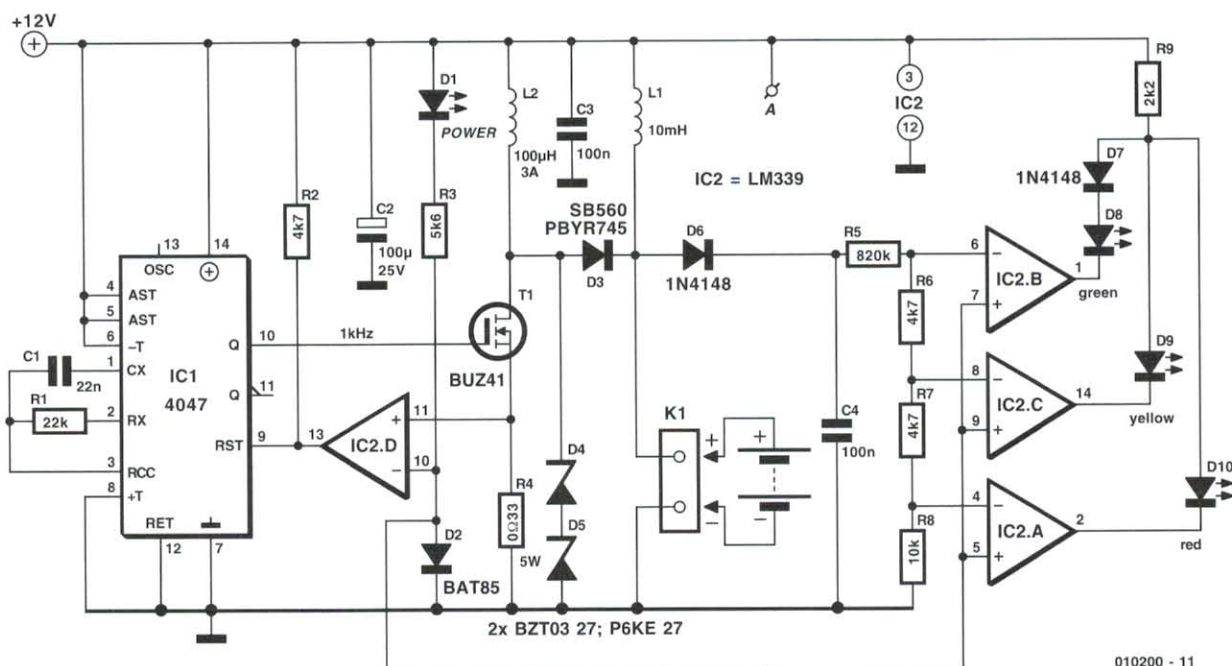
Design av K. Walraven

Med denna krets kan du göra något som tidigare var otänkbart - vända på sulfateringseffekten i trötta batterier. Kretsen rekommenderas också som konditionerare för nya batterier.



I en vetenskaplig publikation nyligen konstaterades att 80% av blybatterierna lägger av på grund av sulfatering. Denna sulfatering beror på ålderdom, laddnings och urladdningsperioder under mindre idealiska förhållanden, eller att batteriet har stått urladdat för länge. Det senare händer ofta med batterier i motorcyklar eller veteranbilar, eftersom dessa ofta inte används under vintermånaderna. För att råda bot på detta presenterade vi en speciell laddare för ett par år sedan, som håller batterierna i god kondition under deras vintersömn.

Vad är då sulfatering? Detta är det som sker då den bly-sulfat som bildas på plattorna i ett batteri under urladdningen ändrar struktur. Det bildas ganska stora sulfatkristaller, vilka blockerar porerna i blyplattorna och således minskar deras aktiva yta. Detta minskar batteriets kapacitet, det kan inte längre lämna stora strömmar och det kan inte längre laddas effektivt på vanligt sätt. När ett på så sätt sulfaterat batteri laddas bildas ledande bryggor (kortslutningar) mellan plattorna och fram till nu har man trott att dessa inte kunde tas bort. Och batteriet hade alltså nått slutet på sitt verksamma liv.



Figur 1. Kretsen består av en pulsgenerator och en indikatorsektion.

Kända knep

I den välkända elektronikandan så tar vi inte trött batteri direkt till en återvinningsstation. Trots allt är ju batterier inte billiga och det är därför värt att kolla om det verkligen inte går att återuppliva det.

Många av er känner säkert till olika knep för att peppa upp trötta batterier. Ett av de bäst kända är att ladda och ladda ur batteriet upprepade gånger. Av ett eller annat skäl så kan man återställa en hel del av den förlorade kapaciteten på detta sätt. I andra fall verkar det som om stora pulserande strömmar in i batteriet har samma effekt.

Både dessa metoder lämnar dock en del övrigt att önska när det gäller att återuppliva svårt sulfaterade batterier.

Bot

Under senare år har flera tillverkare utvecklat metoder för att vända sulfateringsprocessen i blybatterier med skilda framgångar. Arbetsmetoden verkar bygga på någon form av pulsad laddning. Detta i motsats till normala laddningsprocedurer, som i de flesta fall använder en konstant spänning.

Den design som vi presenterar här

representerar den senaste tekniken för återupplivning av blybatterier. Det är en enhet som periodiskt matar korta, men kraftfulla, pulser till batteriet, medan det samtidigt laddas ur något mellan pulserna. Detta är, såvitt vi vet idag, det mest effektiva sättet att bryta sönder oönskade avlagringar av sulfatkristaller och återställa plattorna i någorlunda hyggligt skick.

Eftersom den energi som behövs för laddningspulserna tas från själva batteriet (detta kanske verkar lite konstigt till att börja med, men urladdningen av batteriet är också en del av denna process), så rekommenderar vi er att ansluta en vanlig laddare parallellt med batteriet och återupplivaren när batteriet har mycket lite kapacitet kvar, men detta går vi närmare in på längre fram i artikeln.

För att vara riktigt ärliga måste vi erkänna att våra egna experiment med denna krets inte är tillräckliga för att ge en villkorlös garanti för framgång. Vi har inte haft tillräckligt många batterier att testa på. Men eftersom kretsen är billig att bygga så överlåter vi åt våra läsare att bedöma hur väl den fungerar i verkliga livet.

Pulsgenerator

Som man kan se av kretsschemat i figur 1 är den elektronik som behövs för att bygga återupplivaren inte särskilt omfattande. Kretsen omfattar två delar, en generator byggd runt IC1, IC2d och T1, som skapar laddningspulserna, samt en indikatorkrets bestående av tre operationsförstärkare (IC2a, b, c) och tre lysdioder, vilka visar batteriets tillstånd. Låt oss först titta på pulsgeneratoren. På samma sätt som gäller för resten av kretsen tas matningen till denna från batteriet via K1. När vi talar om matningen så skall denna ha en relativt konstant spänning och vara fri från spikar (förutom de som kretsen själv genererar).

Drosseln L1 har lagts till för att ta bort oönskade spikar medan C2 och C3 fungerar som lagringskondensatorer. Lysdioden D1 tänds när det finns spänning.

För att fortsätta med pulsgeneratoren så skapar IC1 (en 4047) en fyrkantvåg med en frekvens på 1 kHz och en pulsfaktor som normalt är 50%. Så fort som Q-utgången från IC1 går hög börjar FET T1 att leda. Detta får en ström (urladdning) att flyta från batteriet genom L2, som ökar linjärt tills spänningen över R4 är ca 0,35 V, strömmen är då ca 1 A.

I detta ögonblick kommer komparatorn IC2a att switcha över, vilket återställer IC1 och stänger T1. Den lagrade magnetiska energin i L2 omvandlas nu till en spänningsspik, som matas till batteriet via D3.

KOMPONENTLISTA

Motstånd:

R1 = 22k Ω
 R2, R6, R7 = 4k Ω 7
 R3 = 5k Ω 6
 R4 = 0 Ω 33 5W
 R5 = 820k Ω
 R8 = 10k Ω
 R9 = 2k Ω 2

Kondensatorer:

C1 = 22nF
 C2 = 100 μ F 25V radial
 C3, C4 = 100nF

Spolar:

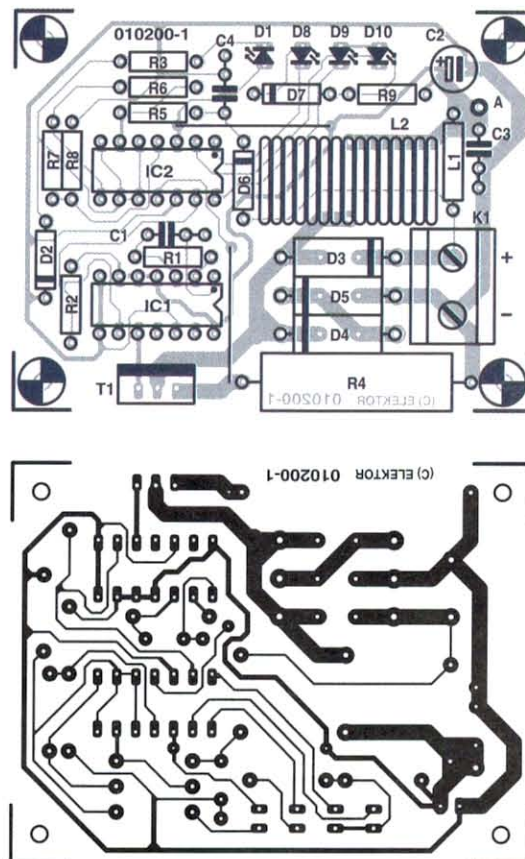
L1 = 10mH
 L2 = 100 μ H 3A dämpdrossel

Halvledare:

D1 = LED
 D2 = BAT85
 D3 = SB560 eller PBYR745
 D4, D5 = BZT03 27 eller P6KE 27
 D6, D7 = 1N4148
 D8 = LED, grön (dagsljus)
 D9 = LED, gul (dagsljus)
 D10 = LED, röd (dagsljus)
 T1 = BUZ41
 IC1 = 4047
 IC2 = LM339

Övrigt:

K1 = 2-pol kopplingsplint för kretskort,
 raster 7,5 mm
 Kretskort 010200-1 (se Läsaerservice eller
 vår websida)



Figur 2. Detta kretskort gör det lätt att bygga återupplivaren.

Storleken på spiken är beroende på batteriets kondition. När batteriet är i hygglig kondition är dess inre resistens ganska låg och toppen på spiken kommer också att vara låg (mindre än 15 V). Vid en hög inre resistens kan toppen vara så hög som 50 V. Det maximala värdet begränsas av de två seriekopplade zenerdioderna, D4 och D5.

Indikatorer

Eftersom batteriets kondition kan avgöras från storleken på laddningspulserna så har vi lagt till en enkel krets som indikerar pulsernas toppvärde. De tre komparatorerna IC2a-IC2c mäter det toppvärde som lagrats i C4 och switchar över vid spänningar på 15, 20 respektive 30 V. Med ett batteri i relativt bra kondition tänds alltså den gröna LED-en D8, med ett halvdåligt batteri den gula LED-en D9 och med ett riktigt dåligt batteri den röda LED-en D10.

Det är en detalj som skall nämnas när det gäller indikatorkretsen. För att undvika att al-

la tre lysdioderna tänds samtidigt när toppspänningen är mycket hög så har de parallellkopplats till ett gemensamt seriemotstånd (R9). Eftersom den röda LED-en har ett lägre spänningsfall än den gula kommer de aldrig att tändas samtidigt. Eftersom den gula och den gröna lysdioden har samma spänningsfall fungerar inte samma knep här. Det är därför som den gröna LED-en har en vanlig diod (D7) seriekopplad med sig.

Själva bygget

Vi har tagit fram ett litet kretskort för detta projekt (figur 2) och därmed är det enkelt att bygga återupplivaren även för er som inte är så vana vid att bygga elektronik. Det är bara att följa komponentplaceringsritningen och komponentlistan noga. Batteriet ansluts till kopplingsplinten K1 (tänk på polariteten!). Glöm inte trådbyglarna, det finns två av dessa och kretsen fungerar inte utan dem.

Eftersom laddningspulserna kan or-

saka högfrekvensstörningar så är det bäst att bygga in kretsen i en metalllåda.

Komponentvalet är inte särskilt kritiskt. Vilken liten Schottkydiod som helst kan användas för D2. För D3 går det bra med vilken Schottkydiod som helst som är på minst 60 V/3 A. Även när det gäller T1 har du mycket att välja på, eftersom vilken effekt-FET som helst på 3 A/100 V går att använda. Även den välkända BUZ10 går att använda, men då skall zenerspänningen minskas till 27 V genom att ersätta en av zenerdioderna (D4 eller D5) med en trådbygel. En viktig sak när det gäller dessa zenerdioder, de kan inte vara normala typer utan måste vara snabba. Zenerspänningen som sådan är inte kritisk, men grundförutsättningen är att den totala spänningen hos zenerdioderna D4 och D5 skall ligga inom området 40-50 V.

Utelämnas dock inte zenerdioderna eftersom detta är ett säkert sätt att förstöra MOSFET T1!

För L2 används en vanlig störnings-

drossel på minst 3 A. Induktansen hos drosseln är inte kritisk, vilket värde som helst mellan 50 μH och 200 μH går bra. Speciella drosslar för switchade nätaggregat går också att använda, de fungerar ofta bättre. Värdet på L1 är inte heller kritiskt utan kan vara mellan halva och dubbla det angivna värdet på 10 mH.

Användning

Det finns tre olika sätt att använda återupplivaren på.

Det första är att använda den i ett befintligt system (i en bil till exempel) för att förhindra att sulfatering överhuvudtaget uppstår i ett batteri. Kretsen integreras med systemet genom att ansluta den direkt till batteriet med så korta kablar som möjligt. Eftersom kretsen kan lämnas permanent ansluten behöver du inte göra någonting mer. Strömförbrukningen är ungefär 20 mA, så batteriet kan laddas ur om du inte använder bilen under längre tid.

Återställning av batterier som redan är sulfaterade kan göras på två sätt. Första sättet är att ladda batteriet, koppla från laddaren och ansluta återupplivaren. Eftersom energin för laddningspulserna tas från själva batteriet kommer detta att laddas ur långsamt. Denna process måste följas noga eftersom ett fullt urladdat batteri måste laddas omedelbart. Förmodligen behövs det många laddnings/urladdningsperioder innan ett mycket sulfaterat batteri kan väckas till liv igen.

Eftersom de metod som beskrivs ovan kräver mycket uppmärksamhet och medför risk för att batteriet onödigtvis lämnas i urladdat tillstånd (vilket är mycket illa när det gäller blybatterier) är nästa sätt förmodligen bättre.

Batteriet ansluts till återupplivaren med en underhållsladdare inkopplad parallellt. Du skall alltså inte använda en laddare som ger 7 A eller mer, utan en som lämnar maximalt 1 eller 2 A. Denna kan lämnas ansluten till batteriet permanent utan problem. Som vi redan nämnt är det möjligt att ansluta återupplivaren permanent inkopplad till batteriet i en bil. Det finns dock en risk för att pulser runt 50 V kan skapas i bilens kablar när den inre resistensen hos batteriet är hög, dvs när batteriet redan lid-

er av sulfatering. Dessa pulser är någonting som elektroniska apparater inte tycker om. Så, om det finns skäl att anta att batteriet befinner sig i ett dåligt tillstånd är det bäst att vara försiktig och inte ansluta återupplivaren permanent. Koppla då först bort batteriet från bilen innan du kopplar in återupplivaren.

Effekt?

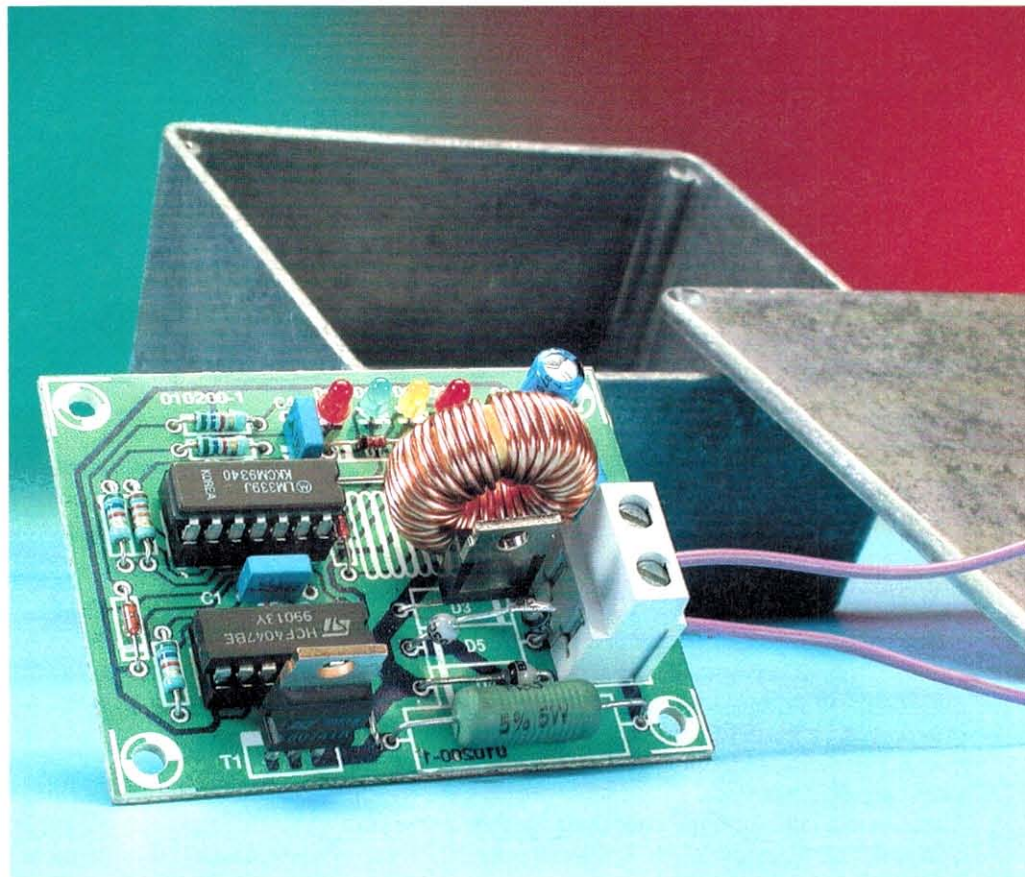
Med de tre lysdioderna är det mycket enkelt att se vilken effekt pulsladdningen har. Om sulferingen minskar ökar den aktiva ytan hos plattorna och batteriets inre resistens blir lägre. De laddningstoppar som levereras av återupplivaren blir lägre. Detta kan du se av färgen på den lysdiod som tänds. Med ett mycket dåligt batteri kommer den röda lysdioden att tändas från början. När pulsladdningen har haft effekt kommer den röda lysdioden att slockna och den gula att tändas i stället. Och när den gröna lysdioden till slut tänds är detta ett tecken på att batteriet är i ett ganska hyggligt till-

stånd igen. En sista koll med en voltmeter skall då visa att batterispänningen (utan laddaren förstås) ligger i sitt nominella värde på 12 V.

Denna kontroll kan utökas med en urladdningskoll. En känd belastning ansluts till batteriet och sedan kollar du helt enkelt hur lång tid som batteriet kan lämna lastström. Den användbara kapaciteten får du genom att multiplicera strömmen med tiden. När ett 12-V batteri belastas med en lampa på 50 W flyter en ström på ca 4 A. Om batteriet klarar att hålla lampan lysande i fem timmar är dess användbara kapacitet alltså 20 Ah.

Om kapaciteten fortfarande ligger lågt under det nominella värde som angivits av tillverkaren kan du fortsätta återupplivningen utan problem. Du skall inte förvänta dig alltför snabba förbättringar eftersom återställningen kan ta flera dagar, eller till och med veckor om batteriet är i mycket dålig kondition.

(010200-1)



Figur 3. Med tanke på risken för högfrekvensstörningar är det bäst att bygga in kretsen i en låda av metall.

I²C servointerface

styr åtta servon via I²C-bussen

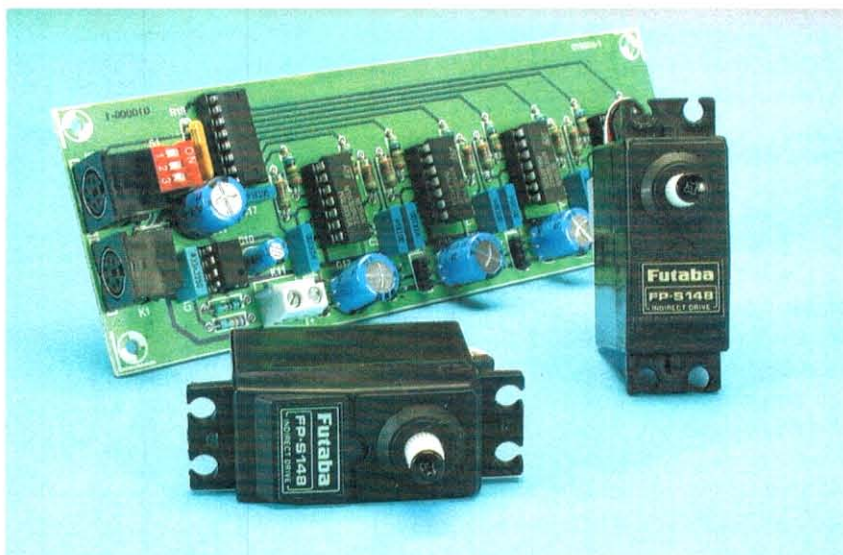
Design av C. Fuchs

Modellservon har blivit mycket bättre under de senaste åren och har nu både bättre vridmoment och hastighet, och de används mer och mer även i andra applikationer. Den krets som beskrivs här kan styra upp till åtta servon via den väletablerade I²C-bussen. Detta kan ökas upp till 64 servon med fler kort. Mjukvaran under Windows 95/98 visar kretsens egenskaper och tillåter enkel programmering på en PC med Visual Basic.

Att generera en noggrann pulsbreddmodulerad (PWM) signal för styrning av modellservon är ingen enkel uppgift för någon dator, speciellt inte om den har Windows som operativsystem. Och vi behöver knappast säga att problemen blir ännu värre när vi lägger till fler servon och behöver generera fler styr-signaler. I dessa situationer kan lite hårdvara ofta utföra underverk. Den lösning vi presenterar här använder ett enkelt kort som kan driva åtta servon styrda från en I²C-buss på ett parallellinterface, såsom det vi visade i nummer 3/1999. Med kortet kan åtta servon styras oberoende av varandra via kommandon som skickas över I²C-bussen. Med adressväljarswitcharna på kortet kan åtta av dessa kort parallellkopplas på samma I²C-buss och därigenom kan upp till 64 servon styras från samma I²C-buss.

Kretsöversikt

Blockscemat över servokontrollkretsen visas i figur 1. NE555-timern (IC6) är konfigurerad som en astabil multivibrator som genererar en puls varje 20 ms som triggar de åtta monostabila multivipporna, vilka bildas av fyra dubbla 556-timers (IC2-IC5). Pulsbreddsignalen på utgången från dessa IC kommer att styra rörelsen hos varje servo som är anslutet till den kanalen. En 1-ms bred puls kommer att flytta servoarmen till det ena ändläget av banan, medan en 2-ms lång puls för armen till det andra ändläget (vanligen 90 graders rörelse). Styrningen är proportionell och mel-



lanliggande pulsbredder flyttar armen till motsvarande mellanliggande position. Styrspänningen för var och en av de åtta monovipporna produceras av en 8-kanals DAC av typ TDA8444, som kan kopplas till en I²C-port och styrs med hjälp av mjukvarurutiner som körs i Visual Basic på en PC.

Snabba servomotorer med högt vridmoment kan generera ordentliga nivåer av elektroniska störningar och trots noggrann kretskortslayout och avkoppling av matningsspänningen så har detta lett till felfunktioner i

TDA8444 vid några tillfällen. Om du planerar att använda flera servon är det bäst att mata servona för sig från en extern spänningskälla och använda de +5 V som tas från I²C-kontakten enbart för kretsen på detta interfacekort.

Kretsdetaljer

IC1 är huvudkomponenten i krets-schemat som visas i figur 2. Det är en 8-kanals digital-till-analog omvandlare (DAC). Spänningsnivån vid de åtta analoga utgångarna kan

utgångspulsen 2 ms. Dessa två pulsbredder representerar det största och det minsta utslaget hos armen och skall ge en rörelse på 90 grader hos den aktiverade armen. I praktiken kan komponenttoleranser ge en offset, eller minska rörelsen, och det kan därför vara nödvändigt att ändra komponentvärdena för att uppnå exakt 90 graders utslag om detta är viktigt för din applikation. Pulsbreddssignalen på 1 ms eller 2 ms till varje servo måste repeteras ungefär varje 20 ms. En stabil multivippa bildad av IC6 ger denna signal genom att generera en kort puls på 1 ms varje 20 ms som samtidigt triggar var och en av de åtta monovipporna. Utgångspulserna tas sedan till de trepoliga utgångskontakterna på kretskortet tillsammans med matningslinjerna för att driva servona.

Olika servotillverkare har tidigare alltid tenderat att använda sina egna stifttilldelningar för servokontakterna och om du har något av dessa äldre servon kan kanske stiftkonfigurationerna i figur 3 vara till hjälp. Var försiktig här, eftersom ett felaktigt kopplat servo kan bli ett dyrbart misstag. Tack och lov håller denna situation på att bli bättre och de flesta moderna servon är nu stiftkompatibla och kan bytas mot varandra.

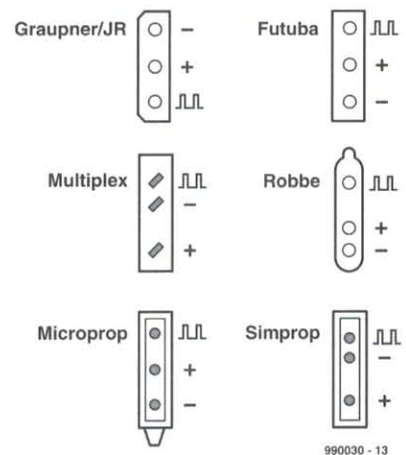
Som vi sagt tidigare kan servon producera brus och kretskortslayouten för denna krets

har därför breda matningslinjer till servokontakterna och stora avkopplingskondensatorer för att minska dessa störningar.

Själva bygget

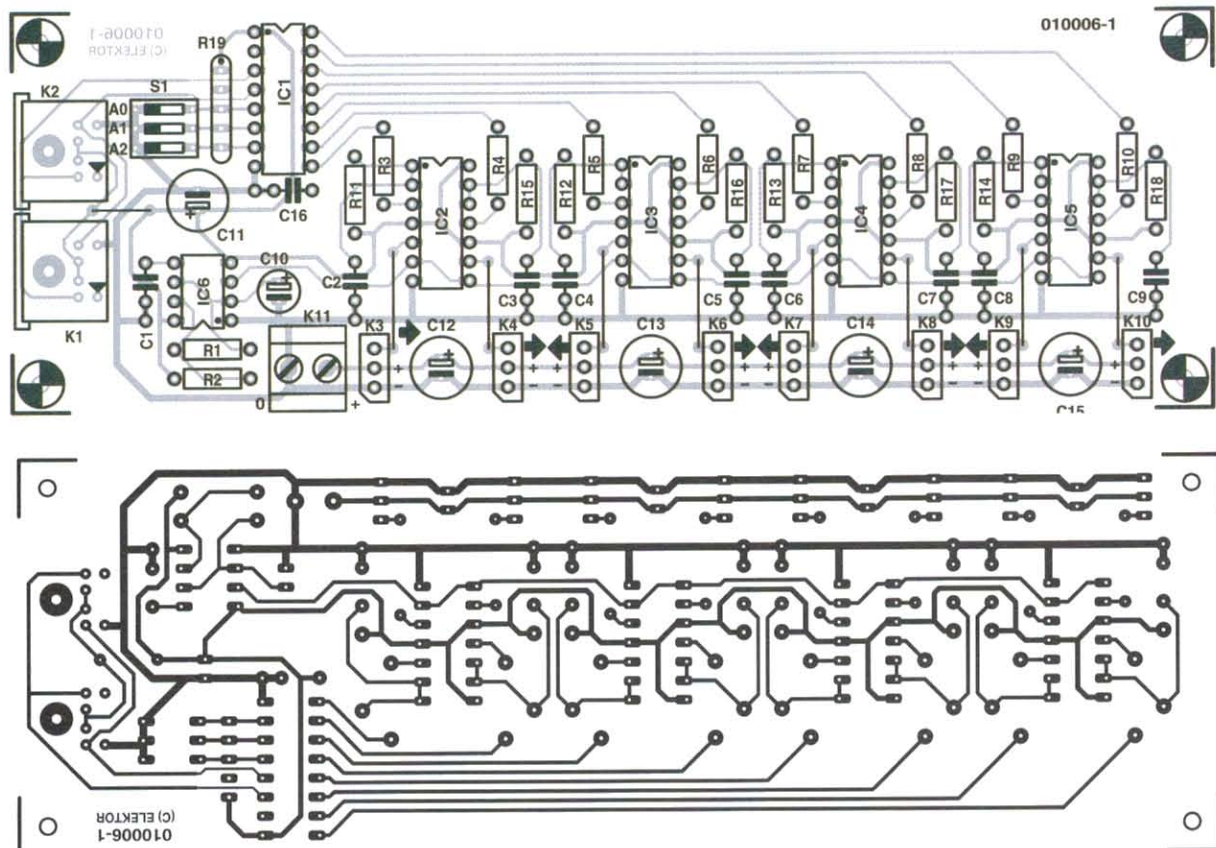
Kretskortslayouten visas i figur 4. Börja bygga kortet genom att montera de mer robusta komponenterna först, dvs trådbyglarna, IC-hållarna, kontakterna och stiftlisterna. Montera därefter de passiva komponenterna. Tänk på att montera motståndsnätet R19 åt rätt håll. Innan IC-kretsarna sätts in i sina hållare är det en bra idé att ansluta kortet till +5 V och kolla att det finns spänning på alla IC-kretsarnas spänningsben. Om allting är OK kan du montera IC-kretsarna och ansluta kortet till I²C-interfacet. Om du har många servon anslutna till kortet är det viktigt att använda en extern spänningskälla på +5 V och inte förlita sig på matningen via I²C-bussen.

Efter uppstart skall du kolla att utgångarna från de åtta DAC-kanalerna är vid 0 V (0,26 V max.). Anslut ett oscilloskop till utgången från IC6

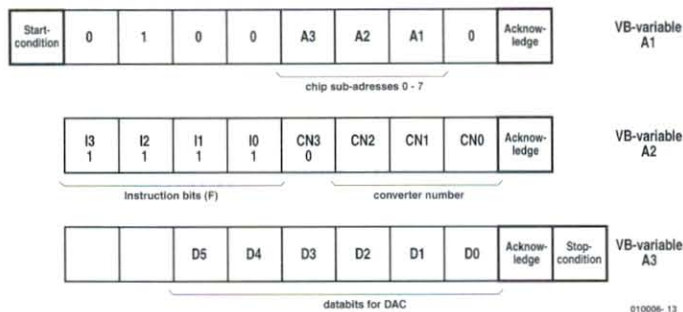


Figur 3. Stifttilldelningen hos några populära servon.

(ben 3) och kolla en den korta 1 ms pulsen upprepas var 20 ms. När ett servo pluggas in i kortet skall dess utgångsarm flytta till ett ändläge. Om alla funktioner fungerar korrekt så skall kretsen upp till TDA8444 vara OK.



Figur 4. Enkel layout på ett enkelsidigt kort (finns inte att köpa färdigt från oss).



Figur 5. Formatet för programmering av TDA8444.

Mjukvaran

Demomjukvaran `i2cservo.exe` är ett 16-bitars program och kan endast köras under Windows 95/98, inte under NT. Filen `iic.dll` innehåller mjukvaran för interfacet till PC-hårdvaran och måste placeras i samma mapp som applikationen eller i mappen `System` i Windows. Visual Basic 3 användes för att skriva mjukvaran. Alternativt i menylistan tillåter val av parallellport LPT1 (378hex) eller LPT2 (278hex) och också den subadress som avkodas på kortet med DIP-switchen S1. En 7 kommer att prata med ett interfacekort som har alla tre switcharna anslutna till +5 V medan en 0 talar med ett kort som

har alla tre anslutna till jord.

De åtta rullningslisterna motsvarar de åtta utgångskanaler från DAC-en och därmed också servona. Genom att klicka och dra med musen på en av dessa rullningslistor kommer motsvarande servo att röra sig. Detta enkla exekverbara program skrivet i Visual Basic finns att hämta, tillsammans med källkoden och filen för kretskortslayouten, gratis från vår Downloadsida. Detta program är en bra startpunkt för utveckla och relativt enkelt att modifiera och expandera.

När programmet först startar är den förvalda subadressen för chipet satt till 7 (alla adressbenen kopplade till +5 V).

```
ChipAdress = Val(Text1.Text)
```

och den använda porten är LPT2.

```
Lpt = Lpt2
```

LPT1 och LPT2 är definierade i `Modul1.bas` och har värdena `&H378` och `&H278` respektive. Båda inställningarna (7 och LPT2) kan naturligtvis ändras.

```
(SubText1_Change,  
SubParallel11/2_Click).
```

Därefter kommer en kontroll för att se om I²C-interfacet är anslutet.

```
ErrorMsg=TestConnected(LPT)
```

Motsvarande felmeddelanden (om det finns något) kommer att skickas ut.

```
SubVScrollX_Change()
```

Under användning läses varje rörelse hos rullisten med `X = 7` till 0, vilket motsvarar rullisterna 1 till 8. Värdet på inställningen läses (i området 0 till 63) och tilldelas till variabeln A3.

```
DacValue = VScrollX.Value
```

Det maximala värdet MAX hos de vertikala rullisterna är satt till 63 och det finns inget behov för att ändra detta värde. De tre minst signifikanta bitarna i A2 innehåller koden för DAC-numret (0 till 7), vilket motsvarar den rullisten. Bit 4 är alltid satt till 0 och alla de högre fyra bitarna är alltid satta till 1 (=240 = Instruction F, sändning av data utan en adress kommer alltid att skriva till samma DAC). Variabeln A1 innehåller bitarna 2, 3 och 4, vilka definierar chipets subadress. De övre fyra bitarna innehåller den fasta chipadressen 0100 (64d).

De tre variablerna A1,A2 och A3 som visas i figur 5 representerar det datatelegram som skickas över I²C-bussen till DAC-en varje gång en rullist flyttas. De rutiner som behövs för detta finns i `iic.dll`. Dessa funktioner definieras under GLOBAL. Att skicka information över I²C-bussen börjar med

```
ErrorMsg = StartCon( LPT)
```

för att skicka I²C-startkonditionen och sedan

```
ErrorMsg=Transmit(LPT,A1..A3)
```

för att sända de tre byten. Efter sändning frisläpps I²C-bussen igen med hjälp av stoppkommandot

```
ErrorMsg = StopCon(LPT)
```

Alla detekterade fel tilldelas de fem felvariablerna `Er1` till `Er5` och visas i det undre statusfönstret. Databladet för TDA8444 kan hämtas från http://www.semiconductors.com/acrobat/datasheets/TDA8444_3.pdf

Felkoder (error)

- | | | |
|---|------------|---|
| 0 | Error 0000 | Inget fel detekterades. |
| 1 | Error 0001 | I ² C-bussen ej ledig. |
| 2 | Error 0002 | Ingen bekräftelse mottagen. |
| 3 | Error 0003 | Kunde ej generera stoppkondition. |
| 4 | Error 0004 | Spänningsfel på I ² C-bussen. |
| 5 | Error 0005 | I ² C-interface inte anslutet. |
| 6 | Error 0006 | Ett odefinierat fel har uppstått. |

KOMPONENTLISTA

Motstånd:

R1 = 100Ω
R2 = 27kΩ
R3-R10 = 6kΩ204, 1%
R11-R18 = 18kΩ27, 1%
R19 = 10kΩ 4-pol SIL-nät

Kondensatorer:

C1 = 33nF
C2-C9,C16 = 100nF
C10 = 1μF 16V radial
C11-C15 = 470μF 16V radial

Halvledare:

IC1 = TDA8444
IC2-IC5 = NE555
IC6 = NE555

Övrigt:

K1,K2 = 6-pol mini-DIN sockel
K3-K10 = 3-pol stiftlist
K11 = 2-pol kopplingsplint för kretskort, raster 5mm
Diskett med mjukvara 010006-11

Omvandlare för video till S-VHS

Från komposit (CVBS) till C/Y

Design av W. Foede

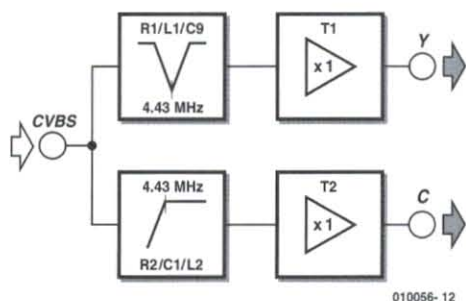
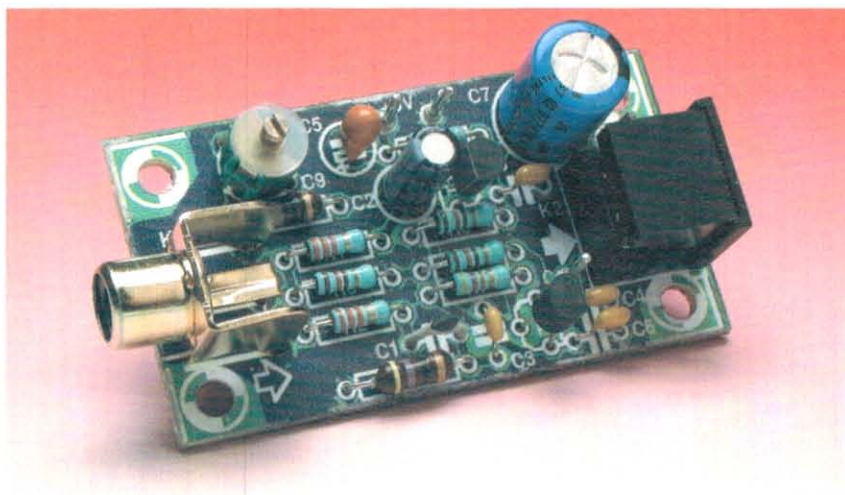
foede.koeln@web.de

Har du problem med att ansluta en standardvideo till din S-VHS utrustning? Då behöver du denna lilla nätta krets. Den tar en sammansatt (komposit) videosignal och omvandlar denna till de Y/C-signaler som används av S-VHS. En framtagen kretskortslayout förenklar byggandet.

En standard videosignal, (också känd som Composite Video Baseband Signal eller, som det förkortas, CVBS) innehåller den svart/vita och bildsynkroniseringsinformationen tillsammans med färginformationen i en signal. S-videosignaler (också kända som Y/C) skiljer sig något i det att dessa två delar av bildsignalen lämnas som två separata signaler. Y är den luminanssignal (ljus) som innehåller både svart/vit och synkinformation med C är den krominanssignal som innehåller färginformationen.

Apparater som använder Y/C-signaler har bättre bildupplösning (större bandbredd), mindre 'cross-colour' störningar (de färgförskjutningar som uppstår om någon har på sig en rutig kavaj t.ex.) samt färre filter för signalen att klämma sig igenom.

Med den krets som beskrivs här kan en stan-



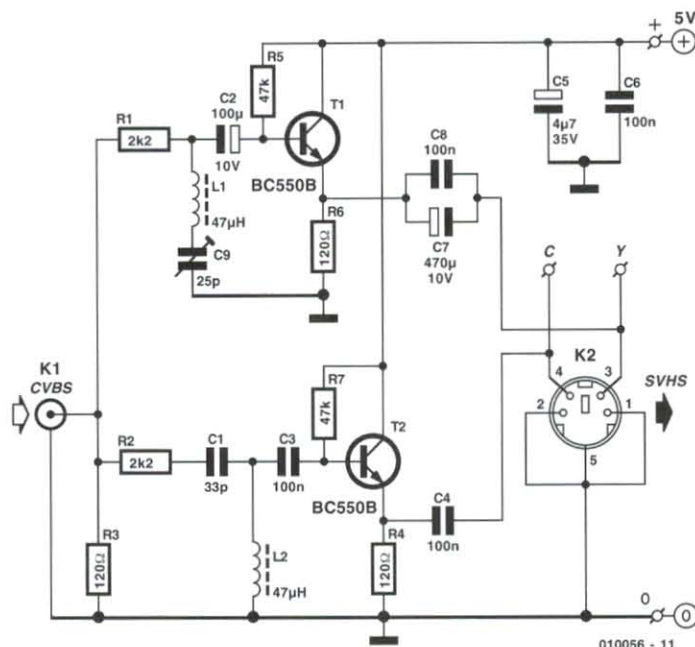
Figur 1. Blockschemat visar delningen av CVBS-signalen i Y och C-komponenter.

dard CVBS-signal anslutas på ingången till en S-VHS apparat. Den kommer dock inte att förbättra bildkvaliteten. Kretsen behöver en spänningskälla på +5 V som kan lämna ungefär 50 mA.

Bandbredden hos en komposit videosignal sträcker sig från 50 Hz upp till 5 MHz. Färginformationen blandas in i signalen på en underbärsvåg vid 4,43 MHz med en bandbredd på -1,2 till +0,6 MHz. Jobbet för vår krets är att tydligt separera Y och C signalerna från varandra (en standard-TV måste kunna göra detta också). Den metod som används här är en av de enklaste. Ett notch (eller

bandstopp) filter avstämt till underbärsvågens frekvens används för att ta bort C-informationen från Y-informationen. En nackdel med denna metod är att bildinformationen utsätts för en bandbreddsminskning från 5 MHz till ca 4 MHz, vilket ger något sämre bildupplösning. C-signalen produceras genom att skicka kompositsignalen genom ett 4,43 MHz högpasfilter för att ta bort det mesta av Y-komponenterna (med lägre frekvens). En del Y-komponenter blir dock kvar och orsakar synliga cross-coloureffekter i bilden.

Moderna TV-apparater utför denna Y/C-separation med hjälp av kamfil-



Figur 2. Kretsschema.

ter och en fördröjningslinje. Fördelen är en större bandbredd och inga cross-coloureffekter. S-VHS videobandspelare använder också denna teknik så att Y-komponenten hos den inspelade signalen får en bättre bandbredd.

Blockschemat visas i figur 1. C-delen av signalen filtreras ut från komposit-signalen med hjälp av en RLC högpasfilterkrets, medan Y-delen använder en RLC-notch, eller bandstopfilter, för att ta bort C-informationen. Transistorerna T1 och T2 fungerar som buffrar och producerar lågimpedanta Y och C-utgångar.

Det kretsschema som visas i figur 2 följer blockschemat mycket nära. R2, C1 och L2 bildar ett högpasfilter för att återvinna C-signalen. Filterutgången köpplas via C3 till basen på transistor T2. Denna transistor har en gemensam-emitterkonfiguration och buffrar filtret medan den samtidigt ger en lågimpedansutgång via C4 till S-VHS kontakten.

Det bandstopfilter som används i Y-vägen består av R1, L1 och C9. L1 är en fast standardspole på 47 μ H medan C9 är en justerbar trimmer. Därmed kan bilden optimeras i viss grad. Y-signalen innehåller frekvenskomponenter som går ner till 50 Hz och signalkopplingskondensatorerna C2 och C7 behöver därför ett större värde än de för C-signalen, annars

skulle signalens lågfrekvensdelar dämpas. De ac-kopplade lågimpedanta signalerna C och Y skickas ut på stiften 4 respektive 3 på S-VHS kontakten K2. Strömförsörjningens avkopplingskondensatorer C5 och C6 fullbordar kretsen.

Ingångs och utgångsimpedansen hos denna krets avviker från det optimala värdet, men valdes för att se till att utgångens signalnivåer var korrekta. Vid 75 Ω skall signalnivåerna vara 0,3 Vpp (burst-amplitud) för C och 1,0 Vpp för Y.

Kretskortslayouten visas i figur 3. Som i alla HF-kretsar är det viktigt att komponentbenen så korta som möjligt när du monterar dem på kortet. Det finns ett jordplan kring alla ledare och det är därför viktigt att du är försiktig vid lödningen så det inte blir någon kortslutning. I stället för riktiga Hosiden S-VHS kontakter har vi specificerat en standard mini-DIN sockel för kretskortsmontering. Den kan lätt ta emot en Hosidenplugg om du tar bort de små stoppinnarna av plast först.

Under inställningen skall trimmern C9 justeras att ge minsta möjliga moiré-effekt (färgmönster) i bilden. Kretsen har testats med framgång på många olika S-VHS ingångar. I allmänhet var bandstopfiltret i Y-vägen nödvändigt, men med ett WinTV-kort märktes ingen skillnad

om filtret var inkopplat eller inte. I detta fall kan du helt enkelt gå förbi Y-kretsen och använda komposit-signalen som Y-signal.

En realitetskontroll till slut. Som vi redan sagt kan kvaliteten på den S-VHS kompatibla utgångssignal som genereras här inte bli bättre än den komposita ingångssignalen. För att få riktig S-VHS kvalitet på bilden måste du ha en 'genuin' S-VHS signal ansluten direkt till Y/C-ingången på din monitor, TV eller videobandspelare.

(010056-1)

KOMPONENTLISTA

Motstånd:

R1, R2 = 2k Ω
R3, R4, R6 = 120 Ω
R5, R7 = 47k Ω

Kondensatorer:

C1 = 33pF
C2 = 100 μ F 10V radial
C3, C4, C6, C8 = 100nF keramik
C5 = 4 μ F 35V tantal
C7 = 470 μ F 10V radial
C9 = 25pF trimkondensator (3-25 p)

Spolar:

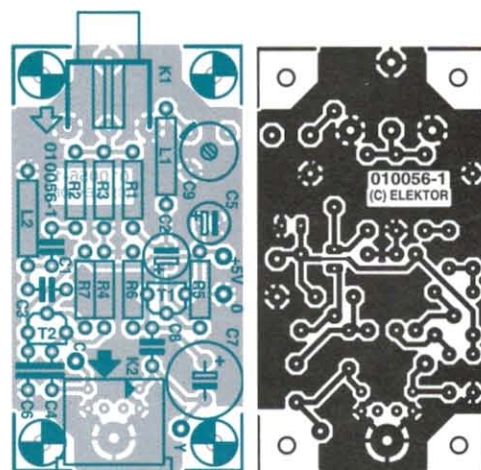
L1, L2 = 47 μ H

Halvledare:

T1, T2 = BC550B

Övrigt:

K1 = Cinch-kontakt för kretskort
K2 = 6-pol Mini-DIN sockel, för kretskort, vinklade stift



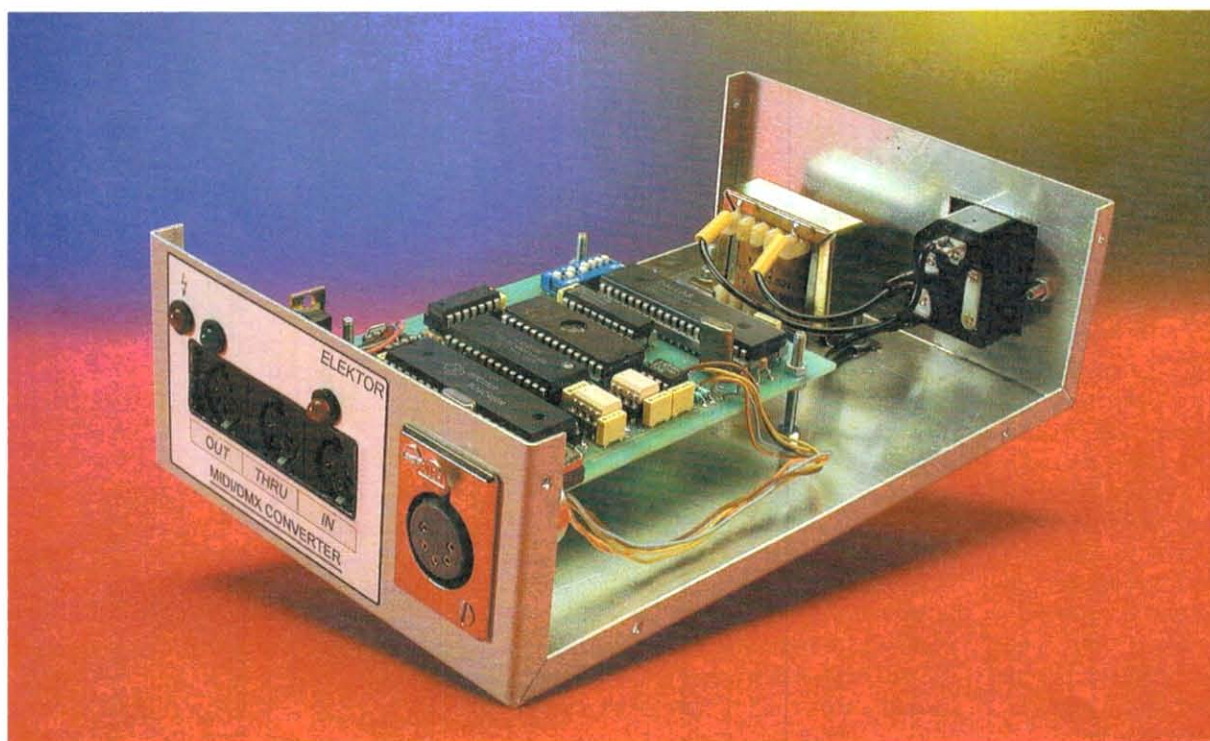
Figur 3. Kretskortslayout. Jordplanet är typiskt för en HF-krets. Kortet finns tyvärr inte att köpa färdigt.

MIDI-till-DMX interface

styr ditt DMX-nät från en sequencer!

Design av B. Bouchez

För inte så länge sedan beskrev vi principen för DMX512-baserad fjärrstyrning, som gör att automatiserade projektörer eller strålkastare kan styras på upp till 512 kanaler via en RS485-länk. Varje DMX-system har en master och ett antal slavar som svarar på DMX-kommandon.

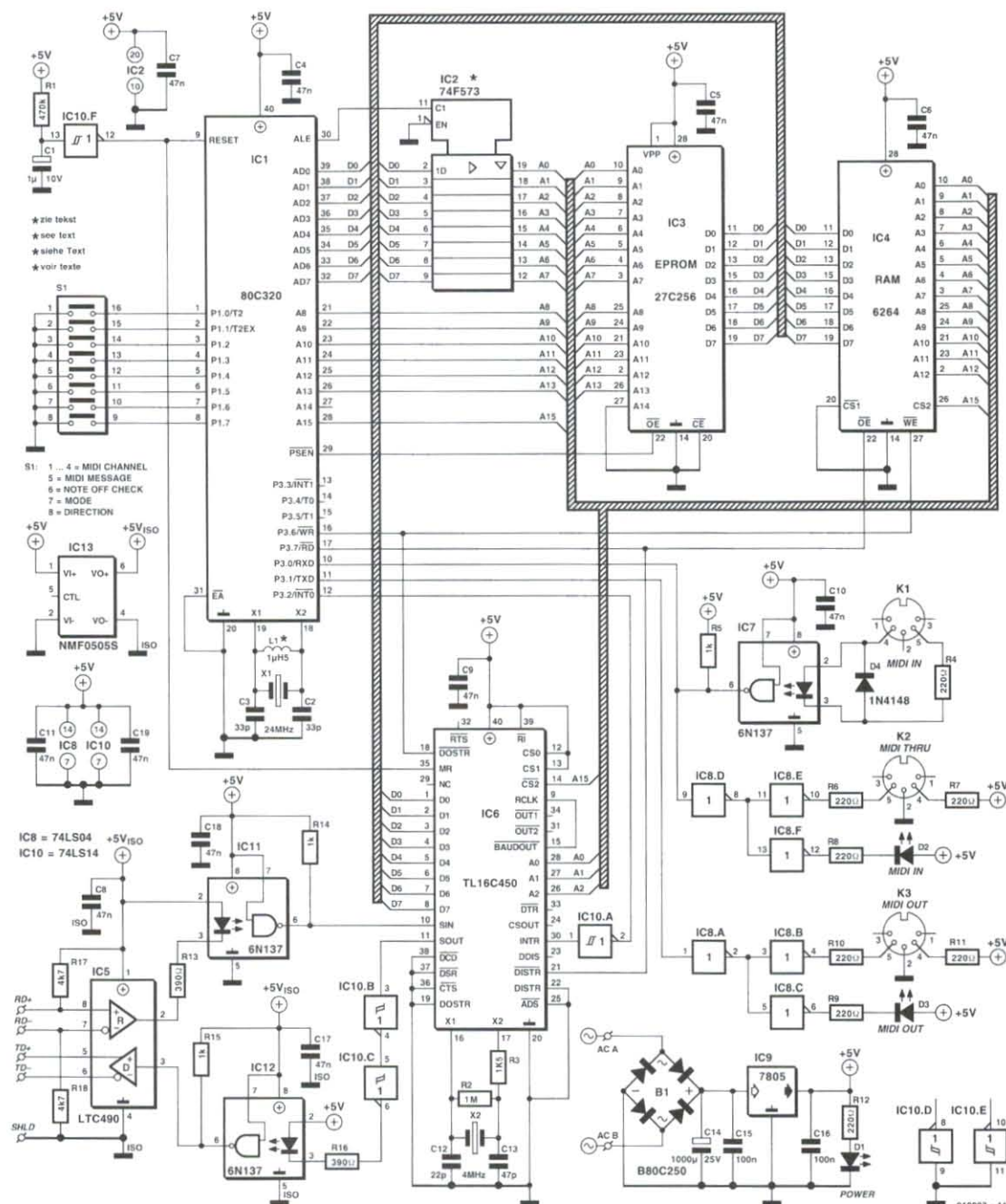


Egentligen finns det två sätt att realisera en DMX512 master, antingen genom att använda en specialkontroll (som MARTIN2518 till exempel) eller med hjälp av mjukvara på en PC. Fördelen med det första sättet ligger i enkelheten och det låga priset. Nackdelen är att möjligheterna, speciellt när det gäller pro-

grammering och minne, i allmänhet är begränsade. Dessutom är användarinterfacet hos några av dessa kontrollers, milt uttryckt, i vissa fall mycket originella, för att inte säga udda.

Men, som för det mesta, med en PC

tillsammans med ett DMX-interface kan du göra i stort sett allting och möjligheterna är oslagbara. Det finns förmodligen inte en enda större konsert, TV-show eller någon annan stor publik händelse utan någon form av PC-styrd scenbelysning.



Figur 1. Detta kretsschema över MIDI-till-DMX interface visar ett fristående mikrokontrollersystem.

En PC-lösning har dock också ett par nackdelar. För det första behövs ett specifikt interface och för det andra är lämplig mjukvara för dessa interface både dyr och krånglig att använda för icke-specialister.

Med det interface som beskrivs här kan du, till en rimlig kostnad, styra vilket DMX-system som helst från vilken MIDI-sequencer som helst, på en PC, en Mac eller fristående. I stället för att kontrollera ljud så styr sequencern DMX-nivåer, färgbyten,

gobo-anrop etc. Detta interface integrerar en annan mycket användbar funktion, en DMX-till-MIDI omvandlare som gör att du kan titta på vad som händer på din DMX-kontroll och identifiera fel om sådana skulle uppstå. Dessutom kan denna funktion användas för att fånga in sekvenser som genereras av DMX-kontrollern i din egen sequencermjukvara.

Till sist har en serie mjukvaruprogram designats speciellt för detta interface, inklusive ett drivrutinsbibliotek så att

du enkelt kan skapa din egen mjukvara under Windows. Denna verktygssats omfattar också två testprogram.

Konstruktören av detta projekt har skrivit två kraftfulla, men lättanvända, ljusstyrningsprogram som använder det interface som beskrivs här.

Kretsen

Som du förmodligen redan räknat ut är interface byggt runt en mikrokontroller som tar hand om de olika dataflödena. Från början

valde vi den välkända 80C32, men efter att ha testat interfacet ute på fältet upptäcktes att mjukvaran belastade CPU:n alldeles för mycket för en pålitlig funktion. Till slut valde vi en 80C320 som är både ben- och mjukvarukompatibel med 80C32-an, men som är tre till fyra gånger snabbare tack vare en helt omdesignad kärna. Inte desto mindre, som du kommer att se längre fram, är det möjligt att använda en 80C32 i en 'light' version av interfacet. Som stöd för den snabba dataöverföringen (250 kbit/s på DMX och 32,25 kbit/s på MIDI) klockas mikrokontrollern med 24 MHz. Även om den aktuella versionen av mjukvaran fungerar perfekt, så skall den i alla fall kunna utvecklas och expanderas i framtiden. Följaktligen används ett externt EPROM (IC3) för att hålla programmet. Bli inte förvånad när du ser en adresslinje dragen till jord i stället för till adressbussen på 80C320. Även om mjukvaran också kan laddas in i en 27C64 eller en 27C128 så är dessa EPROM i allmänhet långsammare och dyrare än 27C256. Eftersom inte allt minne har använts har icke använda ben dragits till jord helt enkelt för att underlätta kretskortsdesignen. Notera att EPROM-hastigheten måste vara 120 ns, eller snabbare, på grund av processorns hastighet. Av samma skäl skall data/adressdemultiplexern, IC2, normalt vara en 74F573. En 74HCT573 behövs endast i några få sällsynta fall, vilket vi återkommer till längre fram.

Eftersom den mängd data som skall hanteras överskrider processorns inbyggda minneskapacitet så används ett externt RAM (IC4). Som vi kommer att förklara längre fram är det möjligt att utelämna detta minne till priset av en del interfacemöjligheter. Som med EPROM-et så kan ett mindre RAM användas, men dessa 'mindre' minnen är svåra att få tag på, speciellt komponenter med snabba accesstider. Vi använder ett relativt stort minne här, som är ganska enkelt att få tag på och som inte kostar speciellt mycket. Även här är det viktigt att använda ett 120 ns eller snabbare minne.

MIDI-interfacet är byggt direkt runt mikrokontrollern och använder dess interna serieport. Det fysiska interfacet runt IC7 och IC8 är klassiskt och överensstämmer till fullo med MIDI-specifikationerna. De två lysdioderna har lagts till för att visualisera dataflödet på MIDI IN och MIDI OUT. Denna typ av utbyggnad kostar praktiskt taget ingenting, men kan vara mycket praktisk om det skulle uppstå problem mellan interfacet och sequencern.

På DMX-sidan är interfacet byggt runt IC6, en snabb UART (Universal Asynchronous Receiver Transmitter) av typ 16C450. Detta chip är en utveckling av 'gamla goda' 8250, som har varit en del av PC-ns serieport sedan be-

gynnelsen. 16C450 och 16C550 kan enkelt uppnå datahastigheter på över 250 kbit/s där 8250-an är begränsad till 19200 bit/s (även om den kan komma upp till 115,2 kbit/s i praktiken).

Denna UART är ett 'måste' i denna krets eftersom den förenklar mikrokontrollerns arbetet i högsta grad. UART-kretsen har en inbyggd Break-detektor, som används för DMX-synkronisering. På 80C320-kretsens serieport skulle en sådan detektering annars varit tvungen att implementeras i mjukvaran.

UART-en har sin egen kristallosillator och en programmerbar delare som gör att i stort vilket baud-rate som helst kan ställas in. Notera att en klocksignal som är 16x baudraten kan hittas på RCLK-benet hos chipet. På detta kort hittar du en frekvens på 4-MHz (16x250 kbit). Detta är praktiskt när kortet skall testas.

Somm du ser är IC10 en Schmitt-trigger av typ 7414. Denna IC används för att skapa resetpuls för UART-en och mikrokontrollern, vilka av mjukvaruskäl måste starta samtidigt. En 7404, som har samma benkonfiguration men inget specificerat triggerbeteende, kan ge allvarliga problem när kortet startas. Det hör till historien att konstruktören av detta projekt använde flera dagar på en av prototyperna med att jaga en icke-existerande mjukvarubug, endast för att upptäcka att UART-en startade efter mikrokontrollern.

En LTC490 omvandlare, med separata sänd och mottagningsbuffrar, används för att transformera TTL-nivåerna (används av UART-en) till RS485-nivåer. Motstånden R17 och R18 är biaskomponenter på DMX-linjerna och används för att undvika falska 'Break' detekteringar när en DMX-ingång inte är ansluten. Utan dessa motstånd skulle en falsk Breakdetektering orsaka felaktig synkronisering.

Speciellt i DMX-nät där avstånden är långa finns det en reell risk för att det uppstår jordslingor. För att förhindra detta är DMX-sidan optiskt isolerad från resten av kretsen. Eftersom DMX är ganska snabbt är det viktigt att använda höghastighets 6N137 optokopplare.

Denna del av kretsen försörjs från en liten DC/DC-omvandlare. Det är möjligt att sänka priset på interfacet

genom att utelämna omvandlaren och optokopplarna. I så fall skall du ansluta SOUT-benet på UART-en direkt till DIN-ingången (ben 3) på LTC490 och SIN-benet på UART-en till DOUT (ben 2) på LTC490. Kom dock ihåg att denna konfiguration i hög grad ökar risken för att interfacet skall skadas om det skulle ske en energiretur på DMX512-linjerna. Och vår erfarenhet säger att detta sker oftare än vad man kanske tänker sig.

Till sist, interfacet konfigureras med hjälp av en oktall (8-polig) DIP-omkopplare, S1, som är direkt, kopplad till processorns P1-port. Det behövs inga pull-upmotstånd eftersom dessa finns integrerade i P1-porten.

Mjukvaran

Värdet på ett mikrokontrollerdrivet kort beror helt och hållet på den mjukvara som finns för kortet. När det gäller det här interfacet så handlar det om en hel del mjukvara! För att ge dig en idé så kan behandlingen av DMX-mottagning använda upp till 60% av processorns beräkningskapacitet. På MIDI-sidan kan det gå upp till 45% på en tungt lastad (=mycket upptagen) MIDI-linje. Dessutom är mottagning av data endast en del av de funktioner som utförs av processorn, den jobbar samtidigt i bakgrunden med att överföra data.

Innan vi går in på de interfacets olika arbetssätt (mod) skall vi titta lite mer allmänt på det.

På MIDI-sidan känner vårt interface till två typer av meddelanden: Note On (90h) och Control Change (B0h). De meddelandetyper som används både för MIDI-till-DMX som för DMX-till-MIDI väljs med hjälp av switchen S1-5. När switchen är till (ON) använder interfacet Control Change. I det andra fallet (OFF) används Note On.

Dessa två MIDI-meddelande valdes eftersom de omfattar två databyte, det första för noten eller kontrollsiffran och det andra för velocitetsvärdet eller kontroll.

Eftersom en MIDI-kanal kan sända 128 noter eller kontroller kommer det första bytet att användas för att identifiera motsvarande DMX-kanal. Det andra databytet används för att ange DMX-värdet för den kanal som

Tabell 1. MIDI/DMX världens överensstämmelse

MIDI-värde	DMX-värde
0	0
0	1
1	2
1	3
2	4
2	5
3	6
3	7
.	.
.	.
126	252
126	253
127	254
127	255

Tabell 2. Val av mod med S1-7 och S1-8

S1-7 (Grundmod/ 4 kanaler)	S1-8 (Riktning)	Mod
OFF	OFF	1
OFF	ON	2
ON	OFF	3
ON	ON	4

valdes med det första bytet. Då MIDI endast stöder 128 värden (DMX stöder 256) utförs en skalningsoperation genom att trunkera (ta bort) den minst signifikanta biten och följande rotation. Förhållandet mellan MIDI och DMX-värden ges i tabell 1. Switchen S1-6 godkänner eller förkastar igenkännandet av Note Off meddelanden (för att det skall vara vettigt för denna switch skall interfacet vara i Note On mod, inte Control Change). Om S1-6 är sluten kommer interfacet att behandla alla Note Off som tas emot på en konfigurerad MIDI-kanal som ett blackout-kommando (släck) för motsvarande DMX-kanal. Tack vare denna funktion blir det i allmänhet enklare att program-

Tabell 4. Val av MIDI-kanal med S1-3 och S1-4 för mod 3 och 4

S1-3	S1-4	MIDI-kanaler
OFF	OFF	1, 2, 3, 4
ON	OFF	5, 6, 7, 8
OFF	ON	9, 10, 11, 12
ON	ON	13, 14, 15, 16

mera MIDI-sequencers. I mod 2 och 4 har S1-6 en annan funktion som gör att ett speciellt mod, som kallas för 'Delta' kan anropas. I Deltamod är det endast de MIDI-värden som ändrats sedan föregående period som returneras till MIDI. Att detta minskar trafiken på MIDI-länken torde vara helt uppenbart. Vad som är viktigare är dock att det tillåter DMX-sekvenser att registreras på en MIDI-sequencer.

Interfacets protokollomvandlare kan användas under fyra mod, vilka väljs med S1-7 och S1-8, se tabell 2.

Mod 1: omvandlaren transformerar MIDI-data på den kanal som valts med S1-1 till S1-4 (se tabell 3) till 128 DMX-värden.

Mod 2: omvandlaren transformerar de först mottagna 128 DMX-kanalerna till MIDI-meddelanden och sänder dem till MIDI OUT till den MIDI-kanal som valts med S1-1 till S1-4. MIDI-meddelandets innehåll definieras av S1-5 (se nedan). Notera att MIDI-transmissioner är kontinuerliga så att data uppdateras hela tiden.

Mod 3: omvandlaren styr 512 DMX-kanaler på DMX-utgången (detta kallas en 'DMX-värld'). Eftersom det inte är möjligt att skicka 512 värden på en enda MIDI-kanal kommer interfacet att använda fyra på varandra följande MIDI-kanaler, där var och en av dessa styr 128 värden. I detta mod ger endast S1-3 och S1-4 baskanalen för MIDI, de övriga är de följande kanalerna. Möjliga värden för MIDI-uppsättningen visas i tabell 4.

Mod 4: i detta mod transformerar omvandlaren de 512 mottagna DMX-kanalerna till MIDI-meddelanden och skickar dem kontinuerligt till MIDI OUT. Som i föregående mod används fyra på varandra följande MIDI-kanaler, valda med S1-3 och S1-4 (se tabell 4). Som i mod 2 väljs MIDI-meddelandetyper med switch S1-5.

Notera att i samtliga fall känner interfacet igen, och skickar DMX-rutor, enbart när 'Start Code' är lika med 0.

Att bygga interfacet

Även om schemat kan verka ganska komplicerat så är det relativt enkelt att bygga detta interface, mycket

Tabell 3. Val av MIDI-kanal med S1-1 till S1-4 för mod 1 och 2

S1-1	S1-2	S1-3	S1-4	MIDI-kanal
OFF	OFF	OFF	OFF	1
ON	OFF	OFF	OFF	2
OFF	ON	OFF	OFF	3
ON	ON	OFF	OFF	4
OFF	OFF	ON	OFF	5
ON	OFF	ON	OFF	6
OFF	ON	ON	OFF	7
ON	ON	ON	OFF	8
OFF	OFF	OFF	ON	9
ON	OFF	OFF	ON	10
OFF	ON	OFF	ON	11
ON	ON	OFF	ON	12
OFF	OFF	ON	ON	13
ON	OFF	ON	ON	14
OFF	ON	ON	ON	15
ON	ON	ON	ON	16

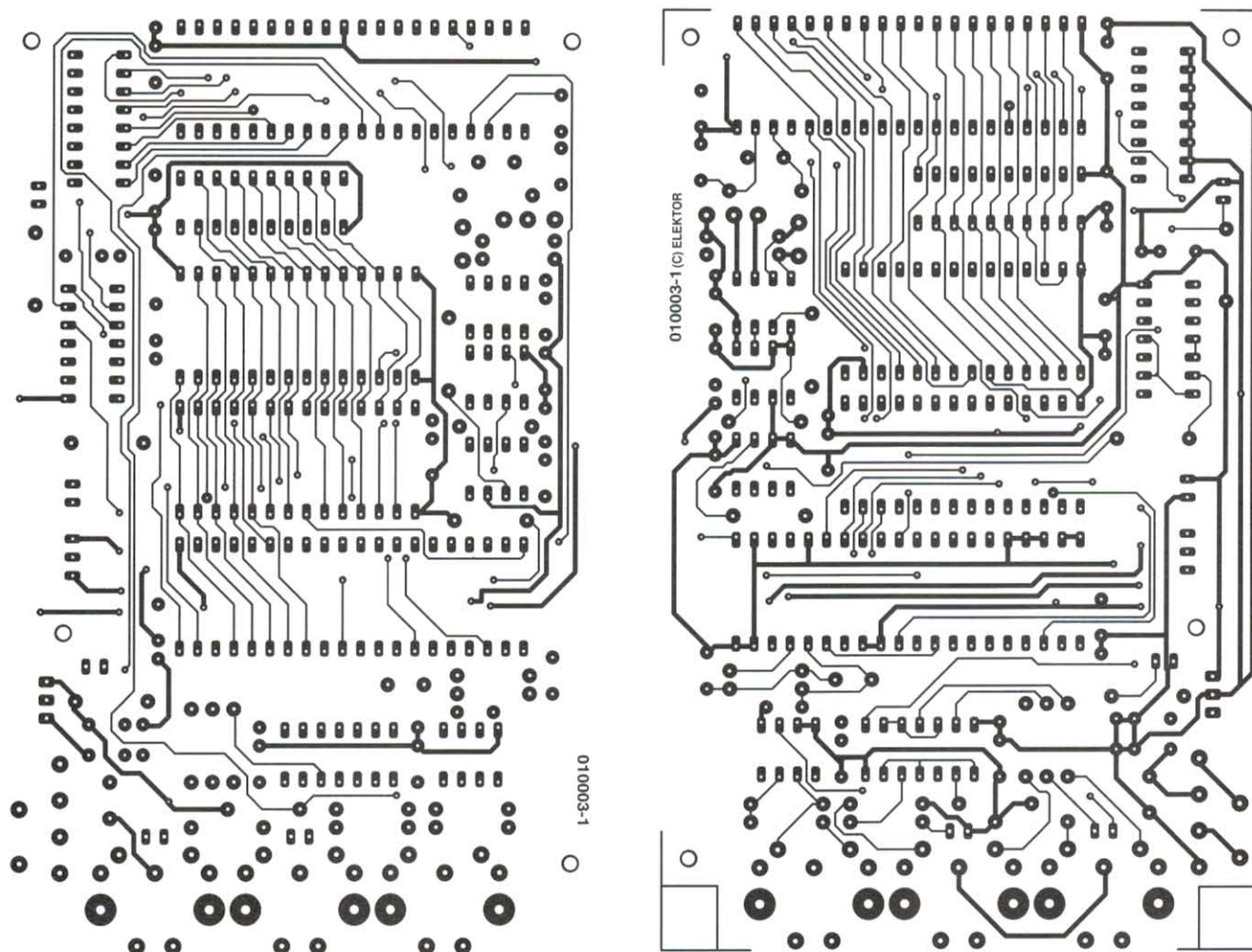
tack vare det dubbelsidiga kretskort som tagits fram. Vi rekommenderar att du använder hållare av hög kvalitet för alla IC-kretsar.

Som vi redan sagt är det möjligt att bygga en 'light' version. I detta fall kan du endast använda Mod 1 (som är det vanligaste och mest användbara modet). För denna version behövs inte SRAM IC4, som kan utelämnas. Eftersom DMX-mottagning inte är aktiv i mod 1 behövs inte heller IC11, R13 och C18. Sist, men inte minst, du kan använda en 80C32, billigare än en 80C320.

Vissa återförsäljare har problem med att specificera maxfrekvensen hos de IC de säljer. Ännu värre, det finns de som säljer 16 eller 20 MHz komponenter och hävdar att dessa är kompatibla med en 24 MHz klocka. En återförsäljare sålde en 8052 (som är en NMOS och begränsad till 12 MHz) som en 80C32 (CMOS) och påstod att den bara blev lite varmare. En processor specificerad till 20 MHz går att köra med 24 MHz under ett tag, men förr eller senare blir det alltid problem. Vi har erfarenhet av en 80C32 som blev 'helknäpp' under interrupter, medan resten av chipet fungerade perfekt.

Om du vill bygga 'light'-versionen rekommenderar vi Philips 80C32 -IBPN, -UBPN eller -IFPN versionerna, vilka är garanterat 24 MHz. Så vitt vi vet är Intelprocessorna hos denna familj begränsade till 20 MHz, men vi är inte helt säkra. Om du vill bygga 'full'-versionen skall du använda en 80C320MCG från Dallas Semiconductor (detta är den version man ser oftast).

För UART-en kan du använda Texas Instruments TL16C450 eller National Semiconductors NS16C450. Du kan också använda TL16C550, eller ekvivalent, utan problem.



Figur 2. Kretskortslayout och komponentplacering. Kortet finns att köpa.

Dessa komponenter kan hittas med mer allmänna typnummer som 16450, 16C450, 16550 eller 16C550. Du kan hitta dessa komponenter på seriekort som du kan köpa löst till din PC, ofta billigare än vad en lös komponent kostar. Använd inte 82C50 eller 8250 eftersom dessa chip normalt inte är kompatibla med 4-MHz kristaller.

Den enda problematiska komponenten (när det gäller tillgänglighet) är DC/DC-omvandlaren IC13, en NFM0505S. Vilket kompatibel, isolerad, DC/DC-omvandlare som helst kan användas, men kolla benkonfigurationen nog eftersom denna kan skilja sig mellan olika komponenter.

När det gäller de två kristallerna på kortet så ser du att lödplattorna är större än isoleringsringarna på de flesta kristaller, vilket kan leda till kortslutning. Vi rekommenderar att du monterar kristallen antingen 1 mm ovanför kortet eller med en plastbricka mellan denna och kortet.

Mikrokontrollerns kristall används med en liten **alternativ** spole, som används för att

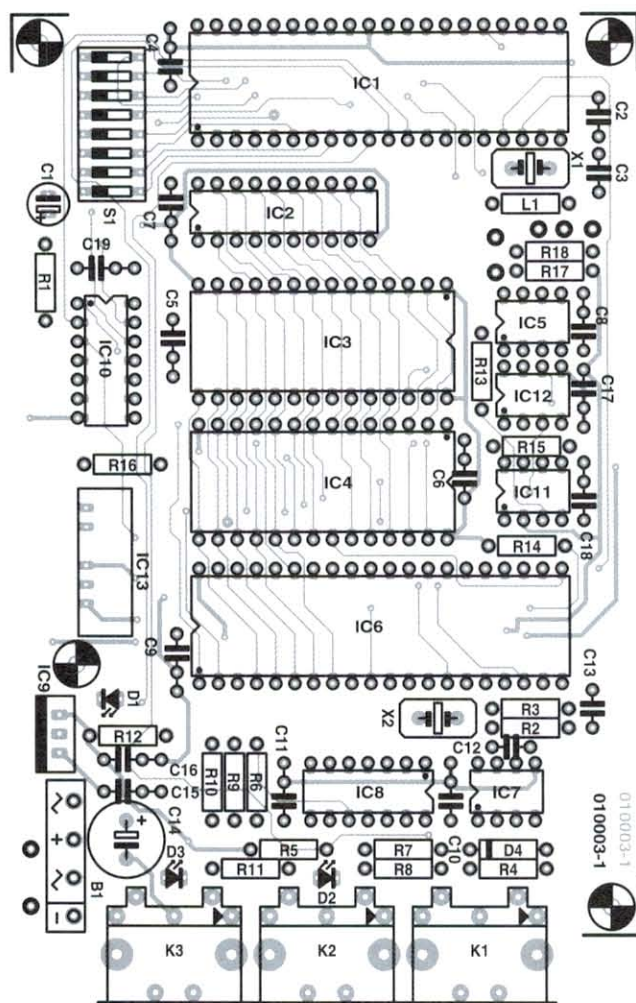
tvinga in kristallen på övertonerna. Testa alltid kortet första gången utan spolen. Om du hittar 24 MHz på processorns X2-ben (med ett oscilloskop eller en frekvensmätare) är allting OK. Om du bara hittar 12 MHz måste du ha spolen. Om kristallen svänger vid fel frekvens kommer DMX-utgången att fungera korrekt, men inte MIDI-ingången eller utgången. Om du

inte har något oscilloskop eller någon frekvensmätare kan du testa kortet i mod 2 eller 4 med mjukvaran DMX Tester på projektdisketten. Om du kan få fram (korrekta) MIDI-data kan du vara säker på att kristallen svänger vid rätt frekvens.

Kylelementet på IC9-regulatorn är i stort sett valfritt eftersom effektförbrukningen är ganska låg. En liten

Tabell 5 : Stifttilldelning på DMX-kontakter

Honkontakt	Hankontakt	Honkontakt
Stift	DMX IN	D
1	Ground (SHLD)	Ground (SH
2	Transmit DMX- (TD-)	Transmit
3	Transmit DMX+ (TD+)	Transmit
4	Not connected	Receive DMX- (
5	Not connected	Receive DMX+ (



KOMPONENTLISTA

Motstånd:

R1 = 470k Ω
R2 = 1M Ω
R3 = 1k Ω
R5, R14, R15 = 1k Ω
R4, R6-R12 = 220 Ω 0.5W
R13, R16 = 390 Ω 0.5W
R17, R18 = 4k Ω 7

Kondensatorer:

C1 = 1 μ F 10V
C2, C3 = 33pF
C4-C11, C17, C18, C19 = 47nF
C12 = 22pF
C13 = 47pF
C14 = 1000 μ F 25V radial
C15, C16 = 100nF

Spolar:

L1 = 1 μ H5

Halvledare:

D1, D2, D3 = LED
D4 = 1N4148
IC1 = DS80C320MCG (Dallas Semiconductor)
IC2 = 74HCT573 or 74F573
IC3 = 27C256 (programmerad, ordernummer **010003-21**)
IC4 = 6264 (RAM)
IC5 = LTC490 (Linear Technology)
IC6 = TL16C450 (Texas Instruments)
IC7, IC11, IC12 = 6N137
IC8 = 74LS04
IC9 = 7805
IC10 = 74LS14
IC13 = NMF05055

Övrigt:

B1 = B80C250
K1, K2, K3 = 5-pol DIN-sockel, för kretskort, stift 180 grader
PC1-PC7 = lödstift
S1 = 8-pol DIP-switch
X1 = 24MHz kristall
X2 = 4MHz kristall
Kretskort **010003-1**
Diskett med projektmjukvara **010003-11**

bit aluminium på några kvadratcentimeter räcker. Om du väljer en inbyggnadslåda av metall (som vi rekommenderar) kan du använda en av sidorna som kylelement. Glöm inte att höljet på C9 är anslutet till jord. Det är bäst att isolera det från lådan för att undvika jordslingor. Strömförsörjningen kan antingen vara en liten extern nätadapter (i så fall kan du utelämnat likriktaren B1 och kondensator C14) eller en 9,5 V, 5 VA nättransformator med lämplig säkring.

Länken till DMX-världen görs med en standard 5-polig XLR-kontakt såsom föreskrivs i DMX-specifikationen. Eftersom det finns många olika stiftkonfigurationer för kretskortsmonterade XLR-kontakter bestämde vi oss för att använda chassimonterade versioner med kopplingstrådar till och från kretskortet.

Det kan vara svårt att hitta 5-poliga

chassimonterade XLR-kontakter av hantyp. Här kan du utnyttja en otydlighet i DMX-specifikationen, då stift 4 och 5 är reserverade för 'alternativa DMX-data' kan du använda dem som DMX-ingång. Stiftkonfigurationen för DMX, i enlighet med DMX-specifikationen, visas i tabell 5. När det gäller inbyggnadslådan rekommenderar vi starkt en metalllåda som är skyddsjordad. Faktum är att kretsen är designad för professionell användning i 'svår' RFI-miljö. Det finns ingenting som är så genant som när ljussystemet 'hänger' sig under en show, därför att processorn låser sig på grund av RFI (radio frequency interference).

Den enklaste lösningen är att bygga in transformatorn i lådan och använda en vanlig nätkontakt med jorden kopplad till lådan. Om du vill bygga en verklig 'high-end' version kan du lägga in ett filter mellan nätkontak-

ten och transformatorn.

En av våra prototyper var mycket nyckfull vid uppstart och gav felaktig UART-funktion. Enda sättet att få den att fungera korrekt var att lägga till en tryckknapp parallellt med kondensator C1 så att vi kunde återställa kortet manuellt. Efter detta fungerade prototypen hela tiden. Även om detta fenomen endast uppstod i en av våra prototyper kan det vara värt att veta. Till slut upptäckte vi att BAUD-OUT-benet på IC6 hade fastnat vid 66 Hz när

det skulle gå till 4 MHz.

När kortet är färdigbestyckat är det dags för det slutliga testet.

Det kritiska ögonblicket: Test

För att kunna testa interfacet behöver du en DMX-mottagare, som du kan hitta inuti en DMX-strålkastare eller projektor. Du kan också hyra en DMX-testare som visar DMX-värden (det finns ljusfirmor som kan hjälpa dig med detta).

För MIDI är det enklast att använda en PC med ett MIDI-interface och som kör Windows 95/98. Här kommer de två specialdesignade mjukvaruverktygen till användning.

Anslut PC:ns MIDI OUT till interfacets MIDI IN kontakt och anslut DMX OUT på interfacet till DMX IN ingången på strålkastaren (eller DMX-testern). **Glöm inte DMX-terminatorn**, speciellt inte om DMX-linjen är lång. Konfigurer DMX-slaven (strålkastaren eller testern) på vilken DMX-adress som helst och välj mod 1 eller 3 på interfacet. Välj också MIDI-kanal och MIDI-meddelandetyp på det.

Spänningssätt nu interfacet. POWER LED skall vara den enda som tänds. Om inte, koppla bort interfacet och felsök.

Starta programmet DMX TESTER som finns på projektdisketten. Välj det MIDI-interface som används för att ansluta ditt interface, den valda MIDI-kanalen och arbetssättet (mod), 1 eller 3. Välj en DMX-kanal som motsvarar strålkastaren eller projektorn med hjälp av comboboxen 'Start' och dra sedan i regeln. DMX-slaven skall reagera för DMX-kommandon. Om du vill kan du välja 'Group' alternativet för att samtidigt skicka den DMX-nivå du ställt in med regeln till alla DMX-kanaler mellan 'Start' och 'Stop'.

För att testa i andra riktningen (DMX/MIDI) skall du ha en DMX-generator, som ett ljusbord för scenbelysning. I vissa fall kan du låta bli det, men testet kommer endast att ske på MIDI-sektionen (om det första testet var OK är det inte så stor risk för att DMX-sektionen inte skall fungera även i den riktningen). Ställ in mod 2 (eller 4) på interfacet och välj en MIDI-kanal med tillhörande MIDI-meddelande att sända. Anslut MIDI OUT kontakten på interfacet till MIDI IN ingången på PC:n. Anslut DMX-generatorn till DMX IN på interfacet, om möjligt med en DMX-terminator.

Starta programmet DMX VIEWER (finns på disketten) och ge det sedan MIDI-kanalen och MIDI-meddelandet som valts på kortet. Glöm inte att välja ett MIDI-interface om det är nödvändigt. I det ögonblick som interfacet får ström skall LED-en MIDI OUT tändas. Om du har en DMX-generator kan du ändra DMX-värdena och verifiera detta på skärmen. Om du inte har det skall detta test köras med

Delta-mod deaktiverat (S1-6 = OFF) annars får frånvaron av värdeändringar på DMX MIDI-länken att 'stallas'.

Om du under testet noterar konstigheter med de DMX-värden som sänds eller tas emot (med VIEWER kan du se att vissa DMX-värden fastnat i felaktiga värden) skall du överväga att ersätta 74F573 med en 74HCT573. Skälet är att vissa RAM-IC inte går så bra ihop med 74F-IC. I dessa sällsynta fall löser bytet till HCT problemet.

Hur används interfacet?

Detta interface är lämpligt för många applikationer, inklusive (naturligtvis) ljusshower, men också för service, underhåll och belysningstester. I det senare fallet är den enklaste lösningen att använda DMX TESTER och DMX VIEWER på disketten (ordernummer 010003-11).

För ljusshower är en MIDI-sequencer det grundverktyg som används ihop med interfacet. Detta interface är i princip kompatibelt med alla sequencers för PC, Mac eller som är fristående. Den enda skillnaden mot den etablerade användningen av en sequencer är att note velocity (eller kontrollvärden) inte styr syntljöd utan ljusstyrka, spegelpositioner, gobotyper och hastighet etc. beroende på typ av DMX-enhet.

För att styra någonting på DMX spelar du bara in toner (eller kontrollvärden) med velocitet (eller värden) som motsvarar de önskade DMX-nivåerna, sedan spelar du upp sekvensen och ser hur DMX-enheterna uppträder. Under de första experimenten rekommenderar vi att du inte använder Note Off eller långa tonomlopp. Detta för att se till att

DMX-kontrollen har tillräckligt med tid att sända ett meddelande. Så länge som du 'känner' effekterna av MIDI till DMX kan du anpassa programmeringen av sequencern efter dina egna behov.

De möjligheter som ges av en sequencer är i stort sett obegränsade, även om det måste sägas att du måste vara helt hemmastadd med din sequencermjukvara eller maskin för att snabbt uppnå goda resultat. Våra 'fält' experiment visade tydligt att programmering och användning av en sequencer var ett alldeles för tungt jobb för många små och medelstora klubbar där DJ:n typiskt arbetar som LJ också (light jockey). För dessa fall har konstruktören utvecklat två program, SoftController I och SoftController II, vilka ger en enkel och halvautomatisk styrning av ett DMX-system.

Att beskriva dessa program här skulle ta oss alltför långt från syftet med denna artikel. Du skall dock veta att dessa presenterar ett ljusbord för scenbelysning vars utseende kan sägas vara standard inom detta område. I stort sett alla menyer är borttagna. För att funktionen skall vara så intuitiv som möjligt behöver du bara klicka på knapparna för att aktivera dem. På samma sätt styr du reglarna med musen osv.

Denna mjukvara används faktiskt i dag i Belgien för att styra professionella belysningssystem.

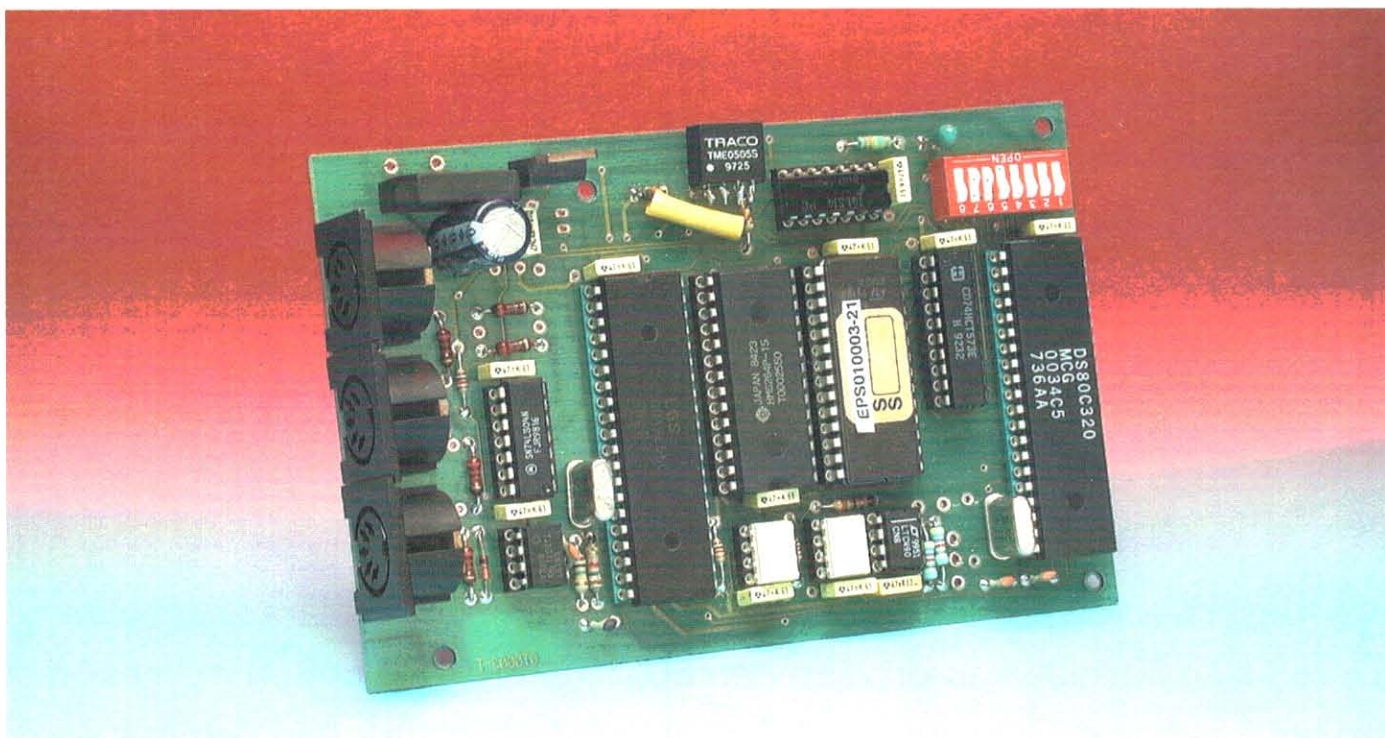
Slutligen, ett litet råd. Läs användarmanualen (skriv ut den först) innan du börjar använda programmen eftersom det finns en hel del möjligheter.

Mjukvaran på disketten har tagit många timmar av hårt arbete att utveckla. Det finns inget kopierings-skydd, inte heller någon utvärder-

På projektdisketten 010003-11

copyright.txt
dmx_tester.exe
midi_dll.dll
midi_dmx.asm
Midi_dmx.hex
model.asm
mode2.asm
mode3.asm
mode4.asm
viewer.exe

författarens copyrightmeddelande
DMX interfacetestprogram för Windows
.dll fil
EPROM-innehåll (assemblerfil)
EPROM-innehåll (hexadecimal fil)
assemblerkodfil för Mod 1
assemblerkodfil för Mod 2
assemblerkodfil för Mod 3
assemblerkodfil för Mod 4
DMX interfacevisningsprogram för Windows

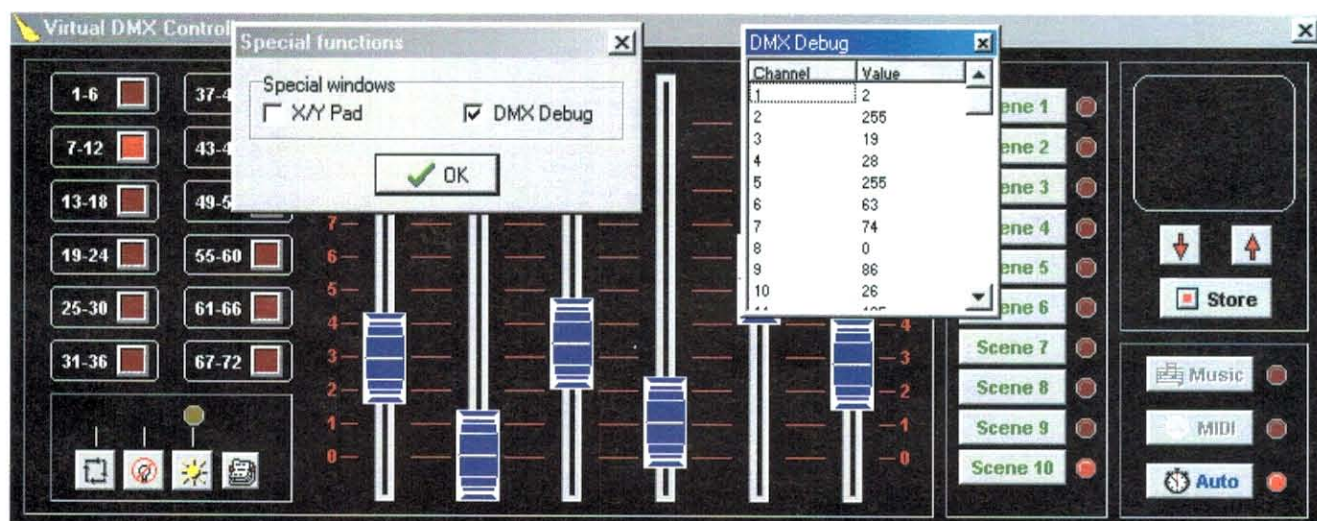


Figur 3. En tidig prototyp av interfacekortet. Vi var tvungna att improvisera en smula runt 0505S på grund av ett litet designfel som rättades till senare.

ingsversion där du måste betala efteråt, eftersom detta skulle störa ärliga användare. Trots detta är den mjukvara som är relaterad till MIDI-till-DMX interfacet inte 'kopieringsfri'. När du köper projektdisketten erhåller du en användarlicens för all mjukvara. Var vänlig och ge inte bort, eller använd, denna kopior av

denna mjukvara utan köp så många disketter som du har interface. Med tanke på detta finns inte denna mjukvara att hämta gratis på vår Downloadsida.

(010003-1)

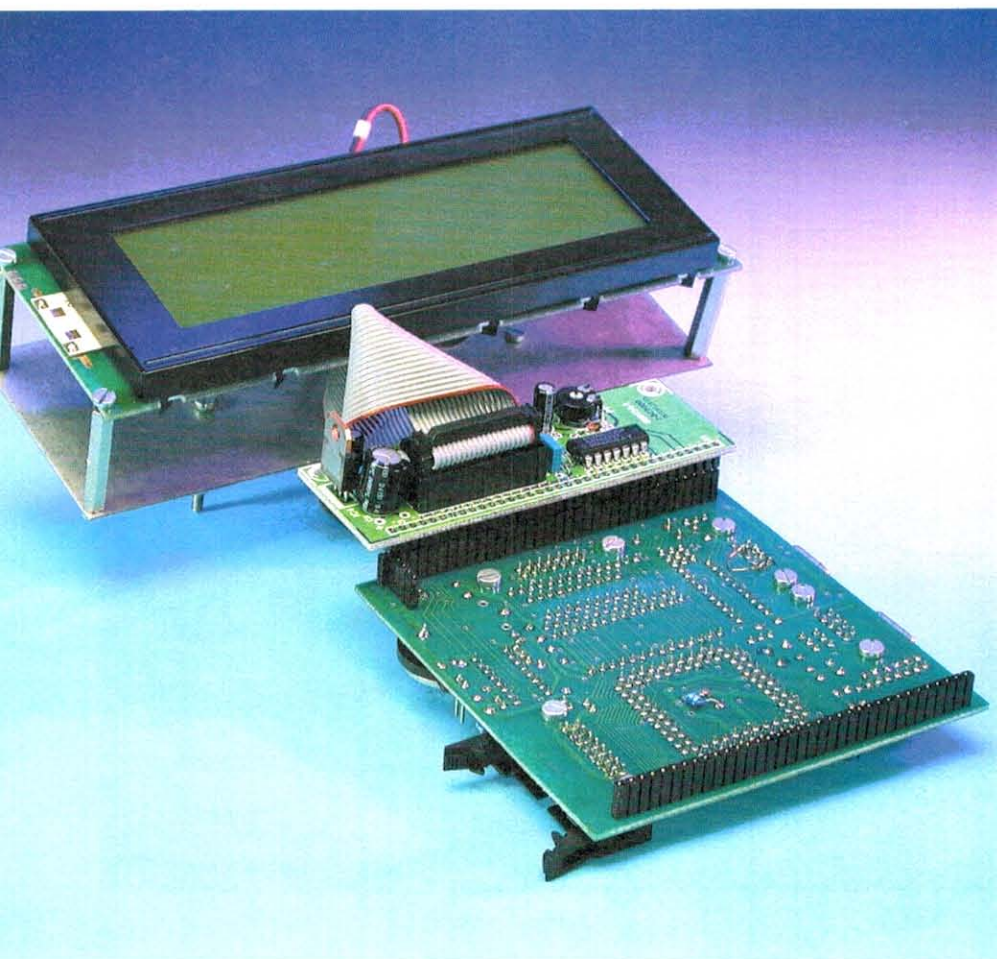


Grafisk LCD-modul för 8051-kontroller

Ilek med pixlar och byte

Design av Luc Lemmens

Här beskriver vi hur en display som använder den populära kontrollern T6963 från Toshiba kan drivas av en 8051 (eller kompatibel) mikrokontroller. Läs hur du kan använda displayen för en grafisk bild och hur du omvandlar en bild i din PC till de pixlar som displayen behöver.



För de välkända alfanumeriska displayer som använts i många av projekten i *Allt om Elektronik* har Hitachi satt en standard med sin kontroller HD44780. Nästan alla tillverkare använder denna kontroller i sina LCD-moduler och i praktiken spelar det därför ingen roll vilken display du använder för ditt projekt, även om kontakten kan skilja sig från ett märke till ett annat.

Läget är i stort sett likadant när det gäller monokroma grafiska moduler, men här är standardiseringen av en kontroller betydligt sämre. En kontroller som används ofta är Toshiba's T6963, så vi bestämde oss för att se hur en display försedd med denna kontroller kunde drivas med hjälp av en 8051, eller kompatibel mikrokontroller. Vår utgångspunkt var det kompakta 80C597-kort som vi beskrev i nummer 1 och 2/2000, tillsammans med dess EEPROM (976510-1), men i princip kan vilket mikrokontrollersystem som helst användas.

Den grafiska display som användes som försökskanin för våra experiment var en Sharp LM24014 modul som har 240x64 pixlar (bildpunkter), men vilken annan display som helst med en T6963-kontroller kan använ-

das för experimenten i denna artikel utan problem. För displayer med ett annat pixelförhållande behövs det lite modifiering i den mjukvara som beskrivs i denna artikel.

Naturligtvis skulle vi kunna säga mycket mer om drivningen av denna displaytyp än vad som får plats i denna artikel. För andra applikationer och alternativ, som att använda standard textmod (jämförbar med en alfanumerisk display), hänvisar vi till databladet för T6963, eller dokumentationen för den displaymodul du har ansluten till denna kontroller.

The LCD-modulen

Tre register i den grafiska modulen används för att driva LM24014, dataregistret, kommandoregistret och statusregistret. Kommunikationen sker genom att först skriva ett, två eller inga värden till dataregistret och sedan skicka en instruktion till kommandoregistret. Efter detta kan ett värde läsas tillbaka från dataregistret, om detta stöds av instruktionen i fråga. Statusregistret måste pollas (kontrolleras) före varje individuell läs eller skrivuppgift, för att se efter om displayen är färdig med föregående uppgift och klar att ta emot och behandla nya data.

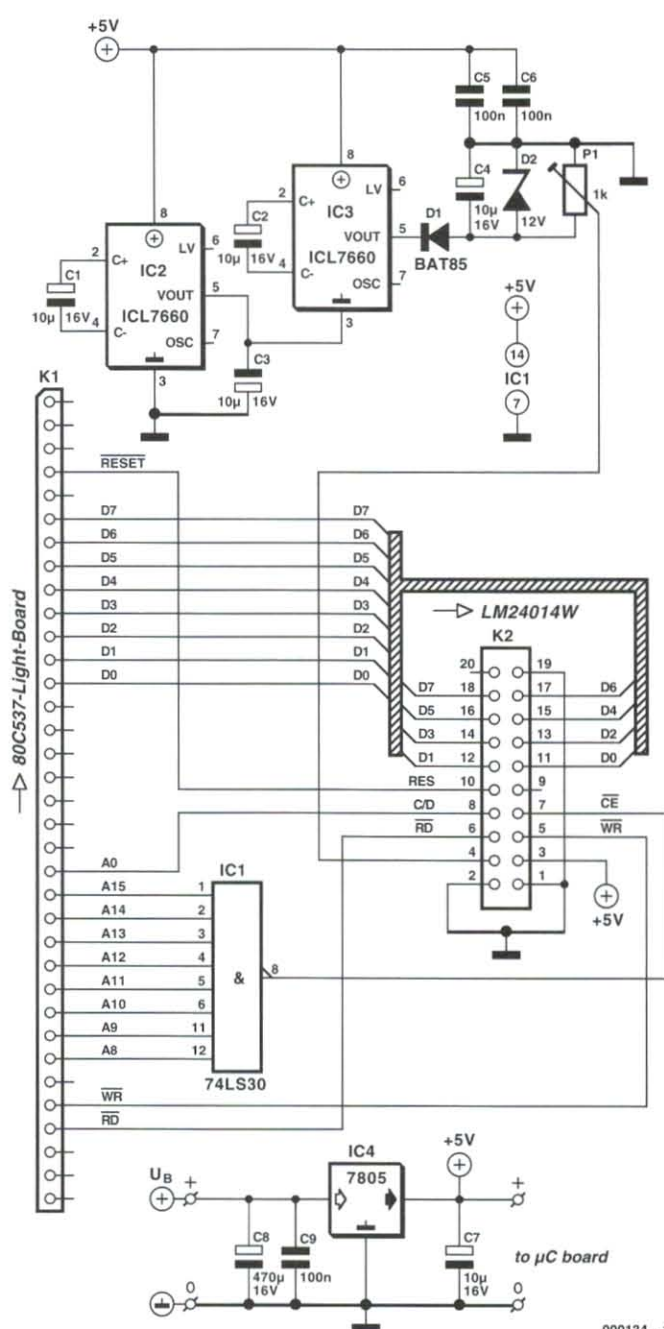
Kommando och dataregistren kan kommas åt direkt, men inte statusregistret. Om vi vill läsa statusen måste vi först skicka instruktionen Read Status till kommandoregistret och därefter kan ett statusbyte läsas från dataregistret. Som man kan se av figur 1 finns det olika statusbiter som visar vilka processer displayen kan hantera. I vissa fall är det nödvändigt att kolla flera statusbiter och verifiera att de har korrekta värden innan man instruerar displayen att behandla ny information.

Interface

De 20 anslutningsbenen på displayen inkluderar 8 dubbelriktade datalinjer, en resetlinje och 4 kontrollsignaler (CE, RD, WR och C/D), vilka vi går igenom längre fram). LCD-modulen har också ett internt tecken-ROM, på samma sätt som de mer välkända alfanumeriska displayerna och via ben 19 kan du välja mellan en teckenmatris på antingen 6x8 eller 8x8 pixlar. Displayen arbetar med 5 V, men för kontrastjusteringen behöver den en relativt stor

Status Bit	Description		Function
STA0 (BUSY1)	Busy flag to indicate whether T6963C is ready to accept instruction	"0" "1"	=NOT READY =READY
STA1 (BUSY2)	Busy flag to indicate whether T693C is ready to accept data read or write	"0" "1"	=NOT READY =READY
STA2 (DARRDY)	Data Auto Read Ready flag (Only valid in Data Auto Read/Write modes (section 8.6))	"0" "1"	=NOT READY=READY
STA3(DAWRDY)	Data Auto Read Ready flag (Only valid in Data Auto Read/Write modes (section 8.6))	"0" "1"	=NOT READY =READY
STA4	-	-	-
STA5 (CLR)	Clear flag indicating operation of the T6963C	"0" "1"	=NOT CLEARED =CLEARED & Operating
STA6 (ERROR)	Error flag for Screen Peeking and Screen Copy commands (section 8.8 & 8.9)	"0" "1"	=Address Pointer Valid = Address Pointer out of Graphic Area
STA7 (BLINK)	Blink flag to indicate status of Blink condition	"0" "1"	=Display OFF =Normal (Display ON)

Figur 1. Innehållet i statusregistret.



000134 - 11

Figur 2. Adapterkrets för anslutning av displayen till 80C537-kortet.

supplementär negativ spänning (mellan -12 V och -6 V enligt databladet). Många moderna displayer kan klara sig med ett diodspänningsfall under jordnivån, eller till och med 0 V. Kolla med databladet om du använder en annan typ av displaymodul.

Displayen ansluts till 80C537-kortet via den enkla adapterkrets som visas i figur 2. Displayens databuss ansluts direkt till databussen på mikrokontrollerkortet. IC1 är en adressavkodare som gör att displayen kan adresseras vid 0FFxxh. Den avkodarutgång som motsvarar denna adress ansluts till CE-linjen (Chip Enable) på LCD-modulen. Adresslinje A0 ansluts till C/D, som är den kontrollinje som tar hand om valet av det interna kommandot eller dataregistret hos displayen. Detta betyder att dataregistret kan läsas från eller skrivas till vid adress 0FF00h, medan kommandoregistret är placerat vid 0FF01h. Reset, RD och WR (Read och Write) linjerna på displayen är anslutna till signallinjerna med samma namn på mikrokontrollerkortet. Interfacekortet är också försett med en 7805-spänningsregulator som ger arbetsspänning för displayen. Denna spänning kan också användas för att mata processorkortet via stiften 5 och 9 (GND respektive +5 V) på K1 på kretskortet. Kretskortslayouten för interfacekretsen visas i figur 3. Då detta kort finns att köpa färdigt från oss borde det inte vara några problem med att bygga kretsen.

Att driva displayen med mjukvara

Flödesschemat i figur 4 visar den följd i vilken data måste nå displayen. Det första du måste göra är naturligtvis att slå på strömmen och låta displayen erhålla en power-on reset. Som man kan se ges denna signal i vårt fall av Reset-signalen från mikrokontrollerkortet.

Innan displaymodulen kan ta emot några bilder är det nödvändigt att ställa in några parametrar. Detta involverar (i godtycklig ordning) att definiera bredden på den grafiska ytan på displayen, specificera startadressen för den grafiska ytan, välja displaymod och aktivera det grafiska displaymodet via mjukvaran. I motsats till en alfanumerisk display finns det inte någon 'äkt' initialisering. De parameterinställningar vi gör här kan självklart ändras senare.

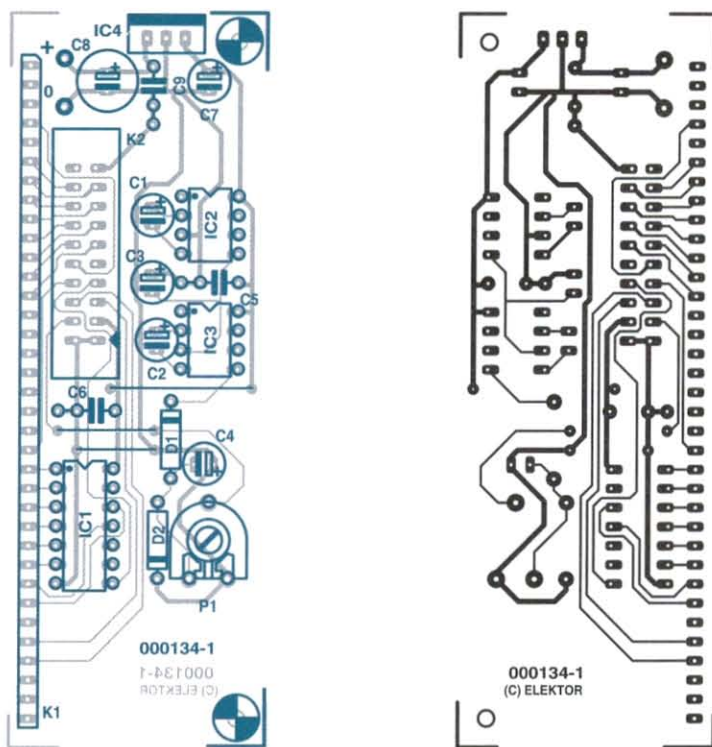
Grafisk yta

Bredden på det grafiska området (GA) beror på nivån hos ben 19 på displaymodulen, vilket väljer en teckenmatris på 6x8 eller 8x8 pixlar. Som man kan se fylls skärmen från övre vänstra hörnet till nedre högra hörnet med rader som är 1 pixel höga och (i vårt fall) åtta pixlar breda. Om vi väljer 6x8 (ben 19 vid +5 V) är de databyte vi skickar till displayen egentligen bara sex pixlar eftersom de två

mest signifikanta bitarna helt enkelt ignoreras. Med 8x8 (ben 19 vid 0 V) visas de två övre bitarna på displayen. Vår display är 240 pixlar bred och vid 8x8 är bredden på den grafiska ytan $240/8 = 30$ (hexadecimal 0E1h) och vid 6x8 är den $240/6 = 40$ (hexadecimalt 028h).

Grafisk hemadress

Minnet i en displaymodul är i allmänhet större än vad som kan visas på displayen. Följaktligen måste vi specificera vilken minnesadress som skall visas i skärmens övre vänstra hörn. Detta kallas för Graphic Home Address (GHA). Innehållet i denna minnesplats innehåller de första åtta pixlarna (små horisontella band) på displayen. Nästa adress innehåller de åtta pixlar som är placerade omedelbart till höger om de åtta första pixlarna. I vårt fall börjar den andra raden på displayen 30 adresser efter startadressen och varje följande n-te rad startar vid adress $GHA + n$ (GA). När en bild har placerats i det grafiska minnet kan vi få den att rulla (scrolla) vertikalt över displayen genom att öka eller minska värdet på GHA med hjälp av heltalsmultipler av GA.



Figur 3 . Kretskortslayout och komponentplaceringsritning för kretsen i figur 2.

Displaymod

Här kan vi, förutom sådant som att switcha markören (cursorn) till och från, specificera vad displayen skall göra om textpixlar och bildpixlar överlappar varandra. I OR-mod kommer en pixel att switchas till om den drivs antingen som en text eller en bildpixel. De övriga alternativen är AND (pixeln är till om den drivs i båda formerna) och XOR (pixeln är till om den endast drivs i en av de två formerna).

Data för den grafiska displayen

När dessa inställningar har gjorts kan data skrivas till det grafiska minnet i LCD-modulen. Med GHA-inställningen har vi redan specificerat vilken minnesadress som innehåller de data som visas överst till vänster på displayen (det första horisontella bandet med 8 pixlar). Vi kan adressera varje byte på displayen genom att sätta Address Pointer (adresspekaren) hos displayen till det värde som motsvarar den tillhörande adressen och sedan skicka ett databyte till modulen. Det är dock ganska klumpigt att alltid vara tvungen att

KOMPONENTLISTA

Motstånd:

PI = 1kΩ trimpot

Kondensatorer:

C1-C4, C7 = 10μF 16V

C5, C6, C9 = 100nF

C8 = 470μF 16V

Halvledare:

D1 = BAT85

IC1 = 74LS30

IC2, IC3 = ICL7660

IC4 = 7805

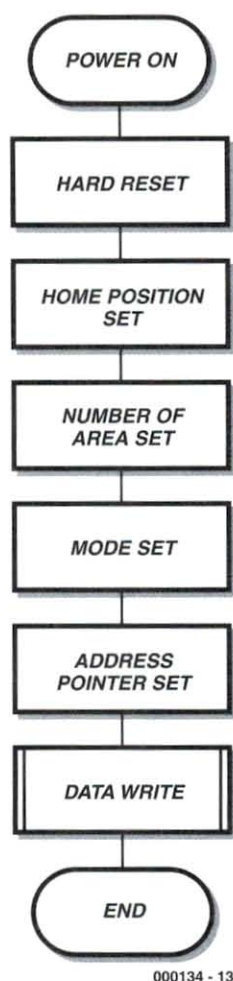
Övrigt:

K1 = 35-pol SIL-kontakt

K2 = 20-pol boxheader

Kretskort **000134-I** (se
Läsarservice)

Diskett med projektmjukvara
000134-II (se Läsarservice)



Figur 4. Detta flödesschema visar i vilken ordning data skall komma fram till displayen.

först skicka två byte för adressen och sedan skicka data till modulen och det är därför som modulen också har alternativet Auto Write. Här behöver startadressen enbart anges en enda gång, följd av alla databytten. Displayen tar då själv hand om uppräknings (inkrementeringen) eller nedräknings (dekrementeringen) av Address Pointer efter det att varje individuellt byte har behandlats. Efter det sista databytet måste detta arbetssätt (mod) switchas från.

Notera att det också är möjligt att ändra en enda pixel. Detta beror på det faktum att det grafiska minnet också kan läsas. Vi kan således läsa det byte som innehåller pixeln ifråga och switcha en bit till eller från med hjälp av AND eller OR-instruktioner och sedan skriva det modifierade bytet tillbaka till samma adress i displaymodulen (naturligtvis är det också möjligt att ändra flera bitar samtidigt).

Att kommunicera med displayen är inte alltför krångligt, men hur översätter du en bild till byte. På den gamla goda tiden gjorde man detta med hjälp av rutat papper, genom att först färglägga de individuella rutorna för att forma bilden i stort och sedan upprepa räkna av block i form av åtta rutor i fyrkant (där en svart ruta betydde '1' och en vit ruta '0'). Och det är denna teknik vi kommer att använda här, men med hjälp av moderna verktyg!

Från bild till byte

Vi kommer att använda PC:n för att preparera den bild som skall återges på den grafiska displayen. Bilden kan ritas i Paint (det standardritprogram som medföljer Windows), eller också kan en befintlig bild förstöras eller förminsas så att den passar in på displayen och därefter lagras i bitmappformat (.BMP). Därefter använder vi en applikation som skrivits i Delphi (version 2.0) för att översätta pixlarna i bilden till databyte för displayen. Dessa byte adderas automatiskt till en assemblerspråkslista, som i sin tur måste översättas till en Intel HEX fil med hjälp av en 8051-assembler. Ett terminalemuleringsprogram används därefter för att ladda in denna fil i programminnet på 80C537-kortet via serieporten och applikationen startas. Efter detta kan vi beundra resultatet av våra

ansträngningar och artistiska inspiration på den grafiska displayen.

Att göra bilder med Paint

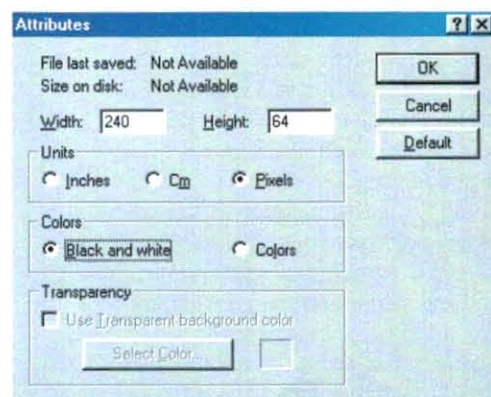
För att kunna göra en bild (eller anpassa den) för återgivning på displayen skall du först starta Paint (Program/Tillbehör under Start-menyn). Klicka sedan på menyn Bild/Inställningar och ange följande inställningar (se figur 5):

Enhet: bildpunkter
Färger: svartvit
Bredd: bredden på displayen i pixlar (i vårt fall, 240)
Höjd: höjden på displayen i pixlar (i vårt fall, 64)

Efter detta kan du börja arbeta med ritverktygen och tillverka en ny bild, eller så kan du ta en befintlig bild i vilket grafiskt format som helst som stöds av Paint (BMP, GIF etc.) och förstora eller förminska den så att den passar snyggt och prydligt in i fönstret. När du stänger filen måste de se till att den verkligen sparas som en monokrom bitmap!

Omvandla bitmappen

Förutom själva bilden innehåller BMP-filer en hel del övrig information (som antalet färger, dimensioner osv.) som vi inte behöver för vår applikation. Vi går inte in mer på denna 'extra' information, utan använder Delphi för att dra ut rena pixeldata från bitmappfilen. I Delphi kan vi dra nytta av en 'canvas' komponent, som är en slags bakgrund på vilken bilden kan placeras. Programmet tar hand om konverteringen av BMP-filen till en grafisk presentation och 'canvasen' blir då en tvådimensionell pixelrad, den moderna versionen av vårt gamla rutpapper. Canvaskomponenten anpassar sig automatiskt till bildens storlek och programmet behöver inte modifieras för att användas med andra storleksförhåll-



Figur 5. Dessa inställningar skall göras när du startar ritprogrammet Paint.

landen eller en annan displaymodell. Med hjälp av en dubbel FOR-loop tittar vi på varje pixel i tur och ordning för att se om den är svart eller vit. Vi lägger alltid in åtta pixlar åt gången i varje byte, eftersom det är denna representation som vi behöver senare för vår grafiska modul.

Delphiprogrammet öppnar direkt assemblerfilen TEST.A51 och klistar data för displayen i denna fil med hjälp av DB (Define Byte) instruktioner. Bilden lagras i målsystemets minne som en tabell, vilket gör att den snabbt kan återges på displayen.

Att återge bilden

Efter föregående steg är allt som återstår det assemblerprogram som tar hand om återgivningen av bilden på displayen. Detta program är ganska enkelt. Först tilldelas logiska namn till adresserna i de tre registren i 8051-an samt data och kommandoregistren i displayen. Därefter sätts displayparametrarna (GHA, GA, Mode Set, Address Pointer Set) och displayen läggs i Auto Data Write mod så att adresspekaren automatiskt inkrementeras efter det att varje byte tagits emot. Vi ser sedan att samma struktur stadigt upprepas, först (om så är nödvändigt) skickas data till dataregistret hos displayen och sedan skickas en instruktion till kommandoregistret så att LCD-modulen vet vad den skall göra med datan. Notera speciellt statuskontrollen **före** varje skrivinstruktion. Därefter följer som det som vi alla väntat på, skrivning av data in i displayens grafiska minne. Det datablock som börjar vid etiketten 'block' adderades till listningen av Delphi-programmet. Etiketten 'enddata' markerar adressen för det första byte som följer efter detta block. Med denna teknik gör vi assemblerprogrammet oberoende av displayens storlek. Du skall alltid komma ihåg att bilden mycket väl kan vara större än de fysiska dimensionerna för displayen, eftersom det grafiska minnet är större än displayens yta (en kritisk kommentar: Storleken på grafikminnet anges inte någonstans i databladet!). Det är fullt möjligt att skriva några extra rader i minnet och sedan få bilden att rulla över displayen (scrolla) genom att inkremen-

Listning av assemblerkod

```
ORG 0100H
; Definition of special function registers in the 8051
; data pointer DPTR, 16 bit register represented by DPL and DPH
DPL EQU 082H
DPH EQU 083H
; accumulator
ACC EQU 0E0H

; two registers of the graphic module: data and command register
data EQU 0FF00H
command EQU data+1

ACALL chk01 ; check if the display is ready to receive data

; the Graphic Home Address will be set. First we'll send the LSB of
; this address (0H)
MOV DPTR,#data ; DPTR points to LCD data register
MOV A,#0H
MOVX @DPTR,A

ACALL chk01 ; check if the display is ready to receive data
; then the MSB (0H)
MOV DPTR,#data
MOV A,#0H
MOVX @DPTR,A
ACALL chk01 ; check if the display is ready to receive command
; and instruction Graphic Home Address Set (42H)
MOV DPTR,#command
MOV A,#42H
MOVX @DPTR,A
ACALL chk01 ; check if the display is ready to receive data

; The following instructions will set the Graphic Area (GA). The display is
; 240 pixels wide, 240/8 = 30 = 001EH
MOV DPTR,#data
MOV A,#1EH ; LSB of GA
MOVX @DPTR,A
ACALL chk01

MOV DPTR,#data
MOV A,#0 ; MSB of GA
MOVX @DPTR,A
ACALL chk01

MOV DPTR,#command
MOV A,#43H ; Graphic Area Set = instruction 43H
MOVX @DPTR,A
ACALL chk01

MOV DPTR,#command
MOV A,#80H ; Mode Set: logical OR
MOVX @DPTR,A
ACALL chk01

MOV DPTR,#command
MOV A,#98H ; Display Mode Set: text off, graphics on
MOVX @DPTR,A
ACALL chk01

; The following instructions will set the LCD graphic Address Pointer to
; address 00H.
MOV DPTR,#data
MOV A,#00H ; LSB of the Address Pointer
MOVX @DPTR,A
ACALL chk01

MOV DPTR,#data
MOV A,#0 ; MSB of the Address Pointer
MOVX @DPTR,A
ACALL chk01

MOV DPTR,#command
MOV A,#24H ; Address Pointer Set = instruction 24H
```

[illegible]

tera Graphic Home Address med multipler av Graphics Area (GA). Om du använder en display med annan storlek måste du ändra värdet på GA i assemblerlistningen i enlighet med detta.

Notera: Det finns också displayer där adresserna inte processas linjärt. Det finns, till exempel, en Toshi-badis-play i vilken det finns ett hopp i minnesadressen halvvägs genom displayen! För en sådan display måste assemblerprogrammet också modifieras i enlighet med detta.

Och slutligen...

Denna källkodsfil måste nu assemblas med en valfri 8051-assembler. En möjlig kandidat är assemblern för den 80C535 programmeringskurs som vi publicerade i nummer 1-4/1994 och som finns att köpa från oss, ordernummer 1811. Efter det att denna fil har översatts till en standard Intel HEX fil kan den skickas till 80C537-kortet via serieporten med hjälp av ett terminalprogram (som HypterTerminal i Windows). Inställningar för serieporten skall vara 9600 baud, 8 bitar, 1 stoppbit och ingen flödeskontroll. Slutligen kan vi beundra den bild på displayen som kostat oss så mycket blod, svett och tårar att producera.

(000134-1)

BHIAB Electronics AB

Katalog på internet

**Nu finns även BHLAB's
surplus-katalog på
internet**

www.bhiab.se
info@bhiab.se

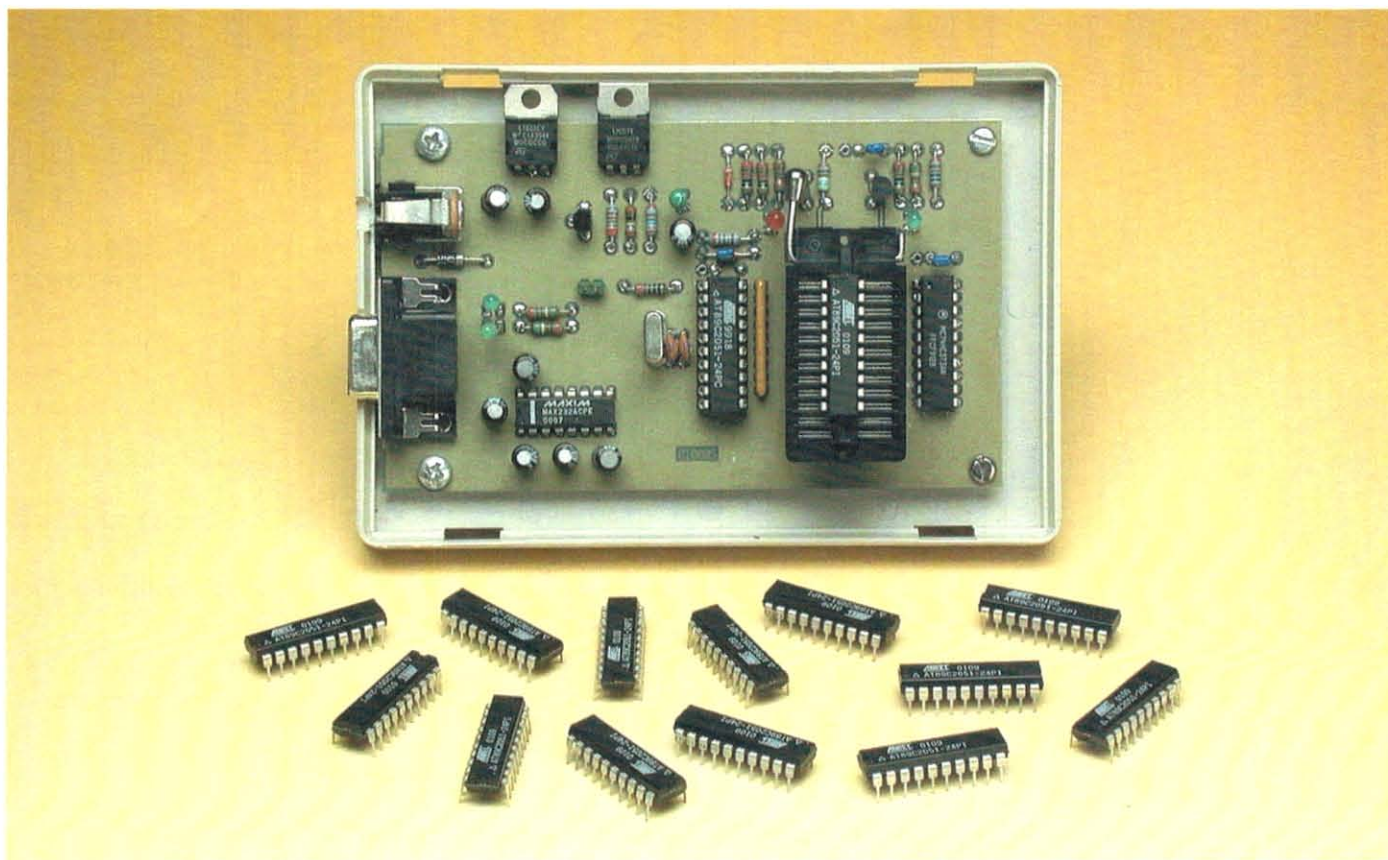
Butik
Brunnsgatan 20
611 32 NYKÖPING
Tel: 0155- 21 32 10

Programmerare för Atmel AVR Mikro

för 89Cx051 med upp till 4k programminne

Design av A. Oyrer

Denna programmerare är lämplig för programmering av de tre populära 20-bens Atmelprocessorerna 89C1051, 89C2051 och den nya 89C4051 med 4k programminne.



Det har blivit tillökning i Atmelfamiljen med kraftfulla, men ändå billiga, MCS51-kompatibla mikrokontrollers med flashminne (PE-ROM), 80C4051 har 4 kByte programminne. Förutom minneskapaciteten har alla famil-

jemedlemmar följande egenskaper:

- 128 byte RAM
- 15 I/O portbitar
- 2 16-bitars timer/räknare
- Interruptarkitektur med två priori-

teringar och fem vektorer

- Programmerbar full-duplex serieport
- Precisions analog komparator
- Integrerad klock/oscillatorkrets

AT89Cx51 familjen arbetar med en klockfrekvens mellan 0 (statisk) och 24 MHz och har två energisparmod som kan väljas med mjukvara: *Idle* (endast CPU:n är stoppad) och *Power down* (RAM-innehållet behålls men alla funktioner är bortkopplade).

Programmering av flash

Programmeringsegenskaperna hos flashminnet är naturligtvis viktiga när det gäller designen av en programmerare. När en mikrokontroller från Atmel lämnar fabriken är programminnet fyllt med FF och färdigt att skrivas till, byte för byte. Så fort minnet programmerats kan ett skrivet byte överskrivas först efter det att hela minnet har raderats. AT89Cx051 inkluderar en PEROM adressräknare, som återställs till 00H på den positiva flanken på RST och inkrementeras på varje klockpuls på XTAL1.

De logiska värdena på portstiften P3.1 till P3.7 bestämmer programmeringsmod, såsom visas i tabell 1. För att skriva in ett program i PEROM rekommenderar Atmel följande programmeringsalgoritm.

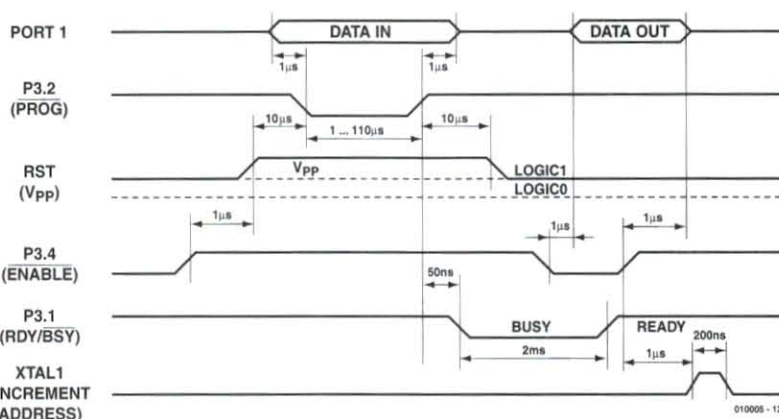
1. Starta genom att lägga matningsspänning till VCC och GND samt håll RST/VPP och XTAL1 vid jord.
2. Ta RST/VPP och P3.2 höga.
3. Sätt portbenen P3.3 till P3.7 enligt tabellen.

För att skriva och läsa programmeringsdata är nästa steg som följer:

4. Lägg det första byte som skall programmeras (in i minnesadress 00H) till bussen P1.0-P1.7.
5. Ta RST till 12 V vilket blir programmeringsprocessen..
6. Lägg en negativt gående puls, med ett pulsomlopp mellan 1 och 100 μ s, till P3.2 för att kunna skriva bytet in i PEROM eller in i låsbitarna. Skrivcykeln avslutas automatiskt efter ungefär 1,2 ms.
7. För att läsa data tas spänningen på RST ner till en normal logiswkt hög nivå och portbenen P3.3 sätts till de lämpliga nivåerna. Det byte

Programmeringsmod för flashminne

Mod	RST/VPP	P3.2/PROG	P3.3	P3.4	P3.5	P3.7
Skriv programdata	12 V	1.2 ms negativt gående puls	L	H	H	H
Läs programdata	H	H	L	L	H	H
Skriv låsbit 1	12 V	1.2 ms negativt gående puls	H	H	H	H
Skriv låsbit 2	12 V	1.2 ms negativt gående puls	H	H	L	L
Radera minne	12 V	10 ms negativt gående puls	H	L	L	L
Läs signaturbitar	H	H	L	L	L	L



Figur 1. Timingdiagram för en skriv/läscykel.

som lagrats på den aktuella adressen kommer fram på port 1.

8. När ett byte har skrivits eller lästs räknas adressräknaren upp (inkrementeras) av en stigande flank på XTAL1. Det nya bytet läggs på P1.
9. Stegen 5 till 8 repeteras tills hela minnet programmerats eller tills slutet av programmeringsdata har nåtts.
10. För att avsluta programmeringsprocessen sätts XTAL1 och RST låga och matningsspänningen tas bort från VCC och GND.

En komplett skriv/läscykel illustreras i figur 1. Portben P3.1, som signalerar **READY/BUSY** visas inte. Databladet säger att omloppet för programmeringspuls ligger mellan 1 och 100 μ s. Eftersom en programmeringscykel varar ungefär 1,2 ms, och inte får lov att påbörja en ny period innan den föregående har avslutats, tas benet P3.1 lågt (**BUSY**) under max 2 ms, under tiden skrivning sker.

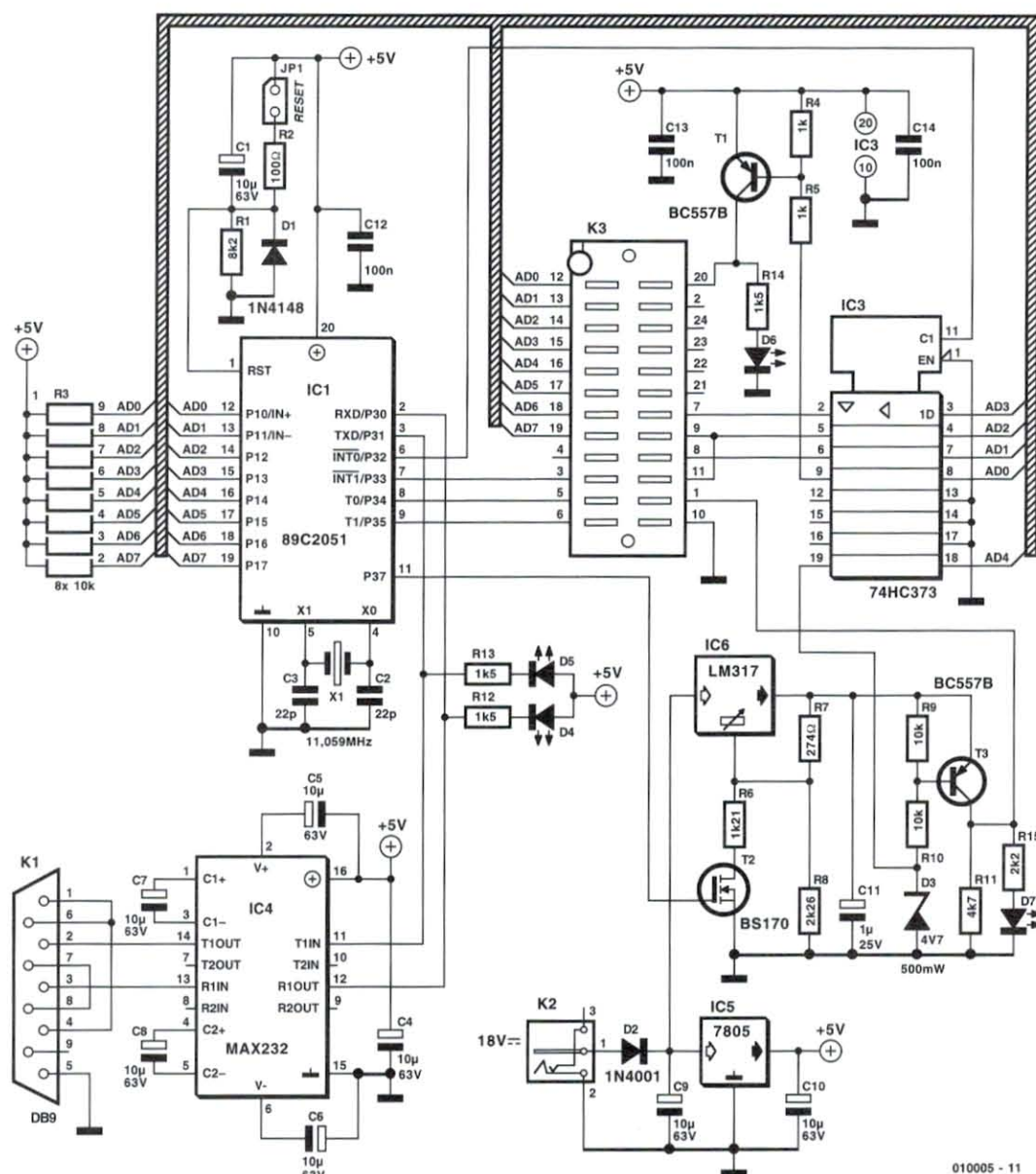
Endast när P3.1 går tillbaka till hög (**READY**) kan en ny skrivcykel initieras.

Som man kan se av timingdiagrammet kan verifiering av skrivprocessen utföras antingen byte för byte under **BUSY**-perioden under skriv (jämför) eller allt på en gång så länge som låsbitarna inte har satts (läs). Det senare utförs på följande sätt:

1. Ta RST från låg till hög för att resetta den interna adressräknaren till 000H.
2. Sätt upp lämpliga kontrollsignaler på benen P3.3 till P3.7 och databyten kommer fram på P1.
3. Lägg en puls till XTAL1 för att inkrementera adressräknaren med en plats, och nästa data kommer fram på P1.
4. Upprepa steg 3 tills hela minnet har lästs.

Låsbitarna kan inte verifieras direkt men de kan identifieras av den effekt de producerar.

Hela minnet, tillsammans med låsbitarna, raderas elektriskt när den korrekta kombinationen läggs till P3.3-P3.7 och P3.2 hålls låg under minst 10 ms. Alla minnesplatser fylls då med ettor. Denna **chip erase** proce-



Figur 2. Kretsschemat för Atmelprogrammeraren med masterkontroller och ZIF-hållare för den komponent som skall programmeras.

dur behövs innan man kan omprogrammera ett redan programmerat minne eller för att kunna radera låsbitarna. Signaturbytet ger information om tillverkare och komponenttypen. De kan läsas på samma sätt som programminnesplatserna, men med P3.5 och P7.7 låga. Möjliga värden är:

000H 1EH: tillverkarkod för Atmel
001H 11H: 89C1051
21H: 89C2051
41H: 89C4051

Programmerarhårdvara

Då, bortsett från storleken på programminnet, alla funktioner och egenskaper hos de tre

mikrokontrollerna är identiska, så är programmeraren inte komplicerad. Den måste bara läsa signaturbytet i början av varje operation för att identifiera typen av mikrokontroller. Den programmerare som visas här implementerar alla de beskrivna programmeringsmoden:

- Skriv
- Läs
- Jämför
- Programmera låsbitar
- Radera
- Läs signatur

I skrivmod checkas signaturbyten för

att kolla att den datamängd som skall programmeras får plats i processorn. I läsmod kan den datafil som genereras vara i Intel Hex eller i binärt format. Läsning av signaturbyten är valfritt och kan om man så vill (i händelse av en felaktig processor) kopplas bort via menyn. I detta fall utgår programmeraren från att den håller på med en 89C4051. Som man kan se av figur 2 är programmerarens hårdvara ganska enkel och består i huvudsak av en master mikrokontroller som behöver innehålla endast 865 byte programkod. IC1, naturligtvis en 89C2051, utför allting som behövs för program-

meringen, den sätter upp det programmeringsmod som anges av användaren, omvandlar den programkod som sänds från PC-n över RS 232-interfacet till parallell form, skickar innehållet i ett redan programmerat minne (och annan information) tillbaka till PC-n och switchar spänningarna (0 V, 5 V och 12 V) i enlighet med timingdiagrammen. Men, även med den bästa viljan i världen, är allt detta inte så enkelt att uppnå med bara de 15 I/O-portar som finns på en 89C2051.

Kommunikationen med PC-n hantearas via portbenen P3.0 och P3.1 Dataöverföringen indikeras av att lysdioderna D4 och D5 blinkar. Portben P3.3 (interrupt 1) används som en ingång för att övervaka BUSY/READY-signalen hos den processor som programmeras ('processor under programmering, eller PUP). De individuella programmeringspulserna lämnas av ett annat portben, P3.5, medan P3.4 används för att ge de flanker på XTAL1 som inkrementerar adressräknaren i PUP.

Nu måste de återstående tio portarna lämna fem kontrollsignaler och åtta databitar, och här krävs lite uppfinningsrikedom. Hela port 1 på IC1 är ansluten till dess motsvarighet på PUP, men de fem portbenen P1.0-

P1.4 är också kopplade till IC3, en 8-bitars latch av typ 74HCT373. Låskretsens enableingång (ben 11) tas hög och latcharna är alltså transparenta, kontrollsignalens utgångar AD1, AD2, AD3 går direkt till PUP (som man kan se av tabell 1 har P3.5 och P3.7 alltid samma värde), AD0 koppar på +5 V matningen till mikrokontrollern via T1 (indikeras av LED D6) och AD4 styr appliceringen av programmeringsspanningen till RST/VPP. När dessa värden är inställda tas latchens enableingång låg igen och värdena lagras.

Atmelprogrammeraren använder en justerbar spänningsregulator för att ge spänningen på RST/VPP. Vid läsning, när det behövs 5 V, leder FET T2 och de två motstånden R6 och R8 är parallellkopplade. Om det behövs 12 V stängs T2 av och endast R8 befinner sig i kretsen. BS170 switchas av ett separat portben i IC1, P3.7. För att spänningen på VPP/RST skall vara vid jord efter en programmerings eller läsningsprocess (så att PUP kan tas ur sin hållare på ett säkert sätt) så switchas LM317 först till 5 V. Därefter sätts AD4 (från latchen) så att spänningen i punkten mellan R10 och D3 är lika med 5 V. Detta stänger av transistor T3 och R11 drar RST/VPP till jord.

Därmed har vi gått igenom alla portpinnarna på 89C051-an.

Mjukvara

På disketten för detta projekt, ordernummer 010005-11, och som finns att köpa från oss (se Läsarservice), finns ett assemblerprogram benämnt *Atmelpr.asm* för mastermikrokontrollern såväl som mikrokontrollerns objektкод i form av en *Atmelpr.hex* fil. Du kan också köpa en färdigprogrammerad mikrokontroller från oss, ordernummer 010005-41. Denna mjukvara kompletteras av *Programmer_eng.exe*, ett program skrivet i Delphi som styr alla funktionerna i Atmelprogrammeraren som läsning, skrivning och radering. Delphis källkod, *source_eng.zip* finns också tillgänglig.

Kretskortslayouten kan du hämta gratis från vår websida www.alltomelektronik.com. Titta under Download. Det finns ingen gratis nedladdning av den ovan nämnda mjukvara på grund av författarens licensrättigheter. Serieporten COM2 används som förval. När programmet startas öppnas ett fönster, delat i två delar (se figur 3). Till vänster väljs olika funktioner från en meny medan funktionerna bekräftas på den högra sidan, till exempel på följande sätt:

```
reading signature byte ok
programming lock bit 1 ok
error: file too large!
```

och så vidare.

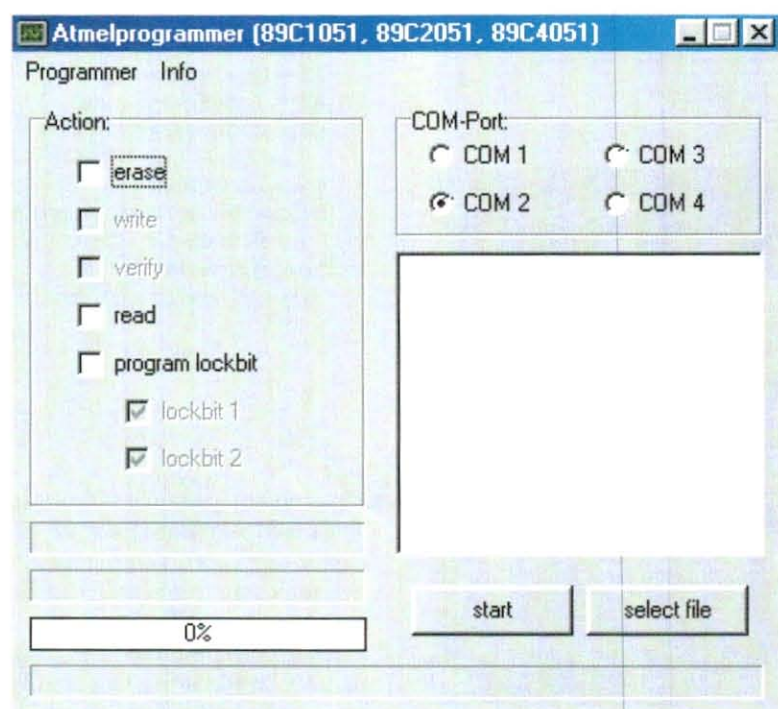
Om Read väljes kopplas alla övriga funktioner bort och det öppnas ett fönster i vilket du kan välja namnet på den fil i vilken innehållet i processorns minne skall skrivas ut. Om filtillägget är hex kommer en Intel Hex fil att genereras automatiskt. Om *bin* specificeras läses en binär fil.

Med knappen *Select file* kan du välja namnet på den fil som skall skrivas. Också här är *hex* och *bin* möjliga.

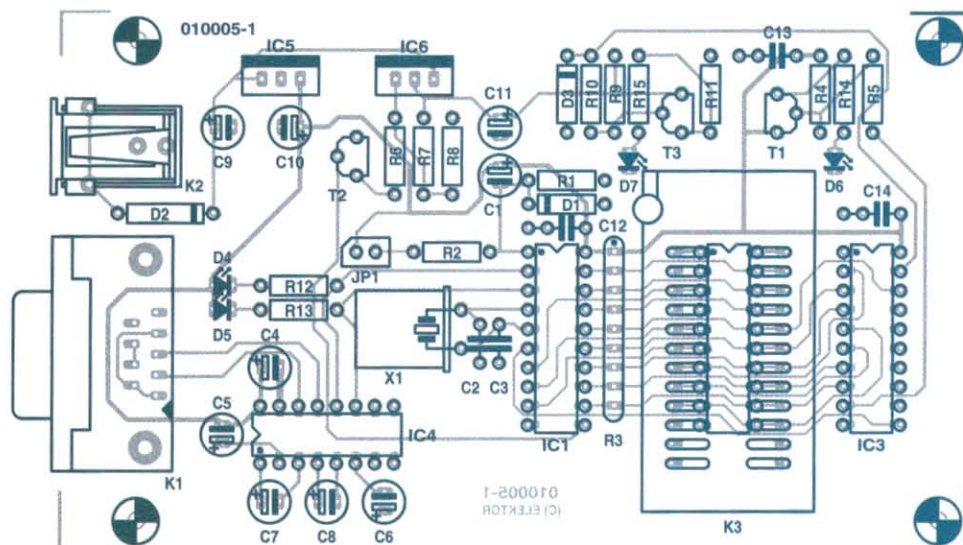
Filstorleken visas under mikrokontrollertypen. Under denna finns en stapel som visar hur de olika funktionerna framskrider. I området längst ner visas filnamn och sökväg.

I menyn under *Read signature byte* kan du välja om signaturbytet skall läsas från början eller inte. Detta alternativ skall alltid vara aktiverat såvida programmeraren inte kan läsa signaturbyten.

Informationen i signaturbyten visas under rutan där du väljer mikrokontrollertyp. Kommandona *Write*, *Read* och *Verify* behöver denna information. I annat fall utgår de ifrån att den komponent som skall programmeras är en 80C4051.



Figur 3. Kontrollpanelen för PC-programmet *Programmer_eng.exe*.



KOMPONENTLISTA

Motstånd:

R1 = 8kΩ
 R2 = 100Ω
 R3 = 8-pol 10kΩ SIL-nät
 R4, R5 = 1kΩ
 R6 = 1210Ω
 R7 = 274Ω
 R8 = 2260Ω
 R9, R10 = 10kΩ
 R11 = 4kΩ
 R12, R13, R14 = 1kΩ
 R15 = 2kΩ

Kondensatorer:

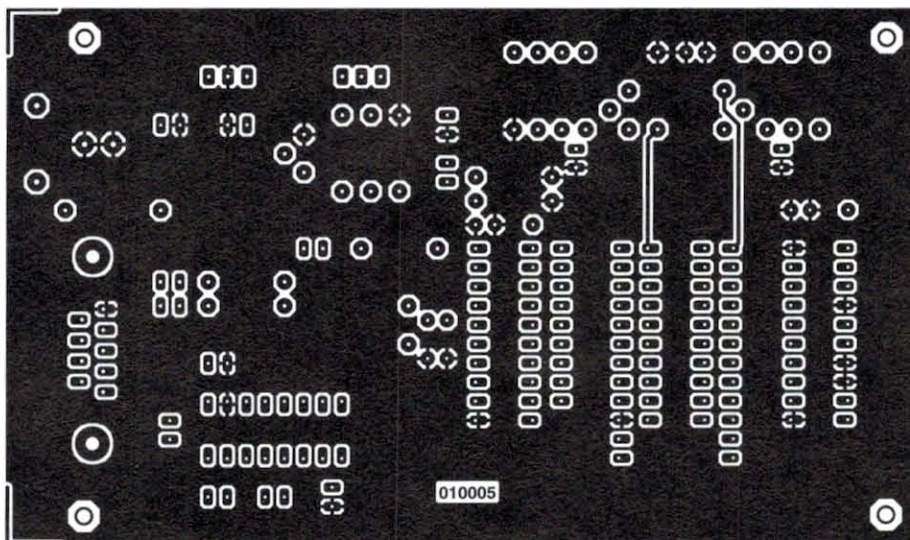
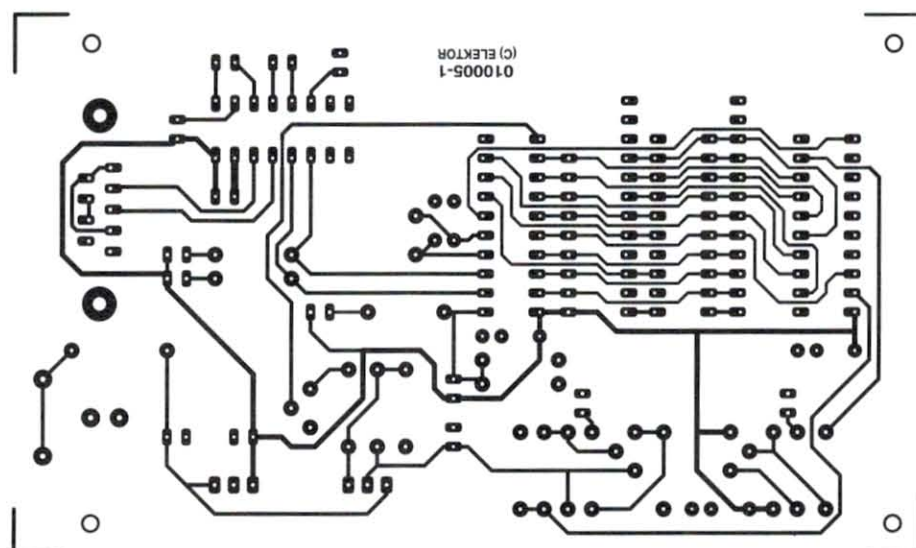
C1, C4-C10 = 10μF 63V radial
 C2, C3 = 22pF
 C11 = 1μF 25V radial
 C12, C13, C14 = 100nF

Halvledare:

D1 = 1N4148
 D2 = 1N4001
 D3 = zenerdiod 4V7, 500 mW
 D4, D5, D6 = LED, grön, dagsljus
 T1, T3 = BC557B
 T2 = BS170
 IC1 = AT89C2051-12PC (order-nummer **010005-41**)*
 IC2 (K3) = 24-pol nollkraftshållare (ZIF)
 IC3 = 74HC373
 IC4 = MAX232
 IC5 = 7805
 IC6 = LM317T

Övrigt:

JP1 = kortslutningsbygel
 K1 = 9-pol Sub-D sockel (hona), för kretskort, vinklade stift
 K2 = nätadapterjack
 X1 = 11.0592MHz kristall
 Låda, storlek 137x95x25 mm (t.ex. Pactec type WM46)
 Kretskort **010005-1***
 Diskett med mjukvara, ordernummer **010005-11***
 * (se Läsaerservice eller www.alltomelektronik.com)



Figur 4. Layout och komponentplacering för det dubbelsidiga kretskortet som designats för detta projekt.

Om programmeraren inte är ansluten visas meddelandet *Check programmer or COM port*. Kontrollera att serieanslutningarna (Rx, Tx och GND) är korrekt kopplade och inte korsade och kontrollera att programmeraren är påslagen. Om filen är för stor för den mikrokontroller som skall programmeras kommer det upp ett meddelande om detta. Ett felmedde-

lande visas också om något fel upptäcks i ingångsfilen.

Själva bygget och kalibreringen

Atmelprogrammeraren byggs på ett litet, dubbelsidigt, kretskort. Layouten och komponentplaceringen visas i figur 4. Programmeringshållaren behöver monteras lite högre än övriga komponenter så att den sticker upp genom lådans lock och så att PUP-en lätt kan sättas dit och tas bort. Detta är enklast att åstadkomma med hjälp av en virsockel och det finns plats på kretskortet för en sådan.

Den 9-poliga D-subkontakten passar perfekt i den rekommenderade lådan, men strömförsörjningsjacket sticker ut lite och måste filas ner.

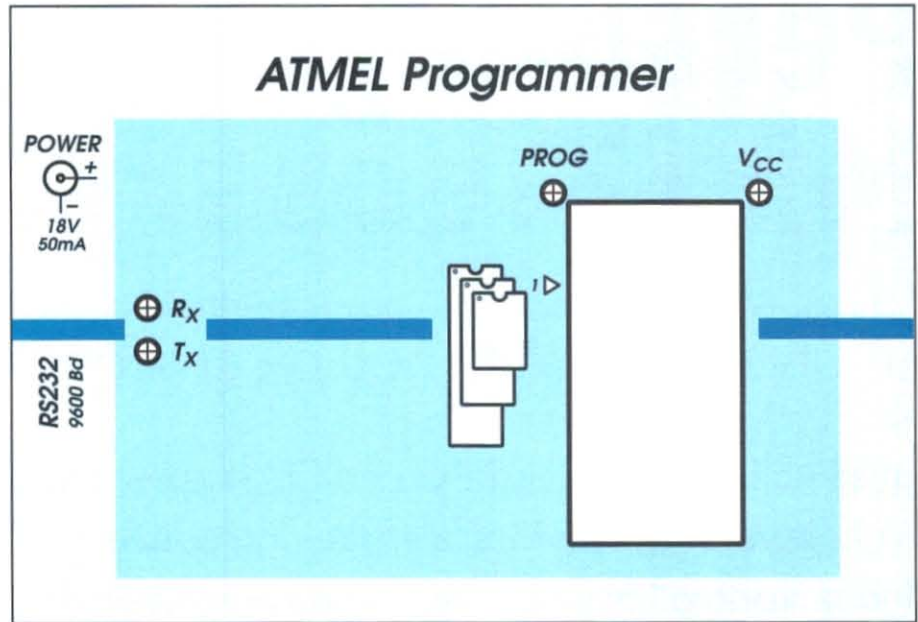
När kortet har bestyckats, kontrollerats och monterats in i lådan (på vilken frontpanelen i figur 5 passar precis) så är det bara en sak till som skall kollas innan du kan programmera din första krets: Programmeringsspanningen.

För att göra detta skall du ansluta programmeraren till din PC och välja vilken fil som helst som du vill skriva. Alternativet *Read Signature Byte* måste vara avaktiverat i annat fall visas felmeddelandet *No Processor*.

Under skrivprocessen kommer RST-benet att dras till 12 V. Om denna spänning inte är korrekt kan R8 ökas något (vilket ökar spänningen), eller minskas något (vilket minskar spänningen).

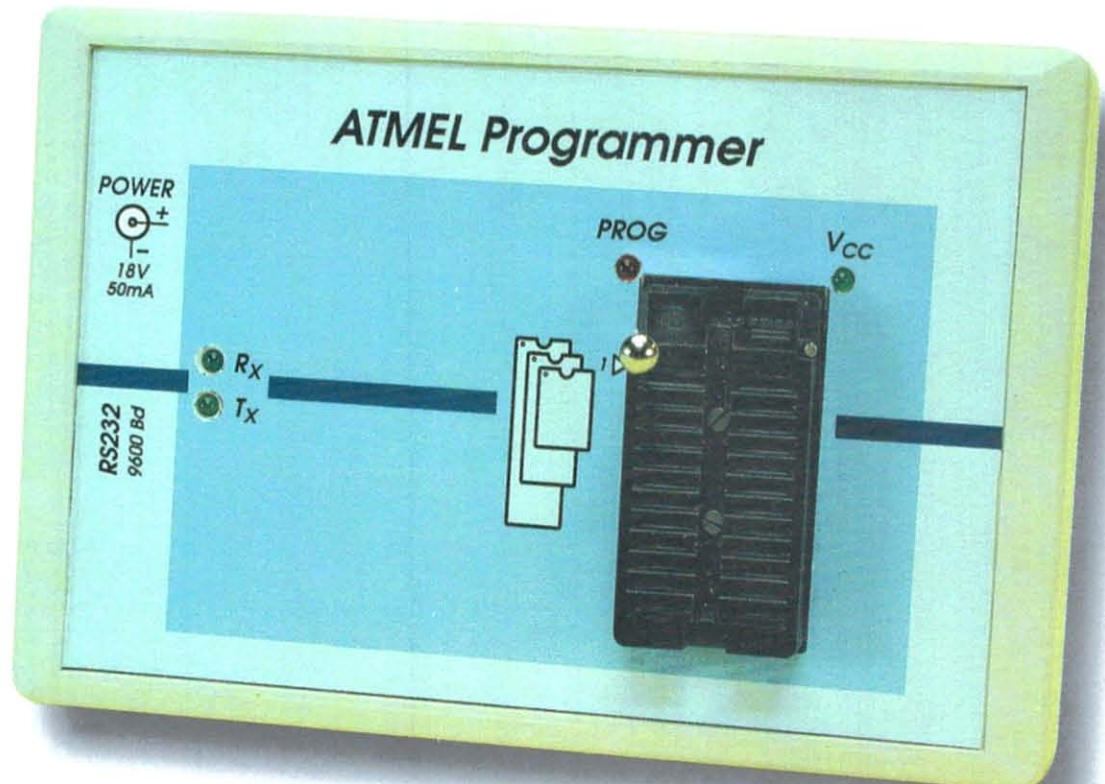
På exakt samma sätt kan 5 V matningen justeras via R6. Alternativet *Read File* skall då väljas och resistensen parallellt med R6 justeras på lämpligt sätt. Därmed är kalibreringen färdig och du kan börja att programmera din första mikrokontroller.

(010005-1)



010005-F

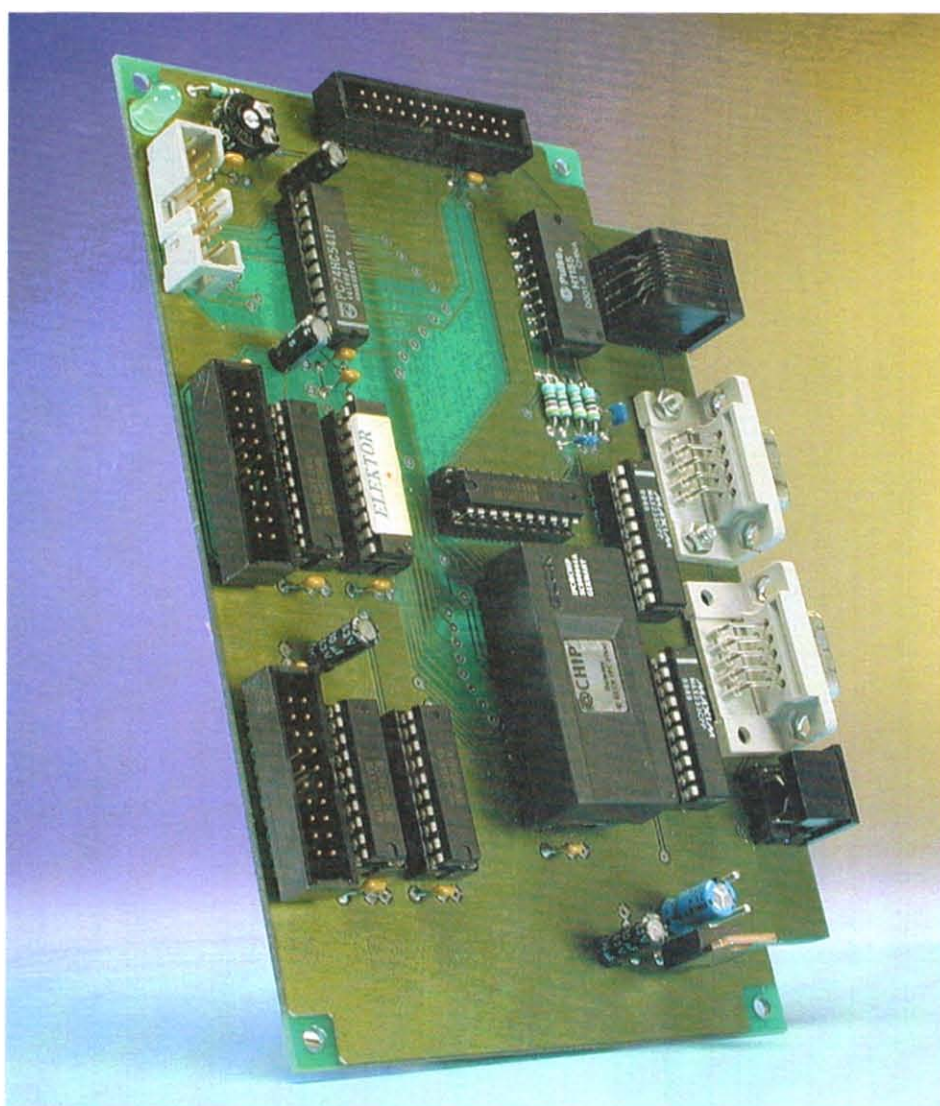
Figur 5. Frontpaneldesign för det där riktigt professionella utseendet.



Personlig Mini Webserver (I)

ett utvecklingskort med en embedded PC

En så kallad embedded PC i en DIL-kapsel bildar kärnan i denna Mini Webserver. I denna första artikel introducerar vi ett utvecklingskort för denna kontroller och visar hur du använder det. Efter detta kan du styra vilken utrustning som helst via Internet eller ett Ethernet nätverk.



Ibland kan en passionerad elektronikentusiast få det intrycket att modern elektronik inte kan implementeras i gör-det-självt form. Mikrokontrollerna har för många ben, ladda mjukvara in i DSP kräver professionell programmeringsutrustning med matchande pris, de passiva komponenterna är så små att du måste ha lupp för att kunna se vad du arbetar med, att löda dem med en vanlig lödpenna är hopplöst och, slutligen, det är inte alla som har utrustning för att tillverka kretskort i flera lager. Men, det behöver inte alltid vara så svårt och den Mini Webserver vi presenterar här är bevis på detta. Denna lilla, lättbyggda krets, som gör att du kan kommunicera på avstånd med elektroniska apparater (så långt som Internet når...) är den allra senaste (och ännu inte massproducerade) produkt som du kan köpa komponenter till för några tior i närmaste elektronikbutik.

I föregående två utgåvor av *Allt om Elektronik* ägnade vi oss åt den ganska komplicerade teorin bakom kommunikationen via ett lokalt nät eller Internet. Kontrollern i Mini Webservern måste kunna hantera en hel serie med kommunikationsprotokoll. Att implementera TCP/IP, HTTP, FTP, SMTP, ett operativsystem osv. i en mikrokontroller är ingen enkel uppgift. Tack och lov finns det dock några

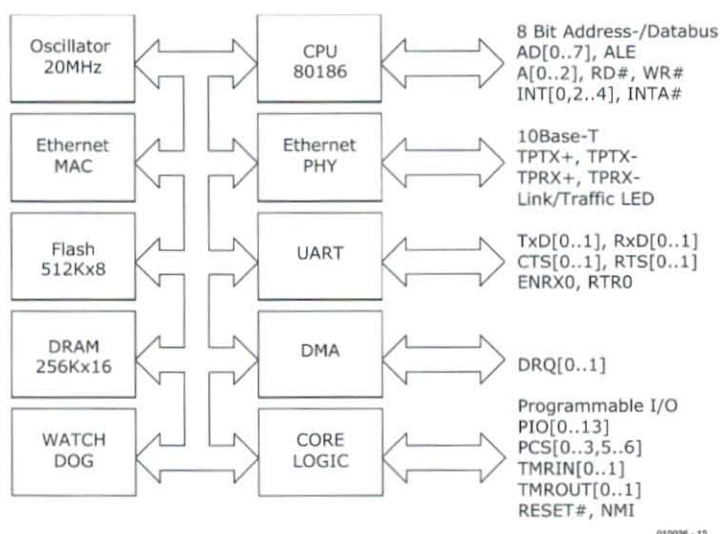
Egenskaper hos den personliga Mini Webservern

- 16-bitars 80186 CPU med 20 MHz klocka
- 512 kB RAM, 515 kB flashminne
- RTOS med Flash datasystem
- Nedladdning av mjukvara via RS232 eller Ethernet
- Protokoll: TCP/IP, PPP, HTTP, FTP, Telnet, POP3, SMTP, ICMP och DHCP
- Modemanslutning: PPP via 9-pol d-sub
- Ethernet: 10BaseT med PHY via RJ45 kontakt
- Terminalanslutning: snabb serieport med RS323 (RXD, TXD, CTS, RTS) via 9-pol d-sub
- I²C buss (master) via 6-pol mini-DIN kontakt
- Watchdog
- 2 timerutgångar, 2 timeringångar
- Detektering av spänningsbortfall (NMI) med sparning av data
- Intel-kompatibel multifunktionell A/D buss
- 16 digitala ingångar
- 16 digitala utgångar
- Direkt LCD-anslutning
- Adresserbar expansionsanslutning (A/D buss, I²C buss, interrupt, ALE och RD/WR)

kommersiellt tillgängliga mikrokontroller som redan har de nödvändiga funktionerna 'on chip', eller åtminstone packade i små supplementmoduler och vi behöver därför inte oroa oss för serverns nätverkssida utan kan koncentrera oss fullt ut på applikationskretsen.

Detta är precis ändamålet med den hårdvara som vi presenterar här. Den har i huvudsak designats som ett utvecklingssystem och ger därför tillgång till alla tillgängliga interface och

portar på mikrokontrollern via olika kontakter. Naturligtvis kan utvecklingskortet också användas som en prototyp genom att endast installera de nödvändiga komponenterna, om det finns tillräckligt med plats i applikationsenheten. Det finns en kontakt på kretskortet som man kan använda för anslutning av applikationsenheten eller en expansion för utvecklingskortet, som D/A och A/D omvandlare eller ytterligare interfacekomponenter.



Figur 1. Blockschemat för IPC@CHIP SC12 single-chip PC.

CPU-funktioner

- Multiplexad adress/databuss (8 bitar)
- 6 programmerbara chip select
- 6 interrupt-kapabla ingångar
- En extern interruptkontroller kan anslutas
- Tvåstegs hårdvaru watchdog
- 3 timers
- 2 snabba räknaringångar
- 2 snabba räknarutgångar
- 2 DMA-kontroller
- 10BaseT interface
- 2 asynkrona serieinterface
- 1 seriellt synkront interface (SSI)
- I²C buss-master kapabel
- Upp till 14 programmerbara I/O-pinnar
- Reset/Power Fail (NMI) generator

Internet-PC på ett chip

En av de få 'embedded' mikrokontroller som finns tillgängliga för oss vanliga dödliga är IPC@CHIP SC12, som tillverkas av Beck i Wetzlar, Tyskland. SC12 är byggd i en 32-bens DIL-kapsel. Som man kan se av blockschemat i figur 1 baseras den på en 80186-processor-kärna som klockas vid 20 MHz och har 512 kB vardera av DRAM och flashminne. Denna grundkonfiguration kompletteras med servicefunktioner som seriell kommunikation via en UART, ett I²C-interface, ett direkt bussinterface och 64 kB minne. SC12 stöder Ethernetkommunikation (10BaseT) via tvinnad tråd med hjälp av Media Access Controls (MAC) per IEEE 802.3.

SC12 överensstämmer med en ISA-arkitektur som har modifierats på några punkter, utan ISA-buss och utan videointerface. Programmeringen av serieportarna, DMA, PIC och timer skiljer sig också från den på en standard IBM PC.

De viktigaste CPU-funktionerna listas i rutan ovan. Förutom denna hårdvara finns det ett integrerat operativsystem som kallas RTOS, vars likhet med DOS ger det välkänt utseende.

Beck kallar, på goda grunder, resultatet för en 'single-chip PC', som är kompatibel med alla utvecklingsverktyg och applikationer för 80C 186 mikrokontroller. Det betyder att alla som redan har programmerat en sådan kontrollert i C, Pascal eller assemblyspråk inte kommer att ha några problem med SC12. Detta är mycket uppmuntrande eftersom utvecklingsverktyg för 80186 och DOS-kompatibla oper-

ativsystem är som sandkorn på en strand, billiga och tillgängliga i massor.

Skulle du ändå inte lyckas med att få ett program att fungera så kan du få hjälp från flera applikationsexempel och program (med källkoder) för grundfunktionerna hos SC12 från både Becks och *Allt om Elektronik* websidor. Naturligtvis är en sådan högteknologisk komponent inte särskilt billig. Priset för en SC12 (september 2001) är 66 Euro och en Borland C kompilator kan fås för 49 Euro om den köps tillsammans med kontrollern. Slutligen behöver du ytterligare 25 Euro (för en en användarlicens för operativsystemet RTOS med multitasking DOS, TCP/IP-stack, Webserver med CGI-interface, FTP-server, Telnetserver och en API för programmering i C, även om de sista av dessa endast behövs för professionella applikationer. Priserna exklusive moms. Med tanke på den lilla mängd hårdvara och mjukvara som du måste bidra med själv och de möjligheter som denna högpressterande Mini Webserver erbjuder så är detta dock ett mycket acceptabelt pris för användaren. Produktområdet hos Beck omfattar också ett litet utvecklingskort (ordernummer DK40), som dock endast har TTL-interface (med RJ12/45-kontakter) och åtta buffrade I/O-anslutningar.

Bussar och interface

SC12 är inte oskiljaktigt länkad till hårdvaran på utvecklingskortet. Du har absolut inget krav på dig att konfigurera en serieport som en terminalport (Tx/D0, Rx/D0) för avbuggningsändamål. Du kan också klara dig utan någon modemanslutning om din applikation uteslutande kommer att använda Ethernet. I det senare fallet blir också det andra seriein-terfacet ledigt.

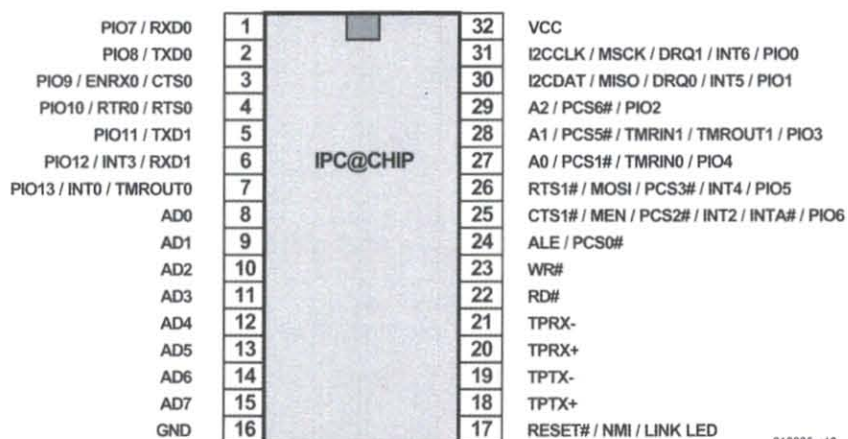
På applikationssidan finns det många portar på SC12 som har flera funktioner. Om du inte behöver någon I²C-buss, till exempel, kan du fritt programmera motsvarande portledare, eller använda dem som interruptkällor.

Vad vi nu sagt är argument för att ta en närmare titt på funktionerna hos de olika benen. Figur 2 visar bentilldelningen på SC12, med många dubbla och tredubbla funktioner.

Adress/Databuss

Det första som väcker vår uppmärksamhet är den synkrona, nivåkänsliga multiplexade adress och databussen (AD0-AD7). De undre åtta bitarna i adressen läggs ut på denna buss i den första perioden av busscykeln, följt av tre dataperioder. Adressfasen kan undertryckas.

Adressbussen kompletteras med de icke-multiplexade, synkrona, adressutgångarna A0-A2. Dessa adressbitar är giltiga under ena



Figur 2. Många av benen på SC12 är multifunktionella.

halvan av klockperioden före AD0-AD7.

En stigande flank på den synkrona ALE-utgången (Address Latch Enable) indikerar en giltig adress på den kombinerade adress/databussen. WR och RD aktiverar skriv och läscykler.

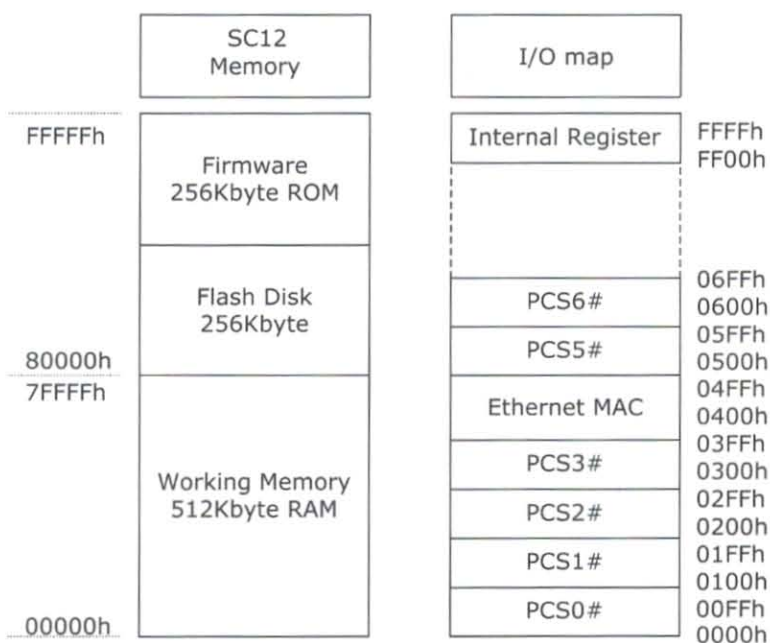
riktning och aktiva nivåattribut är programmerbara. Vissa inställningar är fördefinierade efter en power-on reset, och måste omkonfigureras efteråt av mjukvaran på lämpligt sätt för att erhålla de önskade inställningarna.

Programmerbar chip select

De synkrona väljarutgångarna PCS0-PCS6 styr I/O-accesserna till de perifera minneskretsarna. PCS-utgångarna arbetar tillsammans med den multiplexade A/D-adressbussen.

Programmerbara I/O-ben

De 14 benen PIO0-PIO13 är synkrona open-drain portlinjer, vars funktion,



Figur 3. Minneskonfiguration och minnesmappade I/O basadresser.

Interrupter och timers

Benen INT0, INT2, INT3 och INT4 representerar maskbara asynkrona interruptingångar. Om en interruptsignal kommer på en omaskad ingång avbryts programmet till förmån för den rutin som den interruptvektorn pekar på. Interrupter synkroniseras internt, och kan nivå eller flanktriggas. INT1 arbetar för Ethernet MAC och är inte tillgänglig för externt bruk.

Om INT0 arbetar i kaskadmod, vilket betyder att flera interruptkällor är anslutna via en prioritetskodare, ändras INT2-ingången till den synkrona INTA-utgången. Om INTA är aktiv måste hårdvaran placera ett 8-bitars värde på databussen så att kontrollern kan identifiera interruptkällan.

Mikrokontrollern har två interna timers som kan kommas åt externt via ingångarna TMRIN0 och TMRIN1. Efter intern synkronisering räknar varje positiv flank på TMRIN upp motsvarande räknare (inkrementering). Om den inte används måste TMRIN vara externa kopplad till en Hög nivå.

Timerutgångarna TMROUT0 och TMROUT1 lämnar antingen enkel-pulser eller ett kontinuerligt pulståg med en programmerbar pulsfaktor. Den senare typen av signaler resulterar från en intern kombination av två interrupt och timeringångar i PWD-mod (Puls Width Demodulator). PWD-driver TMRIN0 och INT2 och de inverterade TMRIN1 och INT4. Om INT2 och INT4 är aktiverade, och timern är korrekt konfigurerad, motsvarar pulsfaktorn förhållandet mellan de två timervärdena. I PWD-mod kan TMRIN0, TMRIN1 och INT4 användas som PIOs. Om de inte används ignoreras de internt.

10BaseT Ethernet interface

SC12 inkluderar en komplett Ethernet kontroller/transceiver som är åtkomlig via TPRX+ och TPRX- ingångarna samt TPTX+ och TPTX- utgångarna. Transceivern sänder och tar emot signaler via en utgångstransformator och ett tvinnat kabelpar. En LED ansluten till LINK LED blinkar svagt om det finns en Ethernetanslutning och starkt när det finns datatrafik.

Asynkrona serieportar

SC12 stöder full-duplex överföring av två asynkrona seriella dataströmmar med 7-9 bitars data med transmissionshastigheter upp till 115200 baud via benen TxD0, TxD1, RxD0 och RxD1. Fritt valbar handskakning ges för varje port av CTS0, CTS1, RTS0, RTS1, ENRX0 och RTR0.

Direct Memory Access

Via DMA Request ingångarna DRQ0 och DRQ1 kan en extern komponent annonsera en dataöverföring via en av de två DMA-kanalerna. DRQ0 är flanktriggad och internt synkroniserad, medan DRQ1 är nivåkänslig. DRQ0 är inte låst (latchad) och måste förbli aktiv under överföringen.

Inter-IC buss

SC12 har ett I²C-interface vid benen I2CCLK och I2CDAT. Kontrollern kan driva upp till 127 externa slavar, men är alltid master.

Reset, spänningsavbrott

En nivå som ligger under 0,8 V på det asynkrona RESET-benet avbryter omedelbart alla pågående aktiviteter, nollställer det interna minnet och får CPU:n att hoppa till reset-adressen FFFF0h. Om systemets matningsspänning faller under 4,5 V, eller tiden för watchdoggen går ut, tvingas Resetbenet till jord.

Den synkrona, flankkänsliga NMI-ingången (Non-Maskable Interrupt) är den interruptkälla som har högsta prioritet. Den är helt oberoende av interruptkontrollern för INT. En NMIförfrågan avbryter programkörning, som hoppar till den programlokalisering som indikeras av NMI-vektorn. NMI används tillsammans med den interna watchdoggen.

Vi kan nu avsluta beskrivningen av anslutningarna med en summering av minneskonfigurationen och I/O-basadresserna, vilka visas i figur 3.

Hårdvaruplattformen

Kretsschemat över Mini Webservern i figur 4 låter oss nu se vilka funktioner hos de multifunktionella benen på SC12 som implementerats på utvecklingskortet. Den kombinerade multiplexade adress/databussen AD är demultiplexad av adresslatchen IC2. Adressbitarna A0-A4 matas till en adressavkodare i form av en GAL.

Tillsammans med chip select linjerna PCS1 och PCS5 samt läs/skrivsignalerna väljer den programmerade logiken mottagare för de data som följer efter adressen, under förutsättning att den är närvarande: 16 digitala utgångar, buffrade av IC4 och IC6, 16 digitala ingångar isolerade med IC5 och IC7 samt en 2-raders display. De digitala ingångarna och utgångarna är anslutna till de två kontaktorna K6 och K7, med anslutningarna för varje portben alltid placerade mitt emot varandra. Den demultiplexade adressen A0-A7 och data D0-D7 finns tillgängliga på en expansionskontakt tillsammans med I²C-bussen (SDA och SCL), interruptkälla 0 (INT0) och ALE-signalen. Väljarsignaler för expansionslogiken ges också av GAL-en, med separata signaler för skriv (EXT WR) och läs (EXT RD) operationer. Dessutom har +5 V matningen också dragits ut till expansionskontakten.

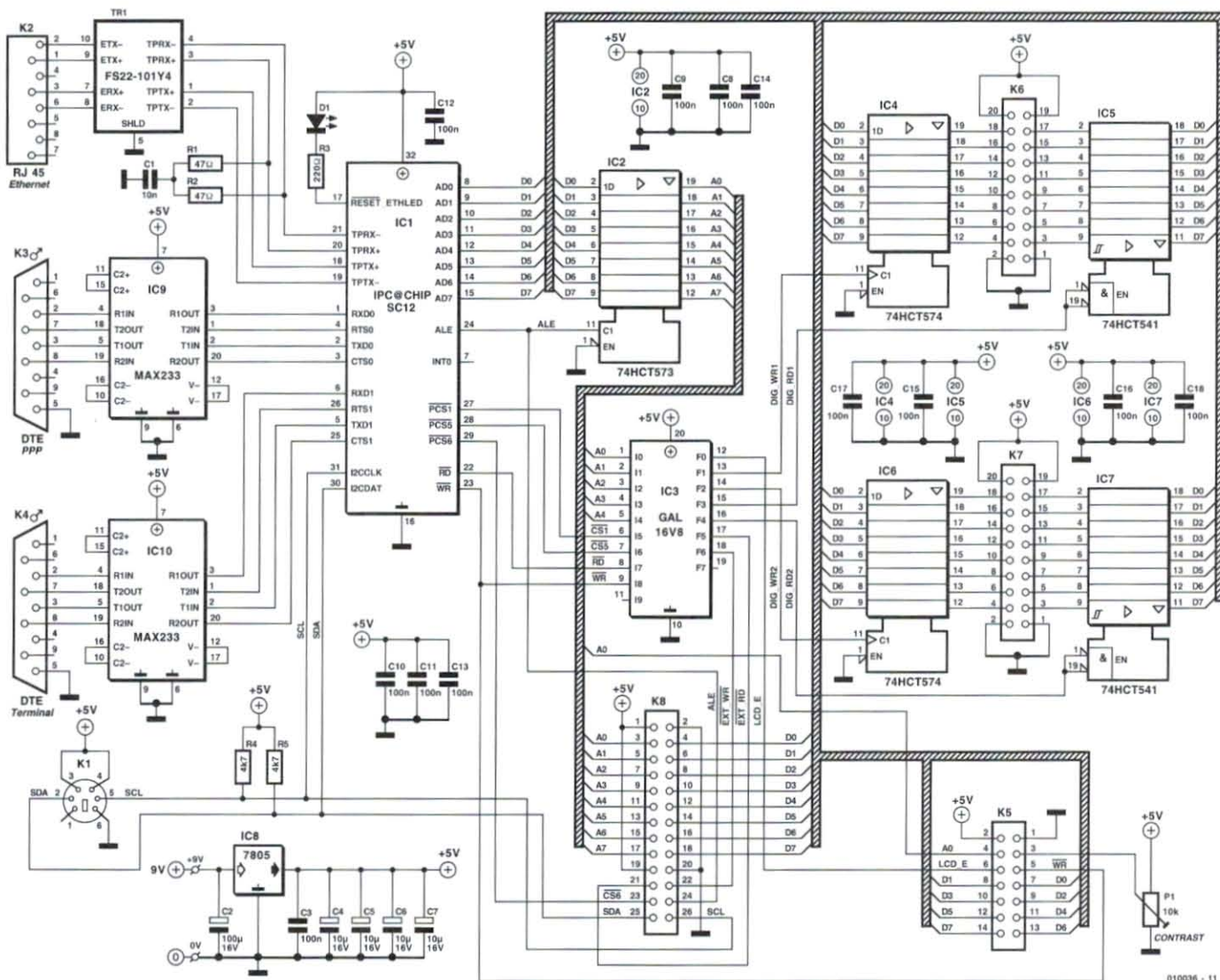
På den andra sidan av SC12 är allting standard: De två asynkrona serieinterfacen för modem och terminal används på det sätt de är avsedda för. Eftersom kontrollern naturligtvis endast ger TTL-nivåer, och endast kan hantera sådana nivåer, finns det två MAX233 interfaceomvandlare (liknar MAX232 men utan externa kondensatorer) i form av IC9 och IC10. Dessa tillåter att signaler med RS232-nivåer kan tas emot eller skickas via de 9-poliga d-subkontaktorna. Notera att båda dessa interface är DTE så du måste använda en nollmodemkabel för PC:n och en vanlig 1:1 kabel för modem.

Anslutningen mellan Ethernetkontrollern/transceivern i mikrokontrollern och nätverket kräver god terminering av de olika signalerna och elektrisk isolering. Dessa krav uppfylls av R1, R2, C1 och den speciella anpassnings-transformatorn Tr1. Det gemensamma benet för Reset, NMI och LINK LED används för att driva en lysdiod som visar att det finns trafik på Ethernet.

Dessutom har vi tagit hänsyn till att det finns en hel del I²C fans bland läsarna av *Allt om Elektronik* och har således försett I²C-interfacet med pull-upmotstånd och en mini-DIN kontakt (K1). På detta sätt kan du enkelt plugga in befintliga applikationskretsar för I²C som du tidigare byggt från beskrivningar i *Allt om Elektronik*.

Matningsspänningen är tillräckligt stabiliserad med en enkel +5 V regulator och några avkopplingskondensatorer. En nätadapter kan användas som strömkälla så länge den klarar att lämna minst 400 mA vid 9 V. Om den anslutna applikationskretsen också måste strömförsörjas skall du naturligtvis ta strömförbrukningen hos denna med i beräkningen.

Layouten för det dubbelsidiga kretskortet är gjord med tanke på utveckling. Allting är



Figur 4. Kretsschemat över utvecklingskortet för SC12.

enkelt att hitta och komponenterna sitter inte alltför nära inpå varandra. Alla kontakter sitter utefter kortets sidor. Du kommer att leta förgäves efter trådbygglar, ytmonterade komponenter och liknande svårödda komponenter.

Detta betyder att det inte är något problem med att bestycka kortet. Alla IC-kretsar monteras i hållare. Var uppmärksam med polarisationen så att du vänder polariserade komponenter åt rätt håll.

När du har avslutat lödjobbet, och noga inspekterat kortet kommer du att förväntansfullt längta efter nästa nummer av *Allt om Elektronik* där Mini Webservern visas i aktion för första gången. I följande artikel får du lära dig hur du arbetar med terminalmjukvara och hur du sätter upp en Ethernetanslutning, en HTTP-länk eller en FTP-server. Dessutom kommer vi att presentera några applikationskretsar som exempel och som visar hur du

KOMPONENTLISTA

Motstånd:

R1,R2 = 47Ω
R3 = 220Ω
R4,R5 = 4kΩ
P1 = 10kΩ trimpot

Kondensatorer:

C1 = 10nF
C3,C8...C18 = 100nF
C4-C7 = 10μF 16V radial
C2 = 100μF 16V radial

Halvledare:

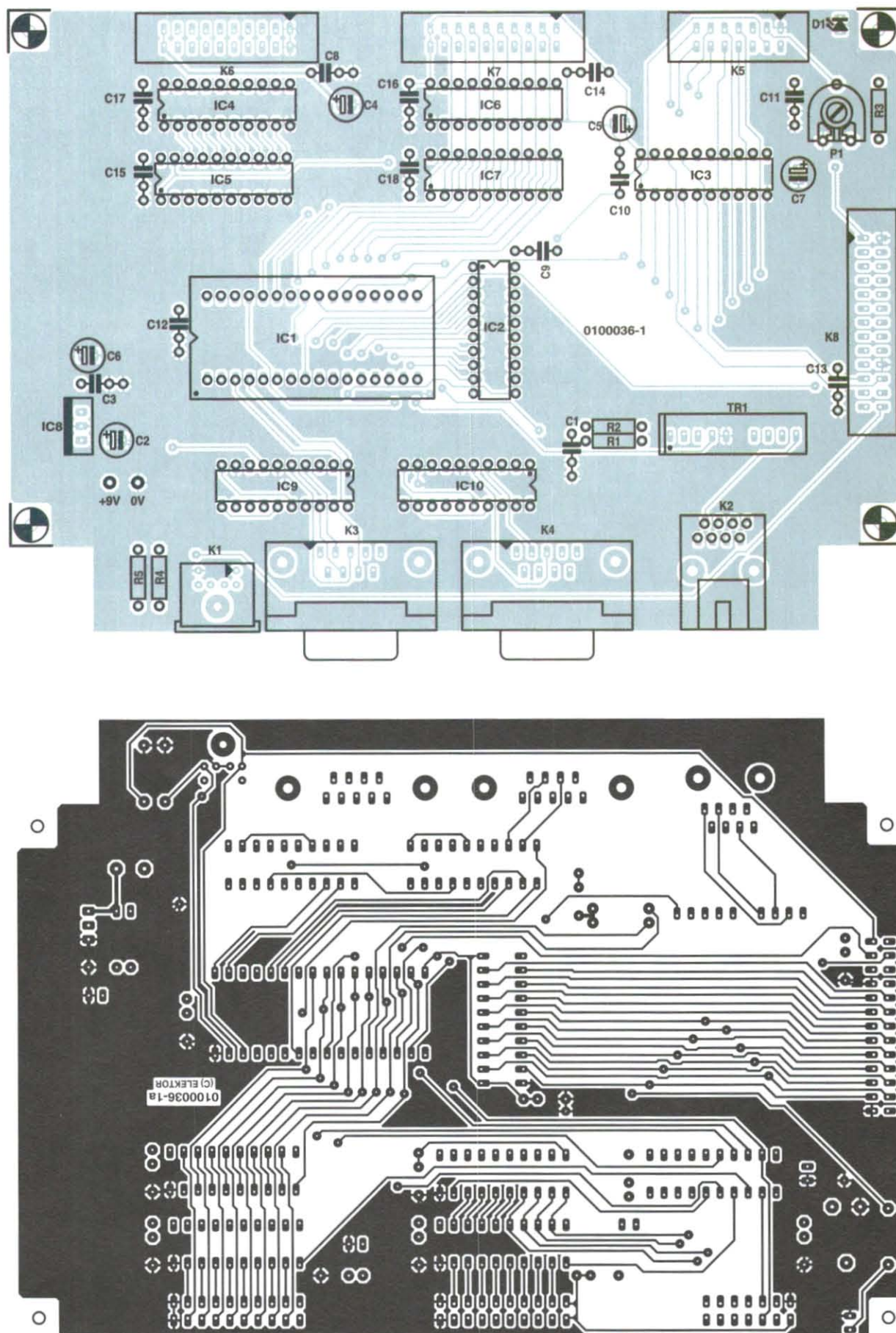
D1 = LED, grön
IC2 = 74HCT573
IC4,IC6 = 74HCT574
IC3 = GAL16V8 (programmerad, ordernr 010036-31)
IC5,IC7 = 74HCT541
IC8 = 7805

IC9,IC10 = MAX233CPP

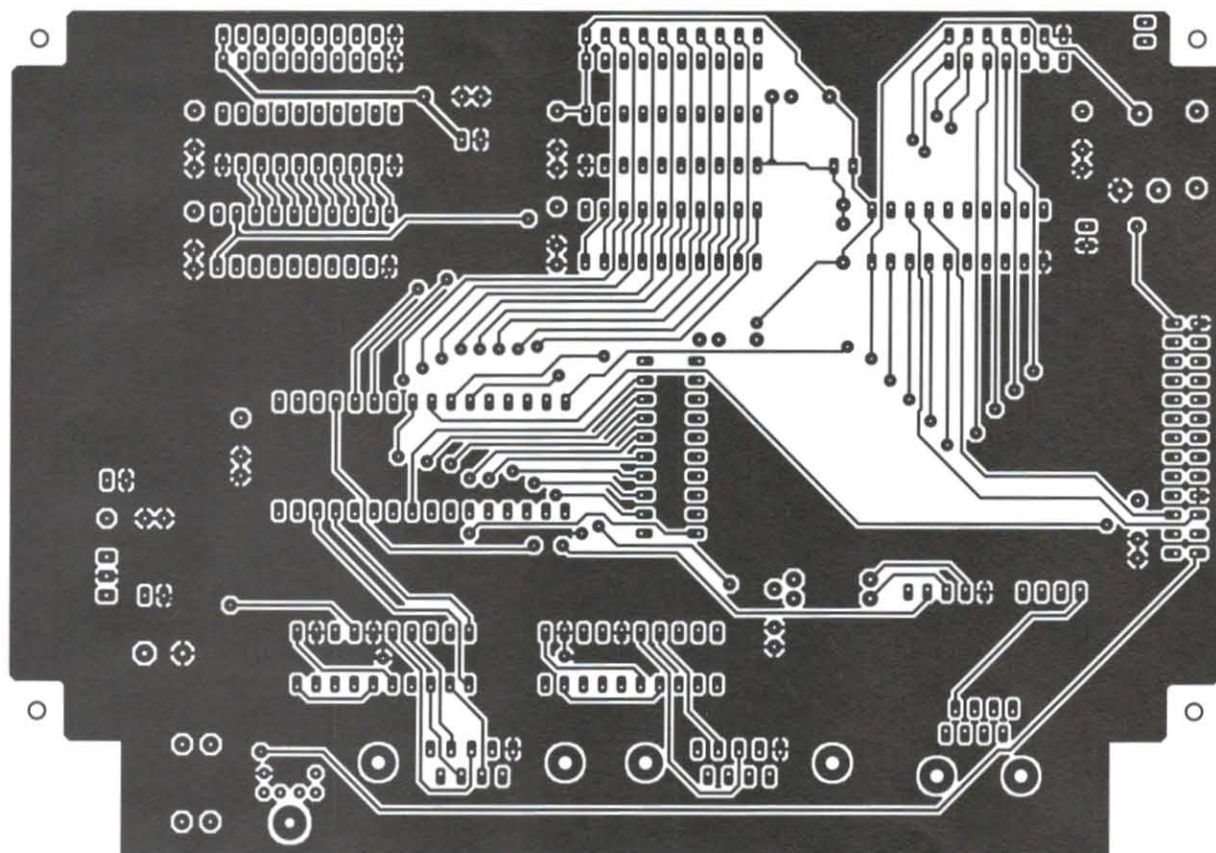
IC1 = IPC@CHIP SC12 (order från Beck, <http://www.bcl-online.de>)

Övrigt:

PC1,PC2 = lödstift
TR1 = FS22-101Y4 (order från Beck, <http://www.bcl-online.de>)
K1 = 6-pol Mini-DIN kontakt (hona), stiften 240 grader, för kretskort
K2 = RJ45 kontakt (hona), för kretskort
K3,K4 = 9-pol Sub-D plugg (hane), för kretskort, vinklade stift
K5 = 14-pol boxheader eller stiftlist
K6,K7 = 20-pol boxheader eller stiftlist
K8 = 26-pol boxheader eller stiftlist
Låda
Kretskort 010036-1



Figur 5. Det dubbelsidiga kretskortet har designats på tanke på att vara lätt att använda.



löser vissa hårdvaru och mjukvaruuppgifter. Om, emellertid, du är så intresserad av Mini Webservern att du inte kan bärga dig tills nästa nummer av *Allt om Elektronik* dimper

ner i brevlådan (för du prenumerar väl) så kan du finna lindring vid webadressen <http://www.bcl-online.de>

Här finns ett stort antal dokument som är relaterade till detta ämne och till SC12-kontrollern.

(010036-1)

Specialerbjudande!

Allt om Elektronik är Sveriges största tidning när det gäller praktisk elektronik. Här hittar du kompletta byggprojekt, praktiska tips och läsvärda artiklar om elektronik och data.

Vi har också under årens lopp givit ut ett antal böcker om elektronik. Dessa böcker har blivit oerhört populära. Den senaste heter '305 elektroniska kretsar'. Om du börjar prenumera på Allt om Elektronik får du denna bok gratis. Ordinarie pris är 349:-.

Om du beställer en 2-års prenumeration får du dessutom boken 'Ytmonterade elektronikprojekt'.

Priser för en helårsprenumeration är 485:- (Norge 420:- NOK + porto)

Priset för en 2-årsprenumeration är 875:- (Norge 755:- NOK + porto)

Samtliga erbjudanden gäller till den 30/11 2001.

☐ 1 år (boken 305 gratis)

☐ 2 år (305-an + Ytmonterade elektronikprojekt)

Namn

Adress

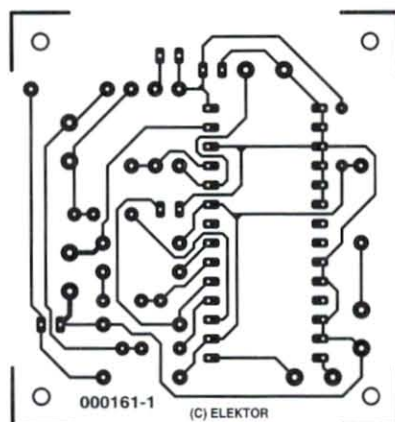
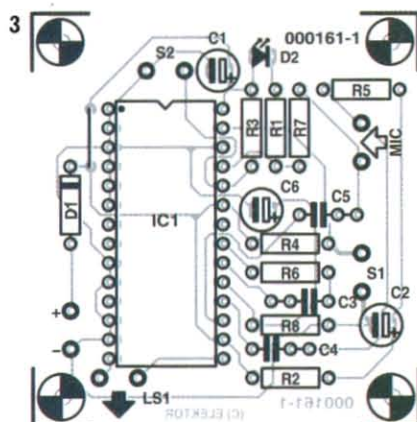
Postadress

Skicka kupongen (eller en kopia) till Allt om Elektronik, Box 178, 444 22 Stenungsund

Du kan också ringa PressData, 08-799 63 25

Du kan också beställa direkt via vår websida: www.alltomelektronik.com

Glöm inte att ange namn och adress. Presenten skickas när prenumerationen är betald.



KOMPONENTLISTA

Motstånd:

R1 = 470Ω
R2 = 100kΩ
R3, R4 = 100kΩ
R5, R7 = 10kΩ
R6 = 5kΩ
R8 = 470kΩ

Kondensatorer:

C1 = 100μF 16V radial
C2 = 220μF 16V radial
C3, C4, C5 = 100nF
C6 = 4μF 16V radial

Halvledare:

D1 = 1N4148
D2 = LED, dagsljus
IC1 = ISD1416 (Conrad Electronics #164984)

Övrigt:

S1 = tryckknapp med slutande kontakt
S2 = tryckknapp med slutande kontakt eller lägesbrytare (Conrad Electronics #700444) eller fotodiod (BPW34)
MIC = elektretmikrofon (Conrad Electronics #302155)
PC1, PC2 = lödstift
LS1 = högtalare 16Ω

plingarna inte lagras digitalt, utan i en slags halv-analog form med en av 256 olika nivåer i varje minnescell. Detta ger en betydligt högre lagringstäthet än vad som är möjligt med konventionell digital lagringsteknik och garanterar dessutom en hög inspelningskvalitet.

EEPROM-et behöver inte skrivas till på en gång. Flera meddelanden kan spelas in och spelas av individuellt. Ingångarna A0-A7 används för att konfigurera komponenten och adressera minnet.

IC-n har fyra kontrollanslutningar, en flänkänslig och en nivåkänslig avspelningsingång, en inspelningsingång och en utgång som dras låg under inspelning. Denna utgång aktiverar mikrofonen och tändar en LED. Ett aktivt 5-poligt

utjämningsfilter används under avspelnningen, anslutet till en utgångsförstärkare med symmetriska utgångar. Som man kan se av kretsschemat i figur 2 kan en 16 ohms högtalare anslutas direkt till dessa utgångar. Kvaliteten på denna högtalare bestämmer i stort sett kvaliteten på ljudåtergivningen. Den externa kretsen runt ISD1416 är minimal. Elektretmikrofonen ansluts symmetriskt via två kopplingskondensatorer och aktiveras endast i inspelningsmod när anslutningen RECDL är låg. Det är bara då som lysdioden tänds.

Inspelningsingången är ansluten till en tryckknapp medan avspelningsingången kan anslutas till en lägesbrytare, en tryckknapp eller en fotodiod. Den flänkänsliga avspelningsingången används och avspelnningen kommer därför inte att stoppa när den väl startats. Fotodioden (typ BPW34) kan monteras parallellt med, eller i stället för, en switch eller en tryckknapp. Denna kan monteras i mitten av en måltavla och när skytten träffar målet med en laserpekare belönas han/hon med applåder och visslingar. Applikationerna är oändliga.

Kretsen kan byggas på det kretskort som visas i figur 3 (finns inte att köpa färdigt från oss). Layouten och komponentplaceringsritningen kan du hämta på vår websida under Download, nummer 8/2001. Referensnummer 000161-1. Kretsen är såpass enkel att den är lätt att bygga på en bit experimentkort (veroboard).

Strömförbrukningen under avspelnning ligger runt 25 mA. Efter avspelnning går IC-n in i 'power-down' mod och då drar kretsen endast 90 μA. De två litiumceller (typ CR2032) som används som strömförsörjning kommer därför att räcka under lång tid.

(000161-1)

Växlande blinker

K. Lorenz

Detta är en astabil multivibrator för allmänbruk som alternativt aktiverar två tunga laster via ett relä (i detta fall är belastningarna 12-V glödlampor). I motsats till en 'analog' flip-

flop är det här inte nödvändigt att använda effekttransistorer med kylelement. Denna växlande blinker kan därför byggas på enklare sätt, mer kompakt och till ett lägre pris.

I viloläge laddas kondensatorn C1 via R1 och samtidigt urladdas den via R2 och P1. P1 måste justeras på så sätt att

det finns tillräckligt med ström för att switcha till transistor T1. Detta skall ske när spänningen på kondensatorn är ungefär 1,2 V om du använder en BC517. Som en följd av detta dras reläet.

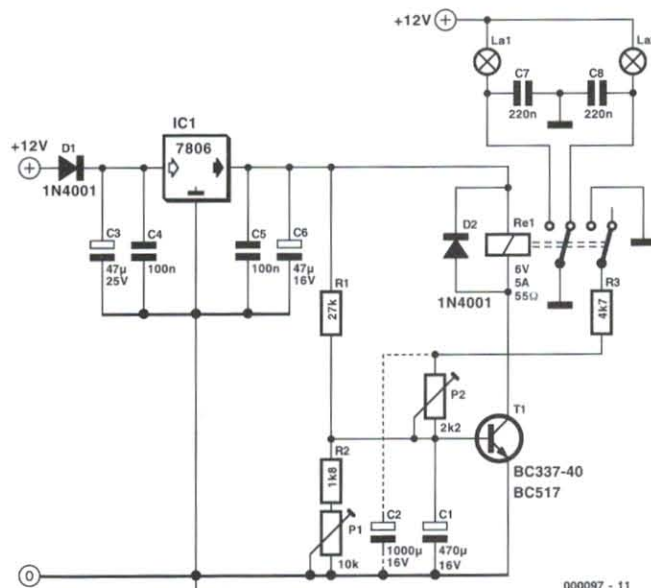
Detta får R3 och P2 att anslutas parallellt med R2 och P1. P2 skall justeras så att det inte finns tillräckligt med ström kvar för att ge basström till T1. Detta får spänningen på C1 att falla och kort därefter stängs transistorn. Relät faller då ut och alltihop börjar om igen.

Matningsspänningen kan komma från en ostabiliserad nätadapter på 12 V (till exempel). Strömförbrukningen beror i huvudsak på de två lasterna eftersom själva kretsen endast drar reläets märkström.

Varje last ansluts direkt till matningsspänningen medan blinkerkretsen erhåller en stabiliserad spänning via spänningsregulatorn IC1. Dioden D1 skyddar kretsen mot ingångsspänning med omvänd polaritet.

För att ställa in kretsen skall du först vrida P1 till minsta och P2 till största resistens. Vrid nu upp P1 långsamt (!) tills relät dras in. Upprepa samma process med P2 tills reläet släpper igen. Med hjälp av denna grundprocedur kan du välja både blinkfrekvens och önskat till/från-förhållande.

Vi använde en BC337-400 för T1. Om denna är svår att få tag på kan du använda en BC517 (Darlington). Den korrekta funktionen hos kretsen beror också på vilken typ av relä som används. I en av prototyperna drogs reläet in OK och aktiverade R3 och R2, men vägrade att göra någonting



mer. Om ditt bygge gör likadant (och du är beroende av en viss typ av relä) kan det hjälpa om du lägger in kondensatorn C2 i kretsen (visas streckad i schemat) för att litegrann fördröja effekten av inkopplingen av P2 och R3.

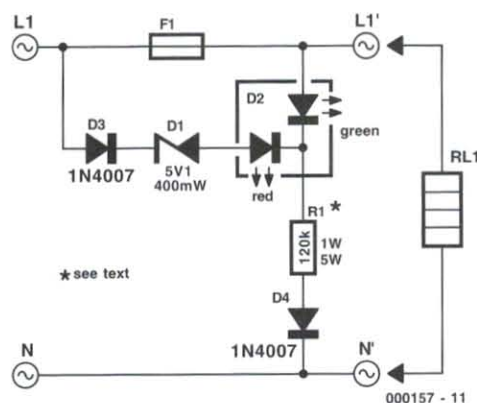
(000097-1)

Indikator för trasig säkring

S. Lenke

Denna krets visar när en apparat fungerar eller när dess säkring har gått sönder. Det är en utveckling av en liknande indikator som vi publicerade 1995, men den är mindre och billigare än den tidigare konstruktionen, trots att den kan arbeta med vilken nätspänning som helst. En tvåfärgad lysdiod (D2) med separata anodanslutningar visar **fungerar** (grön) eller **säkring trasig** (röd). Motståndet R1 begränsar strömmen genom LED-en till ca 2 mA. Lysdioden lyser alltså ganska starkt. Om du vill att den skall lysa ännu starkare kan du minska värdet på motståndet.

Zenerdioden förhindrar att den röda och den grön LED-en lyser samtidigt under normala förhållanden. När säkringen är hel är lysdioderna parallellkopplade, men det större spänningsfallet över den röda LED-en i kretsen betyder att bara den gröna lyser. Dioderna D3 och D4 förhindrar att lysdioderna skadas under den negativa halvan av AC-spänningen. Om kretsen används med DC kan dioderna tas bort. Om kretsen används för att övervaka säkringen i en nät-



ansluten apparat måste du tänka på att komponenterna inte är isolerade från nätet och spänningarna över dem kan vara livsfarliga. Rör inte kretsen när den är inkopplad på nätspänning! (000157-1)

Trimma CD-brännare

högre hastighet med ny firmware

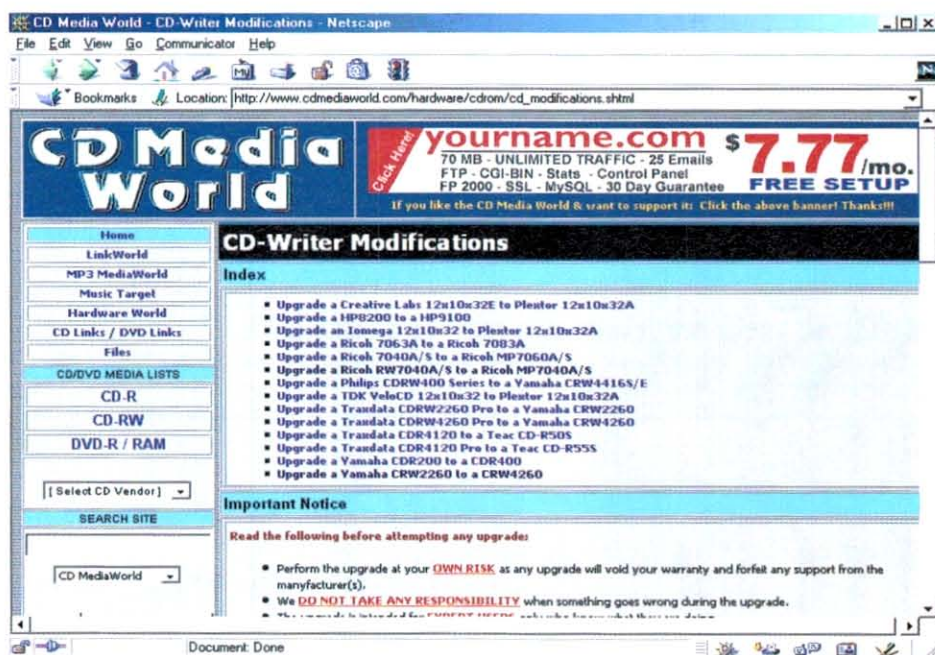
Av Harry Baggen

Det är inte bara bilar, motorcyklar och mopeder som går att trimma, du kan också snabba upp din datorutrustning. Att köra processorer med högre fart än de är avsedda för är en populär sport. Vad dock inte många vet, är att olika CD-brännare också kan 'uppgraderas' till bättre, eller snabbare, versioner.

Nuförtiden är en CD-skrivare, eller brännare som den mer allmänt kallas, praktiskt taget en standardkomponent i en modern dator. CD-R skivor är billiga och CD-brännare är det idealiska sättet att överföra stora datavolymer mellan datorer, göra backupkopior av program och filer, göra audio-CD med dina egna favoritlåtar och att generera MP3-CD.

Utvecklingen av CD-brännare har gått så fort under de senaste åren att det kommer nya modeller, med ännu högre hastighet, i stort sett varje månad. Och bland dem som har en lite äldre modell stiger längtan efter en som är snabbare.

Smarta datoranvändare har listat ut att, på samma sätt som med processorer, så är det möjligt att också mixa med CD-brännare för att få ut högre hastighet. Med några äldre modeller, som vissa Yamaha-enheter, innebär detta några ändringar på kretskortet. Majoriteten av modifieringar idag handlar dock om att anpassa firmware, vilket man kan se som ett 'BIOS' hos CD-brännaren. Om du använder en annan typ av firmware kommer CD-brännaren att fungera som en annan modell. Detta är möjligt eftersom det bara finns ett begränsat antal tillverkare av mekaniken och elektroniken i en CD-brännare. Innanmätet hos olika typer av CD-brännare är ibland så lika varandra att det räcker med en enkel förändring av firmware för att få en annan hastighet. Till exempel, en 8x



Sonyenhet (CRX-140E) kan uppgraderas till en 10x brännare genom att 'flasha' den med firmware från en HP-brännare (som har en Sonyenhet 'under huven').

Det finns två websajter som har samlat ihop ett antal modifieringar för CD-brännare: **CD Media World [1]** och **CDR-Info [2]**.

Det är definitivt intressant att experimentera med dessa modifieringar. Vi måste dock varna för möjliga skador på brännaren. Det kan hända att din brännare inte fungerar när du flashat den med annan firmware, och att den inte kan återställas till ursprungsskick. Prova därför helst det-

ta med en äldre brännare som du ändå tänkt att byta. Om någonting skulle bli fel är ju ingen skada skedd!

(015081-1)

Internetadresser:

- [1] CD Media World:
http://www.cdmediaworld.com/hardware/cdrom/cd_modifications.shtml
- [2] CDR-Info:
http://www.cdr-info.com/tips/hardware_oc.shtml