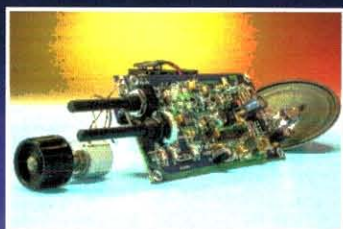


ELEKTOR Allt om **Elektronik**

Sveriges största specialtidning för tillämpad elektronik



AVR-
programmerare



VHF flygbands-
mottagare

Grundkurs i
mikrokontroller-
teknik

ISAK-kuben

8051-kompilerare

LJUSBARRIÄR FÖR FARTMÄTNING

hur snabb är din modellbil?



INTERPRESS 0173-04



7 388017 305909

RETURVECKA

24

Chefredaktör och ansvarig utgivare
Bill Cedrum

Redaktionen

Box 178
444 22 Stenungsund
Besöksadress
Kristinedal 40
444 47 Stenungsund

Redaktionssekreterare

Asta Russell

Utgivare

Electronic Press AB
Box 178
444 22 Stenungsund
Tel: 0303-77 04 90
Fax: 0303-77 04 90
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:- NOK + porto
Norge halvår 255:- NOK + porto
Norge 2 år 755:- NOK + porto

Annonser

Allt om Elektronik
Annonsavdelningen
Box 178
444 22 Stenungsund
Tel: 0303-77 04 90
Fax: 0303-77 04 90
E-mail: annons@alltomelektronik.com

Tryck

NDB-Printing
Zoeterwoude, Nederland 2002

ISSN 0280-8765

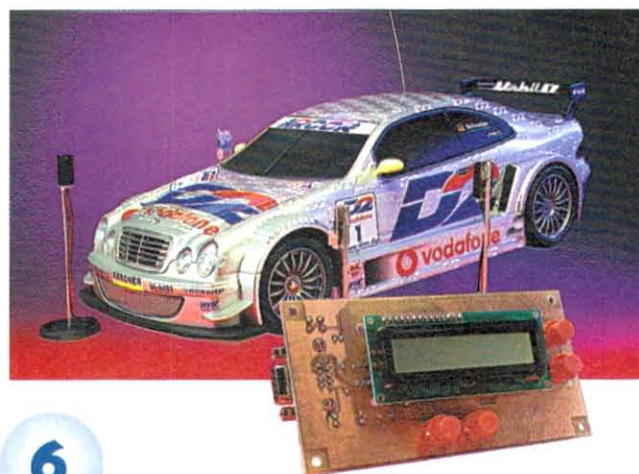
Samtliga artiklar, fotografier, kretskortslayouter och annat är copyrightskyddat och får endast återges efter utgivarens skriftliga tillstånd. Detta gäller även delar av materialet. Allt om Elektronik har, så långt som är möjligt, kontrollerat samtliga byggprojekt. Dock tar vi inget ansvar för fel som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

Copyright: Segment B.V.
(Beek, the Netherlands) 2002
P.O. Box 75
6190 AB Beek (L), The Netherlands

BYGGPROJEKT

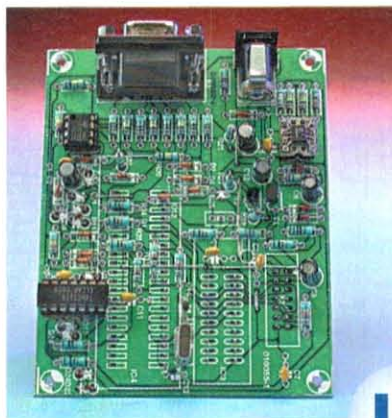
Hastighetsmätningssystem

Med det ljusbarriärssystem som beskrivs i denna artikel kan du göra noggranna mätningar av den absoluta hastigheten hos modellbilar, modellflygplan och andra rörliga föremål. Vid undervisning och träning bildar systemet en perfekt, kontaktlös fartmätare. Vem av er fotbollsspelare har den grymmaste sparken eller snabbaste svingen?



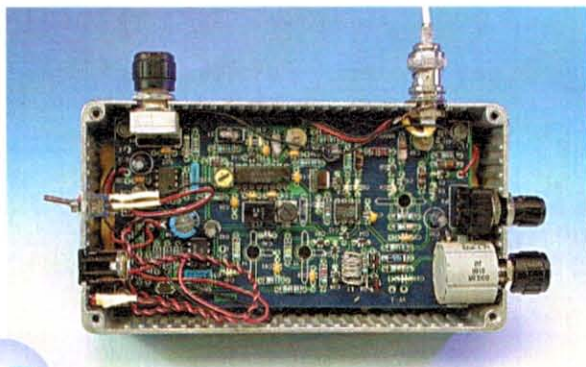
Enkel AVR-programmerare

Tack vare dess höga prestanda, Flash programminne, integrerade hårdvarufunktioner och låga effektförbrukning håller Atmels AVR mikrokontroller på att bli allt populärare, även inom det semi-professionella området.



Flygbandsmottagare

Denna mottagare, speciellt designad för VHF-flygbandet, kombinerar bra prestanda med enkel konstruktion, allt till ett överkomligt pris. Kretsen innehåller inga exotiska komponenter och kan justeras utan specialinstrument. Vi tycker därför att detta är en idealisk nybörjarmottagare för flygentusiaster med båsa fötterna fast placerade på markern.

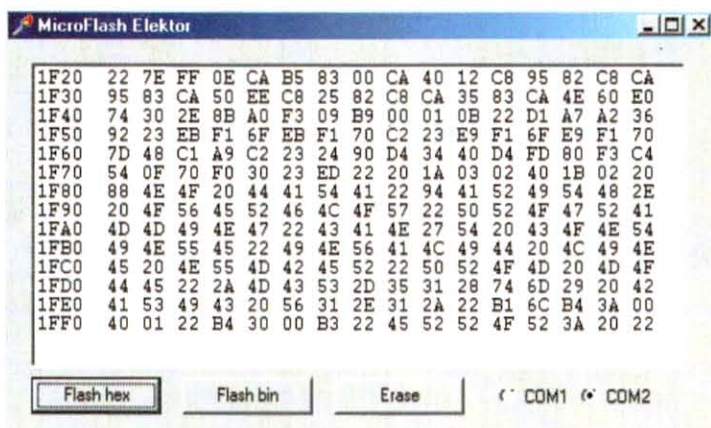


25

INFORMATIVA ARTIKLAR

Grundkurs i mikrokontrollerteknik

I de första två artiklarna i denna kurs arbetade vi med assembler, men nu är det dags för ett högnivåspråk, BASCI-52.



33

Byggprojekt

- 6 Hastighetsmätningssystem
En gör-det-själv mätare med IR-grindar
- 12 Fjärrstyrning med en mobiltelefon
Del 2: Programmering och användning
- 18 Enkel AVR-programmerare
För (nästan) alla AVR-mikrokontroller
- 25 Flygbandsmottagare
En dubbelomvandlande superheterodyn för 108-137 MHz NAV och KOM-mottagning
- 38 ISAK
Del 4: Basapplikationer
- 42 Mångsidig IC för modellbyggare
Del 2: Två kretskort och nio funktioner

Informativa artiklar

- 33 Grundkurs i mikrokontrollerteknik
Del 3: BASIC-52
- 48 Elektronik Online
8051-kompilerare
Freeware, shareware och demoversioner

Ovrigt

- 24 Läsarservice
- 50 Nästa nummer

Hastighetsmätningssystem

En gör-det-självt mätare med IR-grindar

Med det ljusbarriärsystem som beskrivs i denna artikel kan du göra noggranna mätningar av den absoluta hastigheten hos modellbilar, modellflygplan och andra rörliga föremål. Vid undervisning och träning bildar systemet en perfekt, kontaktlös fartmätare. Vem av er fotbollsspelare eller golfare har den grymmaste sparken eller snabbaste svingen?



Ljusbarriärer finns på fler ställen runt oss än vad vi kanske tänker på. Obemärkt gör de sitt jobb i snabbköpskassan där de scannar priset på varorna, känner av fordon i parkeringsgarage, styr banden i videobandspelare i övervakningssystem, håller koll på folk som går in och ur en byggnad. Normalt används infrarött ljus för detta eftersom det är osynligt för det mänskliga ögat och har en hög immunitet mot störande ljuskällor.

Medan enkla ljuskällor som arbetar utan moduler, som de ljusbarriärer som kollar om det finns en CD inmatad i din PC, har ganska

enkla funktioner, så har de ovan nämnda applikationerna i parkeringsgarage och säkerhetssystem betydligt högre krav när det gäller 'brus' immunitet. Lösningen, i nästan samtliga fall, är att använda en bärvåg på 36 kHz. Mottagaren konstrueras sedan till att enbart reagera på denna frekvens. Ett

mikrokontrollersystem används sedan för att analysera mottagarens utgångssignal och utvärdera längden på ett eventuellt avbrott. Detta görs för att förhindra felaktig trigging av ett larm när, till exempel, en insekt flyger eller kryper genom ljusbarriären.

Som visas i figur 1 kan man utföra fartmätningar med en ljusbarriär även om den bara har en sändare och en mottagare. En nackdel med

detta system är risken för inkonsekventa mätresultat om det rörliga föremålet har en oregelbunden form som bryter ljusstrålen flera gånger. Dessutom finns det inget som säger att mätobjektet följer samma, raka, bana genom barriären, vilket också detta kan ge felaktiga resultat.

Flera infraröda detektor-IC har använts i många projekt som publicerats i Allt om Elektronik under de senaste åren. En av dessa, SFH505 från Siemens, har blivit en slags standardkomponent för denna funktion. SFH505 och andra funktionellt kompatibla komponenter (inklusive IS1U detektorer från Sharp) hittar man i många apparater inom konsumentelektroniken. Även om dessa komponenter är mycket mångsidiga och kostnadseffektiva så är de mindre lämpliga för det projekt vi tänkt oss här, huvudsakligen på grund av följande aspekter.

– Utgångens reaktionstid när strålen

bryts har för många tidstoleransfaktorer;

– För att få ett acceptabelt område kan det infraröda ljuset inte moduleras hela tiden. Pulser behöver läggas till för att förhindra att styrelektroniken minskar mottagarens känslighet.

– På grund av de pauser som behövs inom ett modulerat system är hastighetsmätningar inte möjliga eftersom en brytning av strålen (av föremålet) inte kan skiljas från en paus (inlagd av systemet).

Trots den extra hårdvaran är det system som visas i figur 2 ett bättre alternativ, om inte annat så för dess mycket högre noggrannhet och pålitlighet när det används för att mäta hastigheten hos ett föremål som passerar genom barriären.

För att systemet skall vara så flexibelt som möjligt när det används i praktiken, kan du själv ställa in längden på den sträcka som föremålet

Egenskaper

- Hastighetsmätningar från 0.01 till 999 km/h
- Hastighetsvisning i m/s eller km/h på 1x16 teckens punktmatrisdisplay
- Visning av förluten tid (max. 16.777215 sekunder)
- Upplösning 1 μ s
- Avstånd mellan ljusbarriärerna justerbart mellan 1 cm och 255 cm
- Optisk inriktningshjälp
- Enstaka eller kontinuerlig mätning
- Matas från ett 9-V batteri
- Strömförbrukning ca. 45 mA.

skall färdas för att bryta de två ljusstrålarna. Praktiskt gör man detta genom att placera de två barriärerna på lämpligt avstånd från varandra och sedan 'informera' systemet om det exakta avståndet innan mätningarna påbörjas.

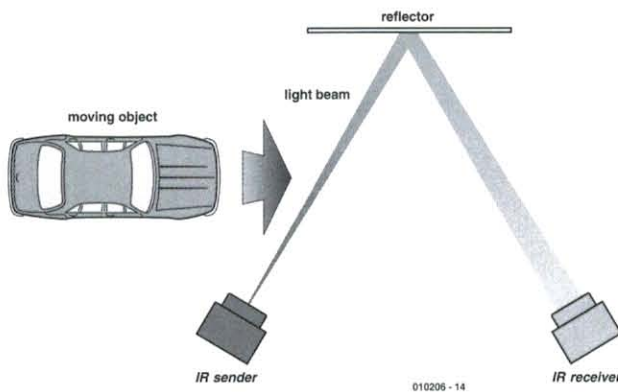
Sändare och mottagare

Kodenshi Company från Korea har utvecklat två moduler (sändare PIE-310 och detektor PID-310) för applikationer med ljusbarriärer. Tack vare den höga integreringen innehåller dessa moduler alla de komponenter som behövs för långdistansmätning i brusimmuna ljusbarriärssystem.

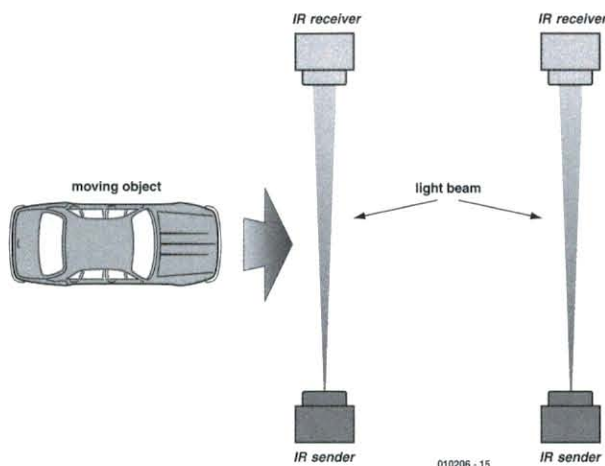
Från databladet (på koreanska) kan vi hämta följande viktiga egenskaper:

- IR LED med intern modulering och tillhörande fotodetektor i ett separat, kompakt, hölje med ett linssystem.
- Storlek ca. 17 x 8 x 7 mm
- Räckvidd 1 till 8.5 m
- Aktivt låg öppen-kollektorutgång
- Styringång på sändaren
- Högt tolerans mot omgivande ljus tack vare optiska filter och intern modulering
- Billiga sensorapplikationer vid större avstånd
- Lämpliga att använda i reflektionsmod
- 3-pinnars trådslutning
- Strömförsörjning 5 V/5 mA (mottagare) och 5 V/15 mA (sändare)
- Switchingshastighet 0.5 ms
- Vinkel vid halva känsligheten ± 5 grader
- Arbetstemperatur -10 till +60 °C

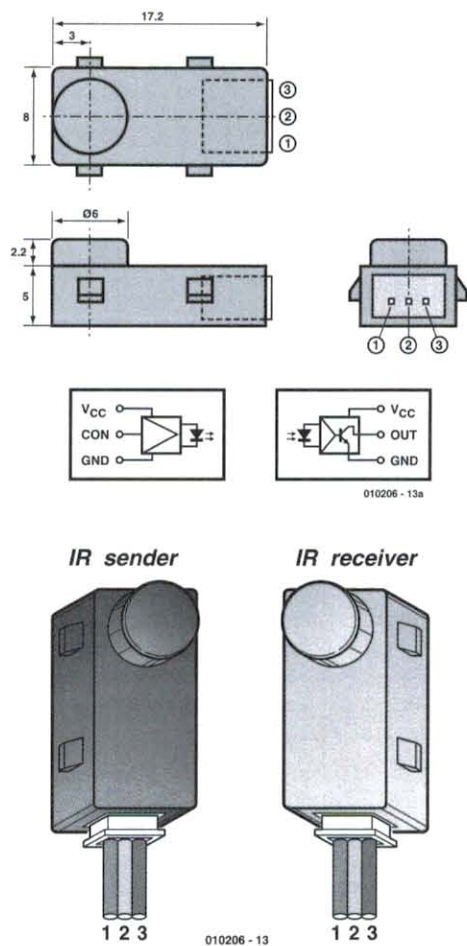
Modulerna, vars benkonfiguration visas i figur 3, är avsedda att användas i papperssensorer, avståndssensorer i reflektionsmod, räknare och registreringssystem eller närhetsdetektorer.



Figur 1. Hastighetsmätning med en ljusbarriär och en reflektor.



Figur 2. Hastighetsmätning med två ljusbarriärers.



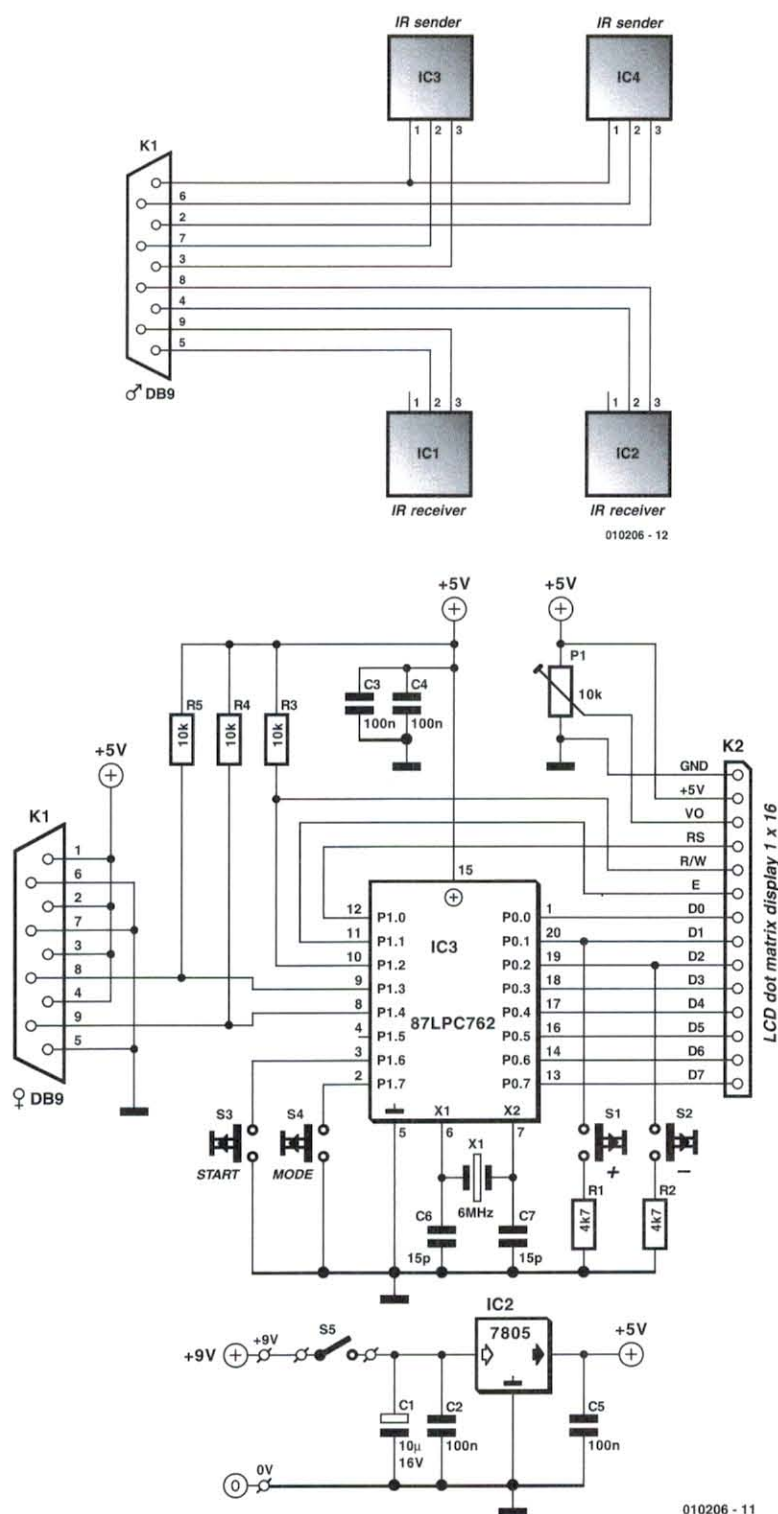
Figur 3. Benkonfiguration hos sändar/mottagarkombinationen.

Enkelheten, det relativt låga priset och pålitligheten hos den krets vi skall gå igenom beror i huvudsak på dessa färdiga moduler från Kodenshi. Anslutningskablar finns tillgängliga som tillbehör. Dessa är dock viktiga eftersom det är mycket svårt att löda fast kopplingstrådar på komponentbenen.

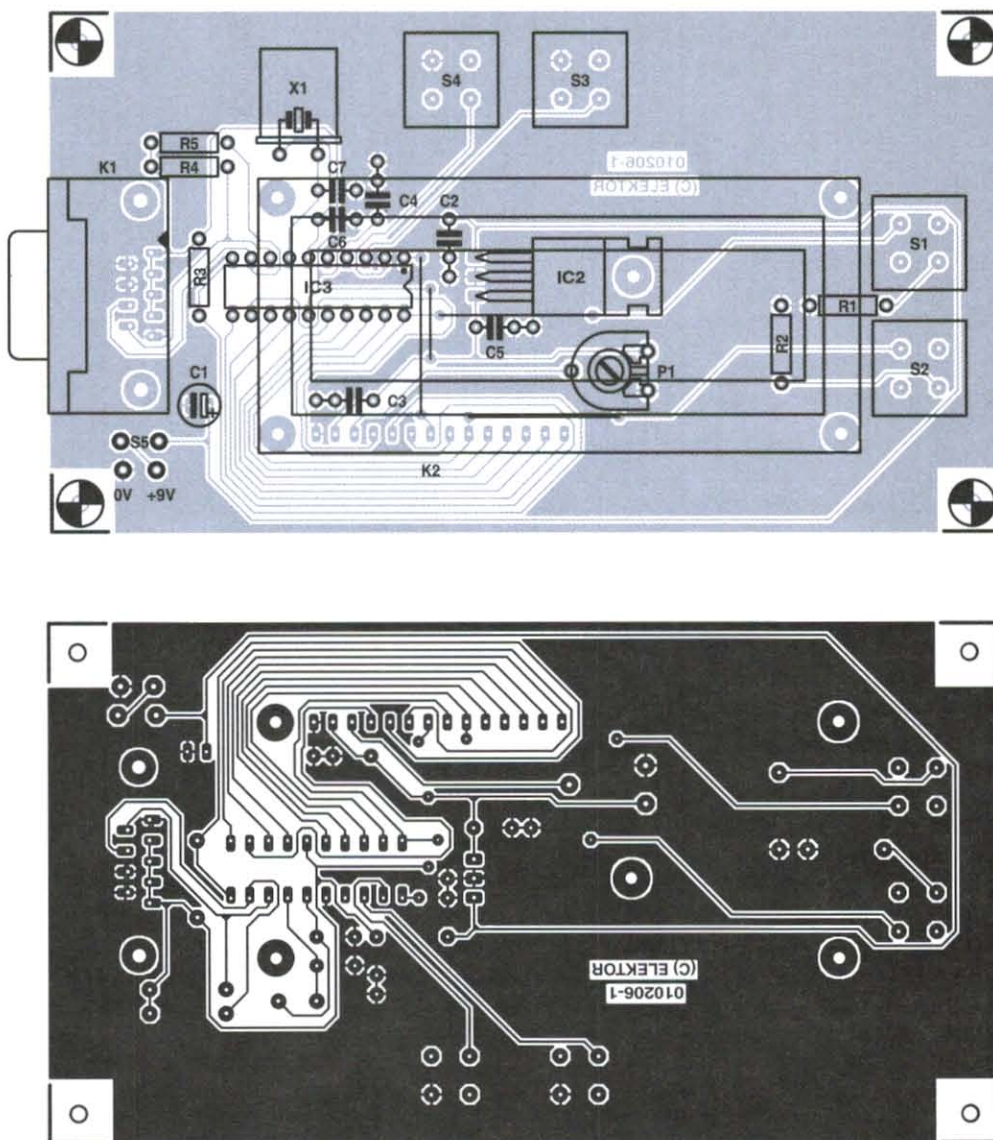
Mikrokontrollern

Tack vare den programmerade mikrokontrollern är kretsschemat ganska rent. Den krets som visas i figurerna 3 och 4 består av två ljusbarriärer, en display för hastighetsvisningen och systeminställningen, fyra tryckknappar och, naturligtvis, en mikrokontroller. För att använda Philips beteckning på den 87LPC762 som används här så är det en 'low power, low price, low pin count microcontroller', dvs en lågeffekts, billig mikrokontroller med få ben. Denna komponent bygger på den väletablerade 8051-arkitekturen. Här klockas den vid 6 MHz (en periodtid på 1 μ s) med tanke på den beräkningsbelastning och höga tidsupplösning som systemet kräver. Huvudegenskaperna för 87LPC762 kan man hitta i databladet:

- 2 kByte ROM
 - 128 Byte RAM
 - 32 Byte användarprogrammerbart EEPROM
 - 2.7 till 6 V matningsspänning
 - Två 16-bitars timer/räknare
 - Integrerad reset
 - Intern oscillator för alternativt bruk
 - 20mA drivkapacitet på alla portpinnar
 - upp till 18 I/O-pinnar
 - 2 analoga komparatorer
 - I²C interface
 - Full duplex UART
 - Seriell in-circuit programmerbar
- Eftersom det inte finns tillräckligt



Figur 4. Kretsschemat för sändar/mottagarkombinationen.



Figur 5. Kretskortslayout för kontroller och display. Kortet finns att köpa färdigt.

KOMPONENTLISTA

Motstånd:

$R_1, R_2 = 4k\Omega$
 $R_3 = 10k\Omega$
 $R_4, R_5 = 100k\Omega$
 $P_1 = 10k\Omega$ trimpot

Kondensatorer:

C1 = 10 μ F 16V
C2-C5 = 100nF
C6,C7 = 15pF

Halvledare:

IC2 = 7805
IC3 = 87LPC762, programmerad,
ordernummer **010206-41**

Övrigt:

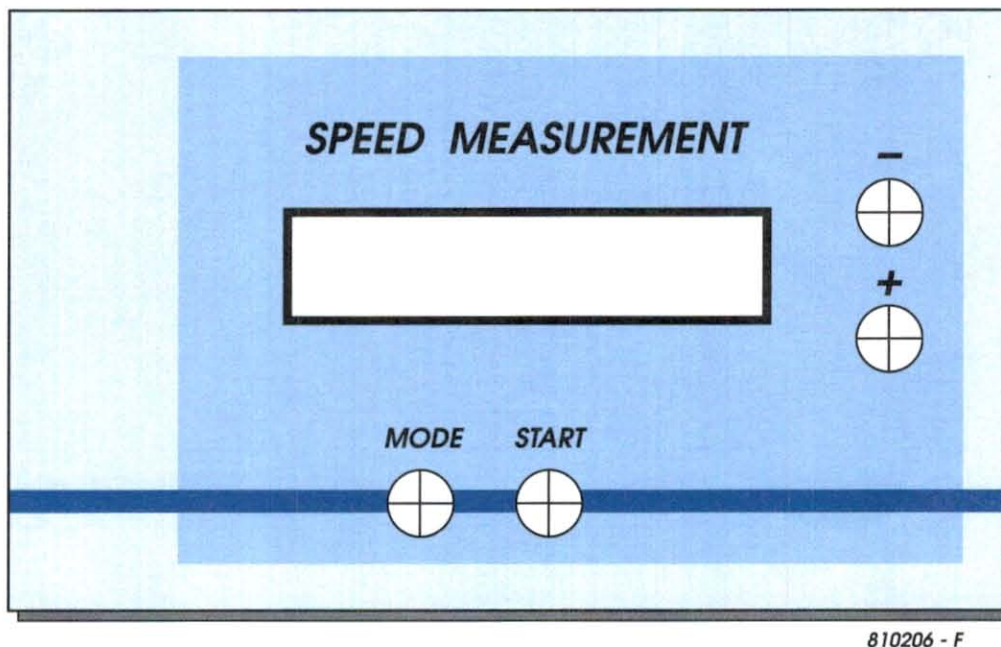
KI = 9-pol d-subkontakt (hona).

vinklade stift, för kretskort
K2 = 14-pol SIL stiftlist
S1-S4 = tryckknapp
S5 = till/fråknapp
X1 = 6MHz kristall
LCD punktmatrisdisplay, 1x16 tecken,
44780-kompatibel med anslutning i
övre vänstra hörnet.
2 kombinationer av IR-sändare PIE-
310 och IR-mottagare PID-310 (Ko-
denshi, Farnell #139-865)
4 modulanslutningskablar, Farnell
#310-0728

Kretskort **010206-I**
Diskett med källkod och hexfiler,
order **010206-II** eller gratis från
www.alltomelektronik.com

med I/O-pinnar tillgängliga för direkt anslutning till tryckknapparna och LCD:n i 8-bitars mod så har vissa av pinnarna fått dubbla funktioner tack vare en intelligent programmering. Tryckknapparnas anslutningar '+' och '-' delar en portlinje med displayens kontrollinjer. För displaykontrollen är dessa pinnar programmerade som push-pullsteg, medan alternativet endast ingång kortvarigt väljs när tryckknapparna pollas (övervakas) för att se om de aktiverats. Motstånden R1 och R2 begränsar kortslutningsströmmen när en knapp trycks in samtidigt som displayen kontrolleras.

Ljusbarriärens mottagarutgångar är kopplade direkt till mikrokontrollerns ingångar. De interna pull-upmotstånden används eftersom ljusbarriären har en utgång med öppen kollektor.



Figur 6. Förslag till frontpanel för Heddic 222 lådan (finns inte att köpa från oss).

Kretsen och själva bygget

Displayen är en alfanumerisk punktmatris med en rad och 16 tecken. Denna version borde inte vara svår att hitta eftersom den betraktas som en slags industristandard, inklusive LCD-kontrollkretsen Hitachi HD44780 och dess kommandoset. Displayen behöver en enkelspänning på +5 V (ingen extra -5 V!). Det är viktigt att kolla benkonfigurationen och anslutningarna samt den interna tilldelningen av RAM-adresserna. Det är bara LCD-moduler med en multiplexhastighet på 1/16 som har den rätta adressordningen 00, 01, 02, 03, 04, 05, 06, 07, 40, 41 till 47. Trots att det skall visas mer information än vad som får plats på en enkelrad, så används en 1-rads display i stället för en 4-raders och användaren uppmanas att trycka på en knapp för att få mer information. Kontrasten justeras med trimpotentiometer P1. Vrid P1 över hela dess område om du inte ser något på skärmen när du startar kretsen första gången och du vet med dig att det inte finns ett fel någon annanstans.

Kretsen matas, som vanligt, via en fast spänningsregulator av typ 7805.

Med tanke på enkelhet och pris så är sändaren och mottagaren anslutna till process-elektroniken via en 9-polig d-subkontakt. Kopplingstrådarna för ljusbarriären skall anslutas enligt figur 3.

I denna applikation används sändarens kontrollgångar inte, de är antingen inte anslutna eller dragna till jord.

Kretsen kan byggas mycket lätt med hjälp av det kretskort som visas i figur 5. Displayen,

tryckknapparna samt till/från-switchen monteras på kortets undersida. Mikrokontrollern monteras i en hållare.

Den föreslagna lådan för detta projekt är en Heddic 222 som, då den är genomskinlig, inte behöver någon öppning för LCD-skärmen. Ett förslag till frontpanel visas i figur 6 och designades med tanke på 222-lådan. Panelen ger instrumentet ett proffsigt utseende.

Praktisk användning

För att kontrollern skall reagera omedelbart till förändringar på ljusbarriärens utgångar så är mätrutinen interruptdriven. Följaktligen pollas inte portpinnarna P1.3 och P1.4 konstant av mjukvaran. I stället används logik inuti mikrokontrollern för att kontrollera tidsmätningen. Fördelen med detta arrangemang är att, förutom den ögonblickliga reaktionen på händelser, mjukvaran har tillräckligt med tid över för andra uppgifter, som att driva displayen och scanna tryckknapparna. Detta är inte möjligt bara så där utan vidare och med vilket ingångsben som helst. P1.3 och P1.4 är 'speciellt preparerade' för att hantera dessa exakta uppgifter. Vid en fallande pulsflykt stannar det interna programmet omedelbart och hoppar till en speciell rutin som skri-

vits just med tanke på denna uppgift.

Alla hastighetsmätningar börjar med att den första ljusstrålen bryts. 16-bitarsräknaren inuti mikrokontrollern börjar då omedelbart att räkna med en hastighet på 1 μ s. Överspill som uppstår adderas i ett 8-bitars register, vilket gör att en maximal period på 16.777215 sekunder kan mätas. Föremålets hastighet, v , beräknas sedan av mikrokontrollern med hjälp av den enkla ekvationen

$$v = d / t$$

där d är avståndet mellan ljusbarriärerna och t är timerens status.

I grundinställningen antar mjukvaran att avståndet mellan barriärerna är 0,1 m, displayen visar m/s och mätmetoden är 'kontinuerligt'. För att förenkla beräkningarna så har upplösningen begränsats till två decimaler, vilket borde vara fullt tillräckligt i de allra flesta fall. Om du vill ha bättre noggrannhet tar du fram din miniräknare samt använder det visade räknarvärdet och avståndet mellan barriärerna. Ta dock alltid fördröjningen på 0,5 ms med i beräkningen. Notera dock att denna fördröjning är ungefär lika stor i båda mottagarna och borde därför inte vara något problem.

Mer viktigt än att försöka krama ut

siffror bakom decimalkommat är att se till att sändaren och mottagaren är korrekt inriktade och fastsatta, samtidigt som det är lätt att justera avståndet mellan dem för de hastighetsmätningar som du vill utföra. När barriärerna är installerade, hopkopplade och mjukvaran switchats till så tar mjukvaran över och guidar dig genom en meny där du kan välja mellan flera olika alternativ.

MODE-knapp

Tryck på denna knapp för att gå genom de fyra menyposterna nedan

TEST – DISTANCE – MODE – SPEED – TEST

TEST visar tillståndet hos mottagarna i ljusbarriären. Meddelandet **OK** betyder 'hög nivå uppmätt' och visar att ljusstrålen från sändaren har plockats upp. **Low** visar att strålen antingen är bruten eller att den inte är korrekt riktad mot mottagaren. Innan några mätningar påbörjas skall båda utgångarna visa OK. Efter en reset eller efter det att kretsen har startats på nytt visas alltid denna meny först.

Det skall noteras att en icke ansluten ljusbarriärmottagare alltid kommer att visa OK, helt enkelt på grund av de interna pull-upmotstånden på kontrolleringsångarna. Ta därför till vana att alltid göra en 'säkerhetskontroll' genom att manuellt bryta strålen medan du tittar på displayen. En annan felkälla är ljuskällans kägelform som kan nå den andra mottagaren om barriärerna har placerats för nära varandra.

När **DISTANCE** visas anges värdet som skall behandlas av mikrokontrollern i centimeter. Utgångsvärdet är 10 cm. Och naturligtvis måste du se till att detta värde är detsamma som avståndet mellan ljusbarriärerna.

MODE låter dig välja mätalternativ:

SINGLE utför bara en enda mätning. När barriärerna triggas

igen utförs ingen ytterligare mätning.

CONTINUOUS innebär att resultatet uppdateras vid varje ny mätning. Det aktuella värdet skrivs över med det nya värdet.

SPEED visar den uppmätta hastigheten hos föremålet. Du kan välja mellan tre sätt: Meter/sekund, kilometer/timme eller sekunder. Den maximala tiden är 16.777215 sekunder. Om du överskrider denna tid visar displayen ERROR.

START-knapp

Denna tryckknapp läses endast i SINGLE-mod. Den startar en mätning när någon av ljusbarriärerna bryts. Displayen visar då READY. Om en barriär misslyckas med att detektera en signal hoppar mjukvaran till TEST och där kan du undersöka vad som är fel. När problemet är löst, och du trycker på START igen, får du fram menyn SPEED eller DISTANCE.

+ och – knapparna

På traditionellt sätt används dessa knappar för att välja olika parametrar i de olika menyerna. I menyn DISTANCE, till exempel, kan du använda knapparna för att ställa in avstånd mellan 1 cm och 255 cm.

I menyn MODE har de båda knapparna samma funktion, de växlar mellan SINGLE och CONTINUOUS. Och slutligen, i menyn SPEED kan

Några användbara adresser

Punktmatrisdisplay:

www.datamodul.de/displaytechnik/lcd-alpha/bt_11608.htm
<http://electronicsassembly.de/>
www.schukat.com (Display Fa. Sharp)

Mikrokontroller:

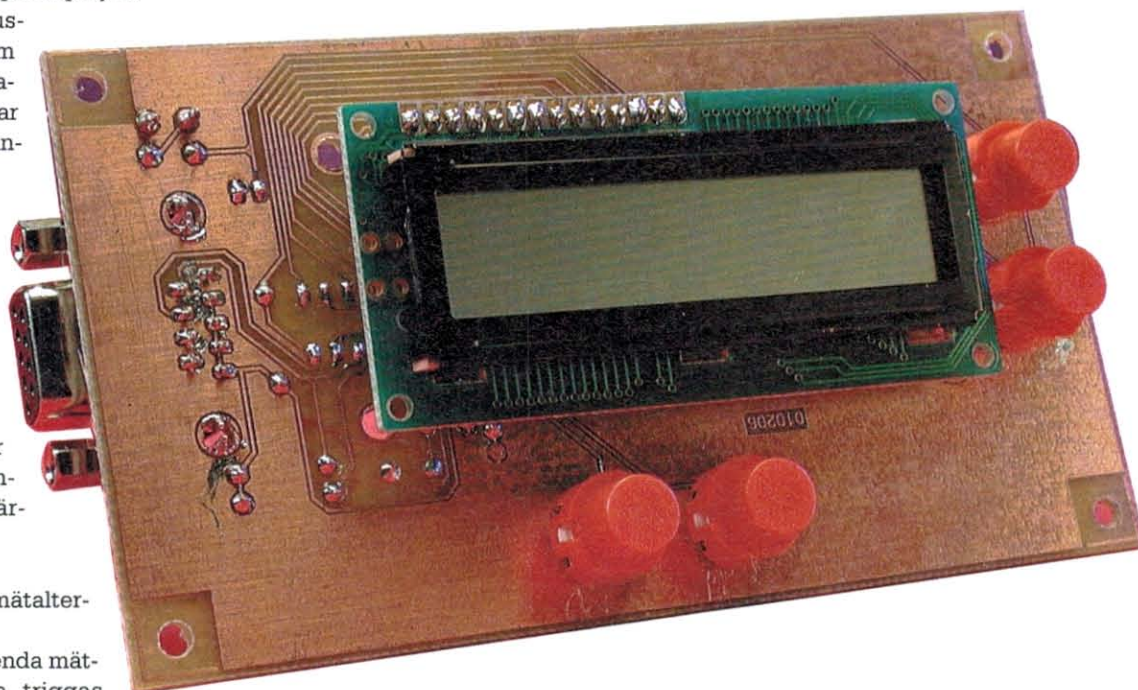
www-us.semiconductors.philips.com/mcu/

Datablad för ljusbarriär:

www.kodenshi.com/pdfs/g-l.pdf

du använda dessa knappar till att välja mellan en display som visar m/s, km/h eller sekunder (det senare för att underlätta för dem som använder miniräknare).

(010206-1)



Fjärrstyrning med en mobiltelefon

Del 2: Programmering och användning

SMS-chipet kollar inkommande text efter lösenord, switchar utgångar interaktivt eller vid en förinställd tid, styr en LCD och skickar statusinformation tillbaka till vilken mobiltelefon som helst.

I den första artikeln i denna serie tittade vi på kretsdesignen och hårdvaran för detta projekt, och nu vänder vi oss till kretskortslayouten, mjukvaran och strukturen för textmeddelanden.

En titt på komponentplaceringsritningen i figur 1 visar att det inte behövs några speciella kunskaper i lödningsteknik för att bestycka kretskortet. Det finns inga trådbygglar och alla IC-kretsar (utom spänningsregulatorn) är monterade i hållare. Alla externa anslutningar sker via stiftlistor eller kontakter som är monterade utefter kortets kanter. LED-staplarna D1 och D2 kan ersättas med vanliga lysdioder i grupper om åtta, men om du använder högintensiva typer måste du använda ett motståndsnät på 1 k Ω eller 1,5 k Ω för R1 och R2. Under alla förhållanden är det nödvändigt att använda den ATC-buffert som specificeras i komponentlistan för IC3 och IC4.

Spänningsregulatorn IC8 behöver ett kylelement som monteras utefter kretskortet. Se till att lägga en isoleringsbricka mellan kylelementet och kretskortet så att det inte blir någon kortslutning.

Till/frånkopplaren S4 kan monteras direkt på kretskortet, eller på frontpanelen om du monterar in kortet i en låda.

Tre skruvar med distanshylsor används för att fästa LCD-modulen på interfacekortet för SMS ExBo. Det fjärde monteringshållet passar inte med hålet i kretskortet, men en skruv och en distanshylsa skall monteras på LCD:n för att ge ytterligare stöd. Låt den undre delen av hylsan vara öppen. Isolering brickor används

Egenskaper hos SMS-chipet

SMS Chip* är implementerat i en AT89LS8252 från Atmel (ett programmerbart 8051-derivat)

- 8 kByte On-Chip Flash-programmerbart minne
- 2 kByte On-Chip Flash-dataminne
- 16 digitala portpinnar (Port P1 och Port P2), individuellt programmerbara som ingångar eller som utgångar
- Serieinterface 1 för kommunikation med mobiltelefonen eller nedladdning av konfigurationsdata
- Serieinterface 2 UART implementerat i mjukvara (9600 baud) för kommunikation med externt mikroprocessorsystem (PLC, PC)
- Real Time Clock (RTC) i mjukvara utan batteriback-up.
- Valfri extern Real Time Clock (RTC) i hårdvara med batteriback-up (IC6).
- Kontroll för alfanumerisk LC-Display för att visa textmeddelanden.
- Kontakt för externt kretsinterface (Port P0).
- Kapsling DIP-40 eller PLCC-44

* SMS-Chipet finns enbart att köpa från:

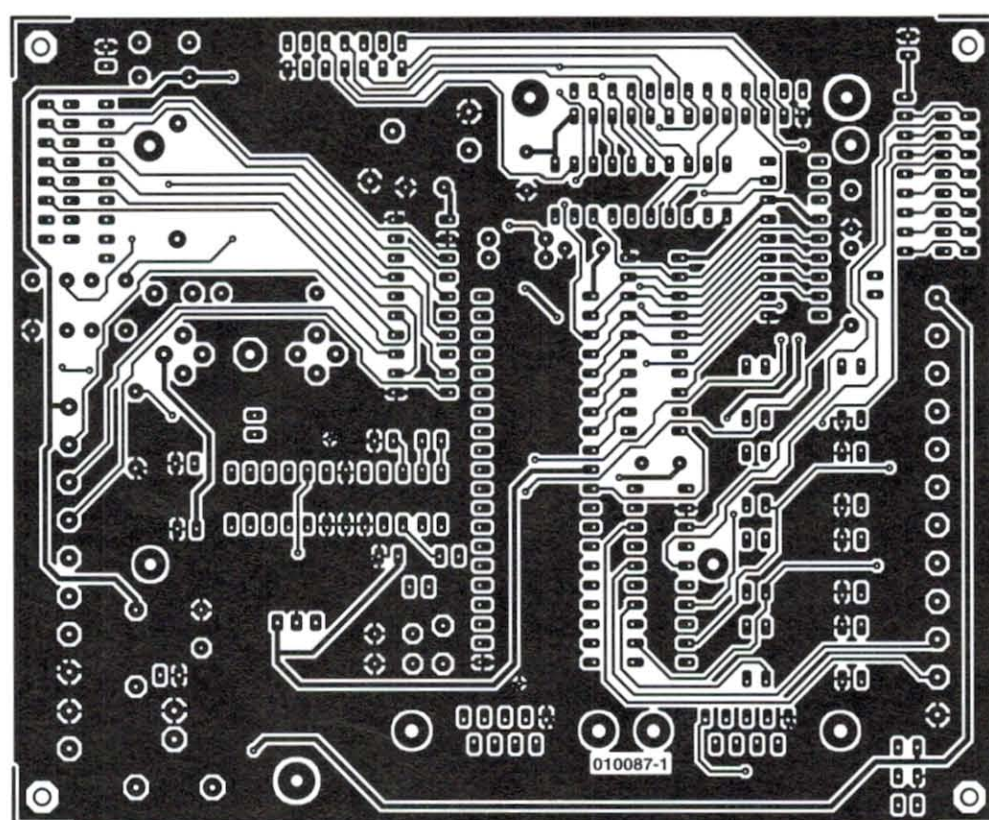
J. Engelmann & U. Schrader
Im Schmiedehofe 14
D-31035 Despetal-Barfelde
Tyskland.
Tel. (+49) 5182 903520
Fax (+49) 5182 903530
Web: www.engelmann-schrader.de
Email: info@engelmann-schrader.de

vänds under de tre skruvhuvudena för att förhindra eventuell kortslutning mellan kretskortsbanorna.

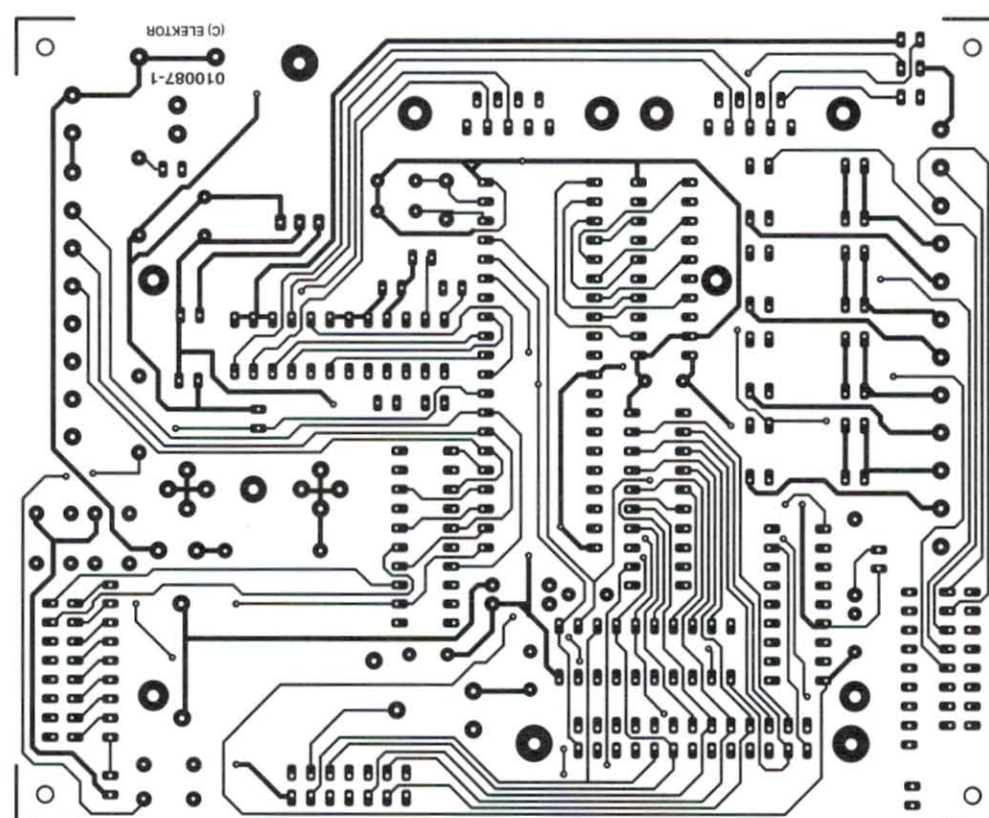
Datakabeln

Anslutningen mellan adapterkortet

och mobiltelefonen kan göras med en färdig datakabel för Siemenstelefoner ur '35-serien. Dessa kablar används normalt för att ansluta mobilen till serieporten på en PC. Det finns dock en stor nackdel med att använda en sådan kabel i vår applikation eftersom

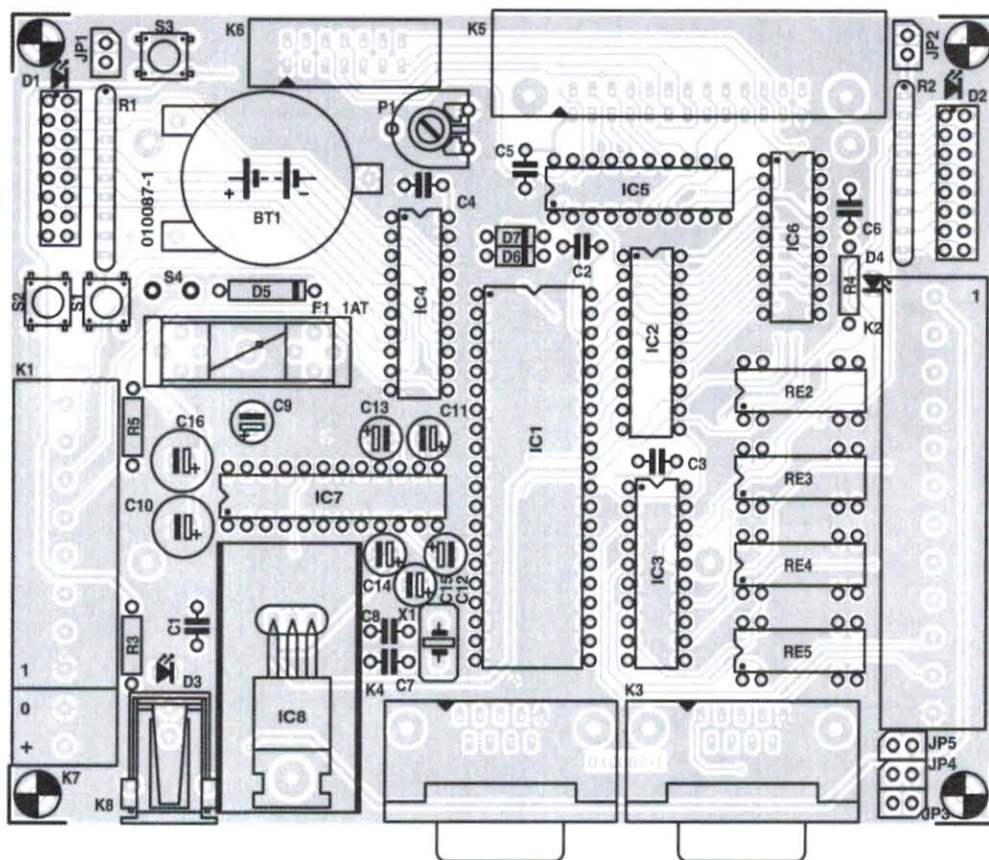


Komponentsida



Lödsida

Figur 1a. Kretskortslayout och ...



Figur 1b. ... komponentplaceringsritning för SMS ExBo.

den inte tillåter att mobiltelefonen laddas från SMS ExBo-kortet. Kabeln kan inte lämna en laddningsström för mobilens batteri. Om du använder denna kabel blir det nödvändigt att periodvis koppla bort telefonen från SMS ExBo och ladda batteriet från telefonens laddare. Om du använder detta alternativ är det viktigt att båda kortslutningsbyglarna J4 och J5 tas bort och att bygeln J3 monteras (se tabell 2 i den första artikeln). Det bästa alternativet är att göra en mindre modifiering av ledningarna i datakabeln. Ta först isär höljet i datakabelns telefonända (se figur 2) och löd fast en extra tråd (samma längd som datakabeln) till stift 3. Dra nu ut denna tråd genom öppningen i höljet och sätt ihop kontakten igen. Montera ett 1 Ω , 1 W motstånd 'in-line' i denna tråd och anslut kabeln fria ändor till +5 V på SMS ExBo (stift 1 på kontakten K2 är lämpligt). Laddningsströmmen kan vara upp till 500 mA så det är viktigt att inte snåla in på den nätadapter som specificeras i komponentlistan. Montera kortslutningsbygeln J3 men montera inte byglarna J4 och J5, dessa är avsedda för framtida användning med andra typer av mobiltelefoner.

Funktionen hos SMS Chip

När kretskortet är färdigbyggt och du har

tillverkat datakabeln kan vi nu rikta vår uppmärksamhet till SMS-chipet och hur SMX ExBo kontrolleras. SMS-chipet måste initialiseras innan det kan känna igen de kommandon som sänds i SMS-texten.

Initialisering av SMS-chipet

SMS-chipet måste konfigureras med information såsom mobiltelefonnumret, förloadade textmeddelanden osv. innan det kan användas. Konfigurationsdata kan anges med hjälp av ett vanligt textredigeringsprogram. Se den tillhörande textrutin för ett exempel på konfigurationsdata.

Konfigurationsdata lagras i ASCII-format i filen SMSCHIP.CFG och det program som används för att sända dessa data till SMS ExBo är DOS-programmet SMSCONFIG.EXE. För att ladda konfigurationsdata är det först nödvändigt att ta bort bygel J3 (byglarna J4 och J5 skall inte vara monterade) och sedan koppla in en vanlig standard seriell (hona-tillhona) nollmodemkabel från COM1 på PC-n till K3 på SMS ExBo. Tryck

på resetknappen S3 och 'mobile ready' indikatorn (D4), etiketterad 'power' i kretsschemat, blinkar för att visa att SMS-chipet väntar under 10 sekunder på att konfigurationsdata skall sändas. Ett dubbelklick på SMSCONFIG.EXE filen öppnar automatiskt ett DOS-fönster och visar överföringen av konfigurationsdata. Vissa av er måste kanske använda COM2-porten. I detta fall skall du använda kommandot

```
SMSCONFIG.EXE /2
```

när du anropar programmet från DOS-fönstret. När överföringen är klar stänger du DOS-fönstret och trycker på resetknappen S3. D4 kommer återigen att blinka under 10 sekunder, (men eftersom konfigurationen redan skett så kan du strunta i D4) och därefter kommer SMS-chipet att leta efter en mobiltelefon som är ansluten till K3. Koppla bort den nollmodemkabel som är ansluten från PC-n till SMS ExBo och anslut mobiltelefonen till K3 med hjälp av datakabeln. Montera nu till-

KOMPONENTLISTA

Motstånd:

R1, R2 = 8-pol 330Ω motståndsnät
R3 = 1kΩ
R4 = 1kΩ
R5 = 4kΩ
P1 = 10kΩ trimpot

Kondensatorer:

C1, C2, C4, C5, C6 = 100nF (5mm raster)
C3 = monteras ej
C7, C8 = 27pF
C9 = 10μF 63V radial
C10 = 100μF 25V radial
C11-C15 = 1μF 16V radial
C16 = 100μF 10V radial

Halvledare:

D1, D2 = LED, 3mm (8 st) eller stapel *
D3, D4 = LED, grön, dagsljus
D5 = 1N4002
D6 = BAT48
D7 = 1N4148
IC1 = AT89(L)S8252-24PC i DIP40-hölje. Programmerad, tillgänglig från Engelmänn&Schrader
IC2 = 74HCT573
IC3, IC4 = 74ACT240*
IC5 = GAL16V8, programmerad, vårt ordernummer **010087-31**
IC6 = RTC72421
IC7 = MAX207 or ADM207EAN
IC8 = 7805 med ICK35SA kylning *

Övrigt:

BT1 = CR2032 Lithium knappcell

med kretskortshållare
F1 = säkring, 1A, trög, med kretskortshållare
JP1-JP5 = bygel
K1 = 8-pol kopplingsplint för kretskort, raster 5 mm
K2 = 12-pol kopplingsplint för kretskort
K3, K4 = 9-pol d-sub (hane), för kretskort, vinklade stift *
K5 = 26-pol boxheader, vinklade stift
K6 = 14-pol boxheader
K7 = 2-pol kopplingsplint för kretskort, raster 5 mm
K8 = jack för nätadapter, för kretskortsmontering
RE1 = monteras ej
RE2-RE5 = 5V relä för kretskort, 1 slutande eller 1 växlande kontakt med flybackdiod (t.ex., Meder 1A72-12D 5V eller Siemens V23100-V4305-C010 eller Conrad Electronics #504580) *
S1, S2, S3 = tryckknapp med 1 slutande kontakt
S4 = kontakt, till/från *
X1 = 11.0592MHz kristall
LC Display = alfanumeriskdisplay med HD44780 kontrollern (eller kompatibel), t.ex. 4 rader med 20 tecken
Kretskort **010087-1**
Diskett med mjukvara **010087-11**
Batterieliminatör (nätadapter) 9V / 1A

* Ses text

baka kortslutningsbygeln J3.

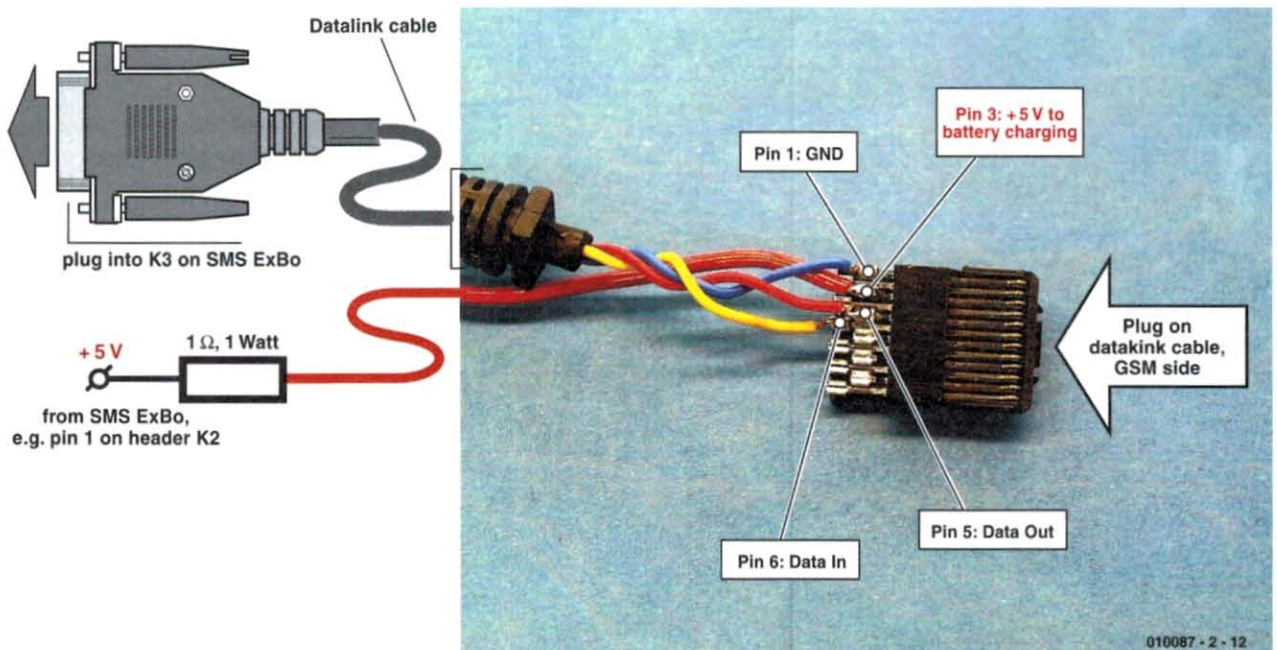
En kort beskrivning av konfigurationsdatafilen tillsammans med giltiga parametrar visas i en textruta på annat ställe i denna artikel, men för mer detaljerad information referera vi till handboken för SMS-chipet.

Instruktionssettet för SMS-chipet

När SMS-chipet har konfigurerats är det klart att kunna ta emot kommandon. Det finns sammanlagt 12 kommandon som SMS-chipet känner igen och som enkelt kan anges med hjälp av SMS-textning (se tabell 1). Alla kommandona har samma struktur. Först kommer lösenordet, sedan kommandot och slutligen avslutningen av meddelandet. Detaljerade beskrivningar angående alla kommandon finns i handboken för SMS-chipet, som kan hämtas gratis från vår websida. Ni som besöker denna websida kommer också att hitta ytterligare ett program som är associerat med detta projekt och som är skrivet i C51. SMS-1.c och dess tillhörande hexfil SMS-1.hex. Med detta program kan alla som har byggt 537 'Lite' datorn, som vi hade som projekt i nummer 1 och 2/2000, använda denna som extern dator på fjärrsidan. I denna konfiguration kommer 537 'Lite' datorn att ta emot SMS-text från SMS ExBo och skicka tillbaka sina egna textmeddelanden samma väg.

Så här fungerar det

När ett meddelande tas emot läser SMS-chipet all SMS-text och raderar sedan automatiskt meddelandet från telefonens minne. Om ett vanligt textmeddelande (utan



Figur 2. Modifiering av en inköpt datakabel för att kunna ladda mobiltelefonens batteri. Se texten.

Gratis mjukvara

För att kunna ladda ner dessa filer går du till vår websida vid www.alltomelektronik.com

SMSCHIPCFG	Konfigurationsdata
SMSCONFIG.EXE	Laddningsprogram för smschip.cfg
SMS-I.c	Kontrollprogram i C för 537 'Lite' datorn
SMS-I.hex	Hexfil för ovanstående program sms-I.c

lösenord eller konfigurationskommando) skickas till telefonen kommer den att försöka avkoda detta, men eftersom den inte hittar något av intresse raderas meddelandet från mobiltelefonen.

En viktig egenskap hos detta system är responstiden för meddelanden via SMS-nätet. Tiden det tar att sända ett meddelande från en mobiltelefon och till dess att det tas emot

av en annan mobiltelefon kan variera från några få sekunder ända upp till en timma. Det är därför viktigt att komma ihåg att detta system inte är lämpligt att använda för att styra eller övervaka tidskritiska processer! Lämpliga applikationer kan vara, till exempel, att styra värmesystem eller övervaka frysanläggningar.

Under uppsättning och testning behöver du skicka iväg en hel del meddelanden till systemet bara för att kunna bekräfta att enheten fungerar korrekt. Ett bra alternativ till att använda en mobiltelefon är att skicka dessa meddelanden genom en Internetansluten PC. Det finns flera webbsidor som låter dig skicka textmeddelanden till mobiltelefoner gratis (prova www.passagen.se till exempel). Detta sparar inte bara pengar utan hjälper dig också att undvika den medicinska åkomma som kallas 'SMS-tummar'. Dessa webbsidor brukar lägga till reklam i början eller efter ditt meddelande, men SMS-chipet kommer högakttningsfullt att strunta i dessa. Det avkodar endast den text som följer efter 'message start' markören (lösenordet) och slutar avkoda när 'message end' symbolen (bakåtparentes) tas emot. All text utanför dessa markörer kommer att ignoreras. Det kan uppstå problem om reklammeddelandet lagts in mitt i textmeddelandet, men detta är inte speciellt troligt och vi har ännu inte stött på någon sida som gör detta. Om detta skulle ske så ser SMS-chipet att meddelandet är korrupt och kommer att ignorera det helt och hållet. Ett giltigt meddelande måste ha tagits emot innan någon av utgångarna switchas. Om du inte gillar reklam så måste du hitta en websida som skickar gratis SMS-meddelanden utan sådan. Internet är dock ingen välgörenhetsinsättning så det kan vara svårt. Det finns sajter på Nätet som erbjuder email till SMS-gateway. Som namnet antyder tillhandahåller dessa en länk mellan textnätet för GSM och Internet så att email kan sändas och tas emot som textmeddelanden på en GSM-telefon. Ytterligare information om detta alternativ hittar du i handboken.

Tabell I. SMS-chipets kommandoinstruktionsset.

(" <lösenord> "		
SET	<stift>	// Sätt specificerat stift till 1
eller		
SET	<identifierare>	// Sätt specificerat stift till 1
RESET	<stift>	// Reset specificerat stift till 0
eller		
RESET	<identifierare>	// Reset specificerat stift till 0
OFF	<stift>	// Sätt specificerat stift till 1
eller		
OFF	<identifierare>	// Sätt specificerat stift till 1
ON	<stift>	// Reset specificerat stift till 0
eller		
ON	<identifierare>	// Reset specificerat stift till 0
PULSE	<stift> <längd>	// Skicka en puls på det specificerade stiftet
eller		
PULSE	<identifierare> <duration>	// Skicka en puls på det specificerade stiftet
OUTPUT	<adress> <värden>	// 8-bitars utgång från XDATA (8 bitars adress)
INPUT	<adress> <räkna>	// 8-bitars ingång från XDATA (8 bitars adress)
WRITE	"<text>"	// Skicka ut till 2:a aerieinterfacet
DISPLAY CLS		// Rensa LC-Display
eller		
DISPLAY SCROLL	"<text>"	// Lägg in text i sista displayraden och scrolla
eller		
DISPLAY	<kolum> <rad> "<text>"	// Lägg in text för rad och kolumninformation
EVENT	<tid> DELETE	// radera tidshändelse
eller		
EVENT	<tid> <adress> <räkna> SINGLE	// Tidshändelse sker endast en gång
eller		
EVENT	<tid> <adress> <räkna> EVERYDAY	// Tidshändelse sker dagligen
TIME	<tid>	// Aktuell tid
REPORT NO		// Inget att rapportera (grundvärde)
eller		
REPORT YES		// rapportera
eller		
REPORT ERROR		// Rapportera endast vid fel
)		

(010087-II)

Exempel på konfigurationsfil

```
-CONFIG
Time :1200
Name :Enterprise
Password :Scotty
Masterno :441794711081
Pin 0 I/O :I
Pin 0 Name :Warp1
Pin 0 Telefon :441781234567
Pin 0 Text :Pressure too high!
Pin 1 I/O :I
Pin 1 Name :ALARM
Pin 1 Telefon :4512345876
Pin 1 Text :Romulan on board!
...
Pin 7 I/O :I
Pin 7 Name :Pin7

Pin 7 Telefon :441704711081
Pin 7 Text :Report pin 7!
...
Pin 8 I/O :O
Pin 8 Name :Pin8
Pin 8 Telefon :440000000000
Pin 8 Text : (Pin8)
...
Pin 15 I/O :O
Pin 15 Name :Photon24
Pin 15 Telefon :440000000000
Pin 15 Text : (Pin15)
Event 1 Time :1200
Event 1 Typ :E
Event 1 Start :00
Event 1 Count :05

Event 2 Time :1205
Event 2 Typ :E
Event 2 Start :10
Event 2 Count :12
Event 3 Time :1210
...
Event 8 Time :0000
Event 8 Typ :O
Event 8 Start :00
Event 8 Count :0
LCD Lines :04
LCD Char/Lines :20
LCD Addr 0 :00
LCD Addr 1 :64
LCD Addr 2 :20
LCD Addr 3 :84
```

Dataheader

-CONFIG

Nyckelord som visar början av konfigurationsdata.

Time

Verklig tid för mjukvaru-RTC (Real Time Clock) (RTC) i SMS-chipet. Notera, inget separerande kolon skall anges mellan timmar och minuter.

Name

Här kan du ange ett identifierande namn till SMS-chipet. Det kommer att användas detta namn varje gång det sänder SMS-text.

Password

Varje SMS-meddelande som skickas till fjärrmobilen måste användas det lösenord som definieras här. Meddelande som tas emot utan detta lösenord kommer att ignoreras.

Masterno

Detta är det huvudtelefonnummer som SMS-chipet kommer att skicka sina meddelanden till.

Digitala I/O-portpinnar

Pin x I/O

Detta definierar pinne x som en ingång eller utgång: I = Ingång, O = utgång. När pinnen är konfigurerad som en ingång, kommer SMS-chipet att automatiskt skicka ut ett meddelande när ingångsnivån växlar från hög till låg (fallande flank). Den skickar bara ett annat meddelande när ingångsnivån går tillbaka till hög.

Pin x Name

Här kan du tilldela en pinne ett namn så att du i ett textmeddelande kan referera till pinnens nummer eller namn, vilket som är mest meningsfullt.

Pin x Telefon

Det mobiltelefonnummer som anges här kommer att ringas upp och skickas meddelande till när en fallande flank detekteras på pinne x. Om inget nummer anges kommer huvudtelefonen (Masterno) att ringas upp.

Pin x Text

Det meddelandet (max 32 tecken) som kommer att skickas när en förändring på pinne x detekteras.

"Pin x Telefon" och "Pin x Text" är endast giltigt om pinne x är definierad som en ingång

Interna och tidsbundna händelser

SMS-chipet kan skicka systeminformation automatiskt vid förutbestämda tider till mobiltelefonnumret (Masterno). Totalt åtta händelser finns tillgängliga (x=1 till 8)

Event x Time

Den tid som SMS-meddelandet kommer att skickas.

Event x Typ

Tre olika typer av händelser är möjliga:

O (Från): Motsvarande händelse avaktiveras.

S (Single): Händelsen kommer att aktiveras endast vid den angivna tidpunkten.

E (Everyday): Händelsen kommer att aktiveras dagligen vid den angivna tidpunkten.

Event x Start

Startadressen (decimal) i ett externt dataminne (XDATA-område) som kommer att inkluderas och skickas ut med SMS-texten.

Event x Count

Här definieras hur mycket data samt startadressen för data som skall sändas. I det visade exemplet skickar händelse 2 ut 12 data-byte i ett SMS-meddelande från den externa minnesadress som börjar vid adress 10 klockan 12:05 varje dag.

Användning av externa data täcks inte i denna artikel, men systembussen för ExBo finns tillgänglig vid K5 för anslutning av externt minne (t.ex. dubbelport-RAM) för att lagra dessa data.

LCD-operation

LCD Lines

Antal rader på LCD-n.

LCD Char/Lines

Tecken per rad

LCD Addr 0 to 3

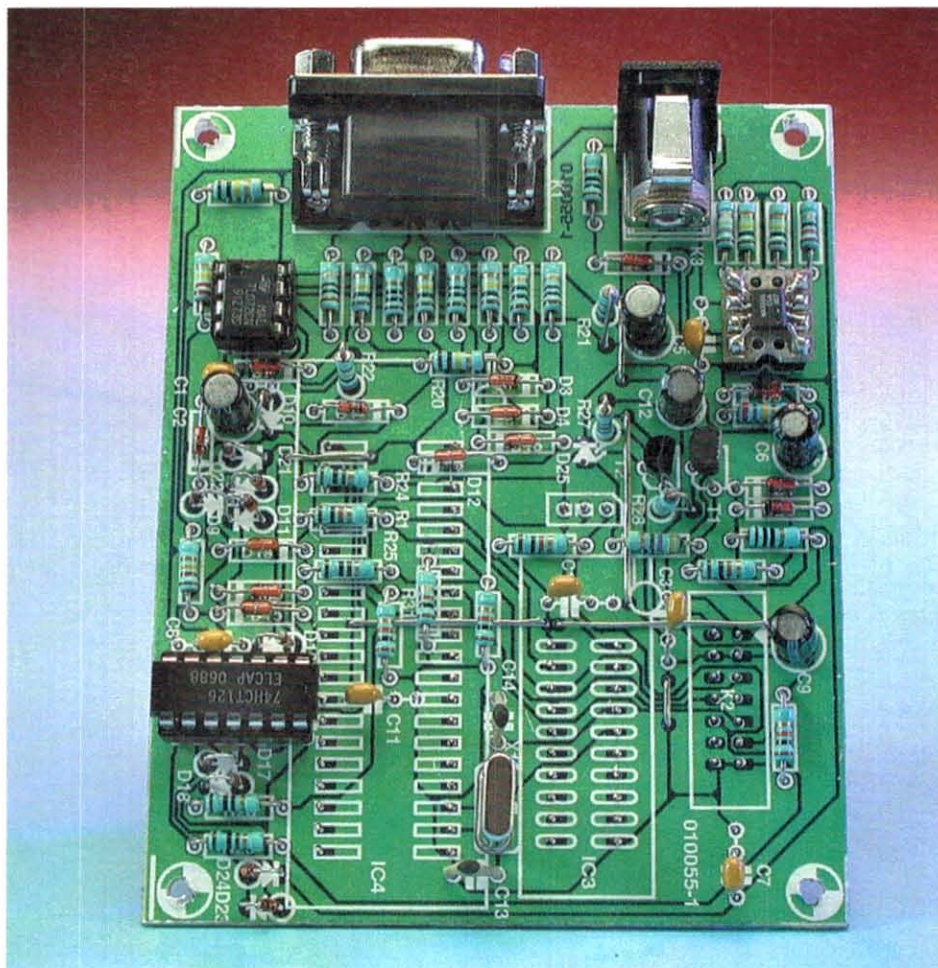
Adressen för första tecknet i varje rad på displayen. Värdena ges i motsvarande datablad.

Enkel AVR-programmerare

för (nästan) alla AVR-mikrokontroller

Design av H.-J. Hanft

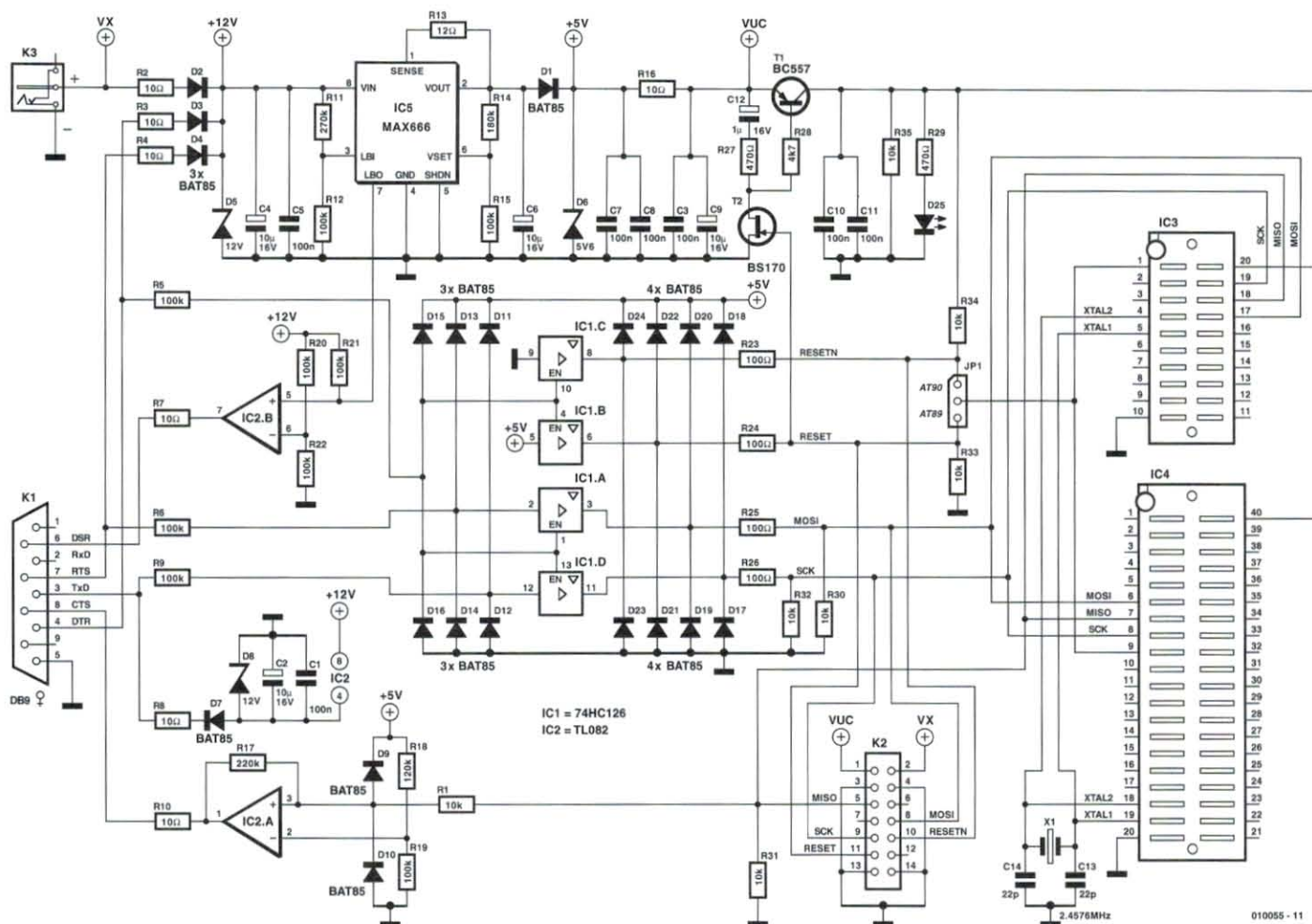
Tack vare deras höga prestanda, Flash programminne, integrerade hårdvarufunktioner och låga effektförbrukning håller Atmels AVR mikrokontroller på att bli allt populärare, även inom det semi-professionella området.



Mikroprocessorer ut AVR-serien baseras på Flash programminne och kan därför programmeras på ett enkelt sätt med ett SPI-interface. Möjligheten att omprogrammera programminnet (upp till 1000 skrivcykler) gör dessa komponenter mycket attraktiva inom det semiprofessionella området.

De dokument och verktyg som du behöver för att programmera dessa mikrokontroller, som assembler och avbuggare, kan du hitta gratis på Internet på adressen www.atmel.com. Dessa verktyg är för Windows (95/98/NT). Du hittar inte några utvecklingsverktyg som körs under MS-DOS här, vilket är ett bra skäl till att utveckla en programmerare och den associerade mjukvaran för körning under Windows.

Denna enkla programmerare är en extremt billig utgångspunkt för utveckling av applikationer som använder AVR mikrokontroller. Vid designen av programmeraren lades stor vikt vid att det skulle vara en billig, enkel och robust konstruktion som klarar av svåra omgivningsförhållanden, som elektrostatiska urladdningar, kortslutningar och inducerade störningar.



Figur 1. I stället för en interfaceomvandlare så ger två opampar och en logisk grind nivåomvandlingen mellan PC-n och mikrokontrollern.

Programmeraren ansluts till datorn via ett RS232-interface, med hjälp av en 1:1 RS232-kabel (inte en noll-modemkabel!). Om du bara vill programmera Flashminnet behövs det inte någon extern matningsspänning för att använda utrustningen. Den

matningsström (några få milliamp) som behövs för att programmera en mikrokontroller kan tas direkt från datorns serieinterface. Programmeraren kan också användas som en in-circuit programmerare för programmering av en mikrokon-

troller som redan sitter i en krets. I detta fall kan det, under vissa förhållanden, vara nödvändigt med en extern strömkälla. Den universella programmeringsmjukvaran är implementerad som en konsolapplikation för Windows 95, 98, NT, Me, 2000 och XP.

Egenskaper

De speciella egenskaperna hos programmeraren är:

- enkel, ekonomisk och robust konstruktion
- universellt SPI-interface, också lämpligt för andra applikationer
- programmeringshastighet upp till 256 baud (beror på PC)
- kan också användas för in-circuit programmering
- ingen extern strömkälla behövs
- 'låg spänning' indikator varnar för otillräcklig matningsspänning
- alla SPI-utgångar är kortslutningssäkra och har överspänningsskydd
- max stabiliserad utgångsström med externa strömkälla: 30 mA
- alla SPI-ingångar och utgångar har automatiskt nivåanpassning
- okänslig för elektrostatiske urladdningar och inducerat brus.

Spänningar à la carte

Den huvudsakliga uppgiften för hårdvaran i AVR-programmeraren är att generera de spänningar som är nödvändiga för programmerings och konverteringssignaler från mikrokontroller till RS232-nivåer och tvärtom. Den positiva matningsspänningen för AVR-programmeraren tas från de två signallinjerna RTS och DTR på datorns RS232-interface (eller från den nätadapter som kopplats till K3). Spänningen på signallinjerna kan ligga mellan -5 till -12 V eller +5 till +12 V. Dessa spänningar är 'OR-ade' och likriktade med dioderna D1-D4 och begränsade till ungefär 12 V av zenerdioden D5 (för att vara på säkra sidan). Motstånden R2-R4 ger strömbegräns-

ning och dämpar störningar som kommer från RS232-interfacet.

I praktiken är den maximala spänningen på RS232-interfacet betydligt lägre än gränsvärdet på 12 V. Men om en yttre strömkälla används måste du se till att den lämnade spänningen inte överstiger 12 V.

Matningsspänningen på 12 V används endast för att försörja operationsförstärkaren (opampen). Resten av kretsen (IC1, en 74 HC126) behöver en stabiliserad spänning på 3,6 V, vilket lämnas av spänningsregulatorn

IC5 (MAX666). Utgångsspänningen från denna spänningsregulator ställs in på 3,6 V med hjälp av formeln

$$R14 = R15 (V_{UT} \div 1.30 V - 1).$$

Motståndet R13 ställer in den maximala utgångsströmmen från regulatorn enligt formeln

$$R13 = (0.5 V) \div I_{CL}.$$

Här är den maximala strömmen runt

30 mA. Eftersom den maxström som kan dras från RTS och DTR-linjerna är ca 10 mA kan du öka värdet på motståndet till 33 Ω om du inte använder någon yttre spänningskälla. Den negativa matningsspänningen för opampen tas från TxD-linjen, vars spänning i tomgångsläge ligger mellan -5 V och -12 V och således är väldigt lämplig för detta ändamål. Den negativa spänningen begränsas till -12 V av D7.

KOMPONENTLISTA

Motstånd:

R1,R30-R35 = 10k Ω
 R2,R3,R4,R7,R8,R10,R16 = 10 Ω
 R5,R6,R9,R12,R15,R19-R22 = 100k Ω
 R11 = 270k Ω
 R13 = 12 Ω
 R14 = 180k Ω
 R17 = 220k Ω
 R18 = 120k Ω
 R23...R26 = 100 Ω
 R27,R29 = 470 Ω
 R28 = 4k Ω

Kondensatorer:

C1,C3,C5,C7,C8,C10,C11 = 100nF
 C2,C4,C6,C9 = 10 μ F 16V radial
 C12 = 1 μ F 16V radial
 C13,C14 = 22pF

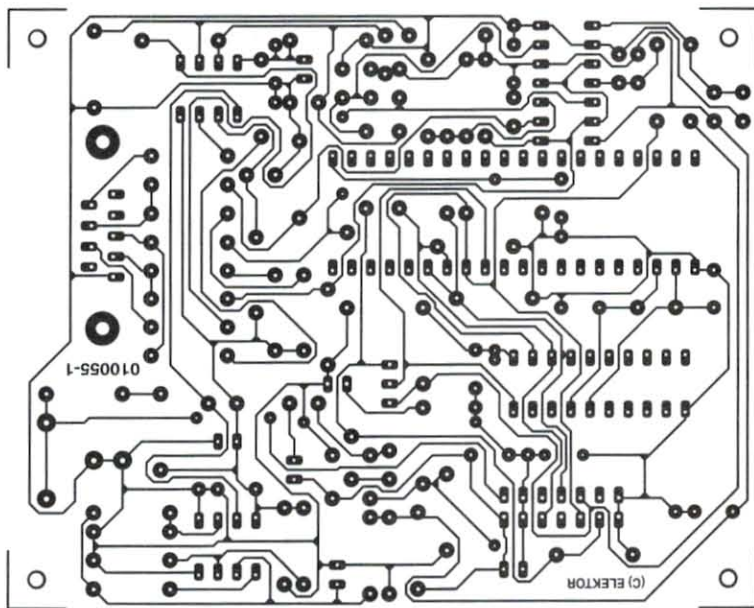
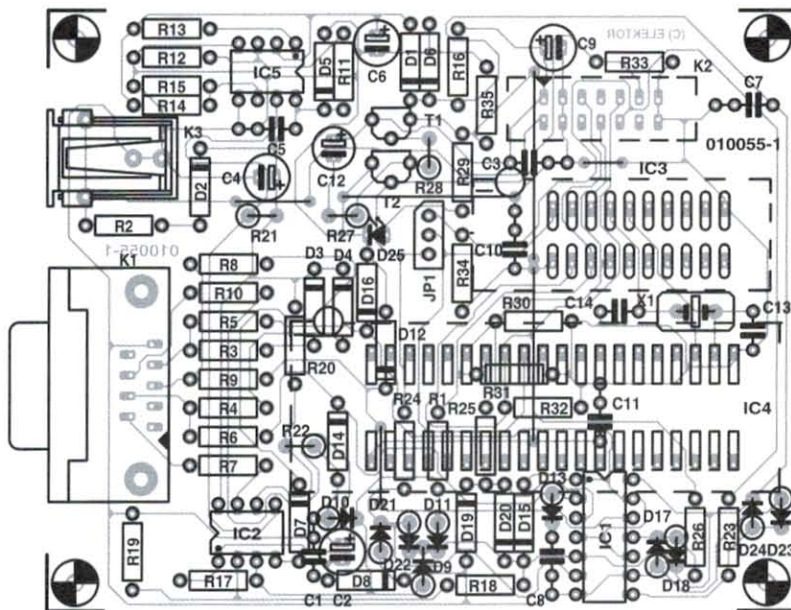
Halvledare:

D1-D4,D7,D9-D24 = BAT85
 D5,D8 = zenerdiod 12V 500mW
 D6 = zenerdiod 5V6 500mW
 D25 = LED, röd, dagsljus
 T1 = BC557
 T2 = BS170
 IC1 = 74HC126
 IC2 = TL082
 IC3 = 20-way ZIF-hållare
 IC4 = 40-way ZIF-hållare
 IC5 = MAX666CPA eller -EPA

Miscellaneous:

JP1 = 3-pol stiftlist med bygel
 K1 = 9-way d-sub (hane), vinklade stift, för kretskort
 K2 = 14-pol boxheader
 K3 = jack för nätadapter
 X1 = 2.4576MHz kristall
 Kretskort 010055-I
 Diskett med mjukvara 010055-I I
 (se Läsaarservice)

Layout och mjukvara kan hämtas gratis från www.alltomelektronik.com



Figur 2. Alla komponenterna sitter på ett enkelsidigt kort (finns att köpa). De två ZIF-hållarna, bygeln, LED-en och hållaren för in-circuit programmering sitter på kortets undersida.

MAX666 mäter ingångsspänningen via spänningsdelaren R11/R12. Utgången från batteriövervakaren, ben 7 (LBO = Low Battery Output) blir aktiv om spänningen enligt formeln

$$R11 = R12 (V_{BATT} \div 1.30 \text{ V} - 1)$$

faller under minimivärdet. Denna signal (open drain) omvandlas till en RS232-signal (DSR) av opamp IC2b och utvärderas av mjukvaran för AVR-programmeraren. Mjukvaran ser till att det alltid finns tillräckligt med spänning. En extern strömkälla kan behövas för att programmera EEPROM (beroende på vilken PC som används), men det är inte nödvändigt för att programmera AVR-kontrollern.

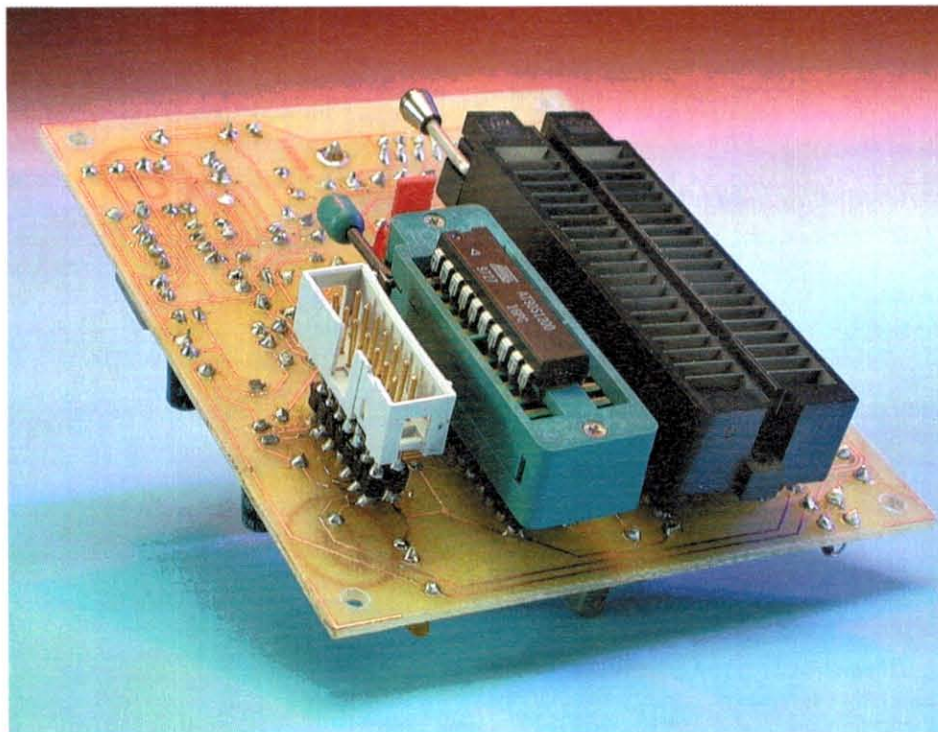
När du sätter i och tar ur mikrokontrollern måste den stabiliserade spänningen från MAX666 kopplas bort från hållaren. Denna uppgift tar mjukvaran hand om via DTR-linjen, med hjälp av de två switchnings-transistorerna T2 (BS107) och T1 (BC 557). RC-filtret R27/C12 fördröjer switchningen av spänningen för att kunna behålla den spänning som behövs för att radera chipet (Chip Erase) under Reset-pulsen. Motståndet R35 laddar snabbt ur restspänning så fort som matningsspänningen för mikrokontrollern har switchats från.

Från TTL till RS232 och tvärtom

Mikrokontrollerns TTL-signal MISO omvandlas till RS232-nivå och kallas sedan för CTS. I stället för den vanliga MAX232 använder vi här en opamp (IC2a) kopplad som en komparator, exakt som för den tidigare beskrivna omvandlingen från LBO till DSR. Detta är inte bara billigare, utan vi kan också ställa in specifika referens (tröskel) spänningar och switchningshysteres. För in-circuit programmering i synnerhet är detta ganska viktigt eftersom matningsspänningen för AVR-mikrokontrollern, som lämnas av målkretsen, kan variera över området 2,7 V-6,0 V. De relevanta formlerna är

$$U_{on} = U_{ref} - R1 / R17 (U_{IH} - U_{amin})$$

$$U_{off} = U_{ref} - R1 / R17 (U_{IL} - U_{amax})$$



Figur 3. Du kan också använda en 24-bens hållare...

för till och frånslags nivåerna och

$$U_{ref} = V_{cc} R19 / (R19 + R18)$$

referensspänningen. Naturligtvis gäller samma sak för IC2b.

Omvandlingen av signalerna DTR, RTS och Txd till motsvarande signaler RESET, MOSI och SCK implementeras med hjälp av IC1 (en 74HC126). De fyra utgångssignalerna (RESET, RESETN, MOSI och SCK) aktiveras av en hög nivå på DTR-linjen. För höga spänningar på IC-ingångarna minskas till rätt nivå av två dioder och ett strömbegränsande motstånd på varje ingång.

För drivutgångarna ger två dioder också överspänningsskydd, vilket är mycket viktigt när man använder långa flatkablar för in-circuit programmering.

Alla komponenter (vilket också är det enda rätta för en bra programmerare (som visas av komponentplaceringssritningen i figur 2) är monterade på kortet, inklusive kontakterna för:

- en nätadapter,
 - en 1:1 RS232 kabel till PC:n och
 - in-circuit programmering (K2).
- Självklart finns det nollkraftsocklar för 20-bens och 40-bens DIL-kapslar

för de mikrokontroller som skall programmeras. Dessa skall monteras på kortets undersida. Detta gäller också K2, lysdioden och bygeln JP1. 20-bens ZIF-hållare går inte att få tag på, men en för 24 ben fungerar lika bra, som kan ses av figur 3. De extra benen kan helt enkelt 'dingla' i luften.

Kretskortet är enkelsidigt (och därmed billigare) och det finns därför flera trådbyglar på kortet. Att montera komponenterna behöver knappast förklaras närmare.

Mjukvaran för AVR-programmeraren

Mjukvaran inkluderar alla inställningar och funktioner som är nödvändiga för seriell programmering av mikrokontrollern:

- val av COM-port, programmeringsmod och programmeringshastighet
- läsning, skrivning och verifiering av programminnet (Flashminne)
- läsning, skrivning och verifiering av det integrerade EEPROM-et (finns inte i alla AVR)
- radering av minnet (Flash och EEPROM)
- inställning av Lockbiten

Inställningarna och funktionerna kan specificeras antingen direkt som kommandoparametrar eller som parametrar i en kommando fil (AVR.PROG.COM). Filen genererad av AVR-assemblern eller kompilatorn i den Atmelspecifika 'Generic Hex Code' kan användas.

das för att programmera mikrokontrollern direkt. Tabell 1 visar de parametrar som behövs för programmering.

Följande är ett exempel på anrop av AVR-programmerarens mjukvara i en kommandorad:

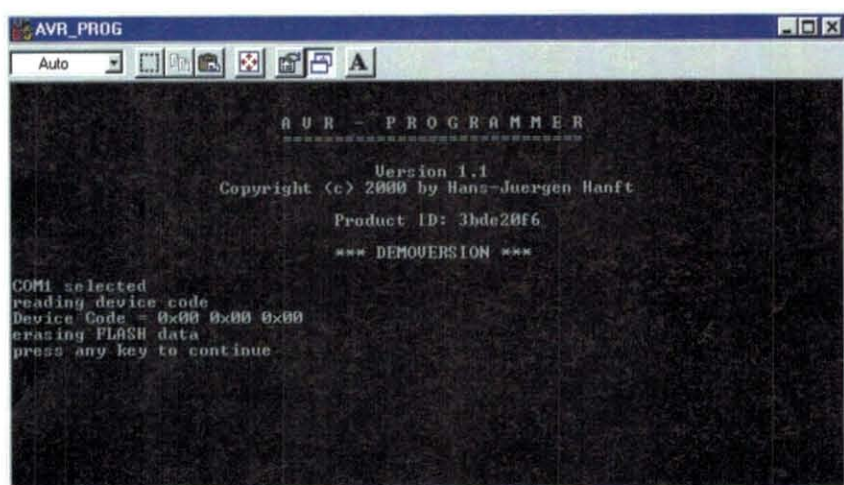
```
C:\AVR_PROGRAMMER\AVR_PROG /COM1
/WRF(dice.hex,,)
```

Detta får filen `dice.hex` att programmeras in i Flashminnet. Alternativt kan man specificera supplementära start och slutadresser för programmeringen. Detta är dock inte nödvändigt såvida inte endast en del av programminnet skall skrivas till. Om du redan har skrivit till program-

minnet måste funktionen `ERASE-CHIP` först köras. I detta fallet ser kommandoraden för programanrop ut på detta sätt:

```
C:\AVR_PROGRAMMER\AVR_PROG/
COM1/EC /WRF(dice.hex,,)
```

Funktionsparametrar	Beskrivning
<code>/RDF(filnamn, startadress, slutadress)</code> <code>/RDFLASH(filnamn, startadress, slutadress)</code> <code>/READFLASH(filnamn, startadress, slutadress)</code>	Läser Flashminnet i AVR-kontrollern och lagrar data i en fil.
<code>/WRF(filnamn, startadress, slutadress)</code> <code>/WRFLASH(filnamn, startadress, slutadress)</code> <code>/WRITEFLASH(filnamn, startadress, slutadress)</code>	Skriver data från en fil till Flashminnet i AVR-kontrollern.
<code>/VYF(filnamn, startadress, slutadress)</code> <code>/VYFLASH(filnamn, startadress, slutadress)</code> <code>/VERIFYFLASH(filnamn, startadress, slutadress)</code>	Jämför data i Flashminnet hos AVR-kontrollern med data i en fil.
<code>/RDEEP(filnamn, startadress, slutadress)</code> <code>/RDEEPROM(filnamn, startadress, slutadress)</code> <code>/READEEPROM(filnamn, startadress, slutadress)</code>	Läser EEPROM-minnet i AVR-kontrollern och lagrar data i en fil.
<code>/WREEP(filnamn, startadress, slutadress)</code> <code>/WREEPROM(filnamn, startadress, slutadress)</code> <code>/WRITEEEPROM(filnamn, startadress, slutadress)</code>	Skriver data från en fil till EEPROM-minnet i AVR-kontrollern..
<code>/VYEEP(filnamn, startadress, slutadress)</code> <code>/VYEEPROM(filnamn, startadress, slutadress)</code> <code>/VERIFYEEPROM(filnamn, startadress, slutadress)</code>	ämför data i Flashminnet hos AVR-kontrollern med data i en fil.
<code>/RDDC</code> <code>/RDDEVICECODE</code> <code>/READDEVICECODE</code>	Läser typkoden för AVR-kontrollern.
<code>/EC</code> <code>/ERASECHIP</code>	Raderar minnena i AVR-kontrollern (Flash och EEPROM).
<code>/SLB1</code> <code>/SETLB1</code> <code>/SETLOCKBIT1</code>	Sätter Lock Bit 1 hos AVR-kontrollern.
<code>/SLB2</code> <code>/SETLB2</code> <code>/SETLOCKBIT2</code>	Sätter Lock Bit 2 hos AVR-kontrollern.
<code>/WAIT(x)</code>	Abryter utförande av kommandon under x millisekunder($0 \leq x \leq 60000$).
<code>/BREAK</code>	Avbryter kommandon. Fortsätter med nästa kommandon när någon tangent trycks in.
Inställning av parametrar	Beskrivning
<code>/COMx</code>	Specificerar port-COM. $x = 1-8$ (startvärde $x = 1$) <code>/SCK(x)</code>
<code>/SCK(x)</code>	Specificerar programmeringshastighet. $x = 1-12$ (startvärde $x = 7$)
<code>/MODE(x)</code>	Specificerar programmeringssätt: $x = 0 \rightarrow \text{AT90S}....$ (startvärde) $x = 1 \rightarrow \text{AT89S}....$
<code>/+MOSI</code>	Sätter positiv logik för MOSI-signalen. (startvärde)
<code>/-MOSI</code>	Sätter negativ logik för MOSI-signalen.
<code>/+MISO</code>	Sätter positiv logik för MISO-signalen. (startvärde)
<code>/-MISO</code>	Sätter negativ logik för MISO-signalen.
<code>/+RESET</code>	Sätter positiv logik för RESET-signalen. (startvärde)
<code>/-RESET</code>	Sätter negativ logik för RESET-signalen.
<code>/+TESTVCC</code>	Aktiverar verifiering av matningsspänning före programmering (startvärde)
<code>/-TESTVCC</code>	Avaktiverar verifiering av matningsspänning före programmering.



Figur 4. Mjukvaran (Windows konsolapplikation) för programmeraren.

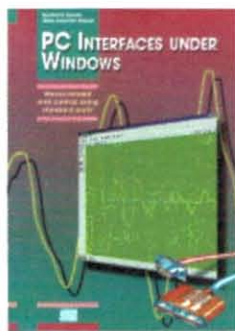
Det är också möjligt att specificera supplementära start och slutadresser för datajämförelse. Detta ger ett funktionsanrop som har följande form:

C:\AVR_PROGRAMMER\AVR_PROG\COM1/VYF (dice.hex, 0x20, 0x2F)

Mikrokontroller-EEPROM kan programmeras på liknande sätt.

(010055-1)

Ny bok med CD-ROM



PC-interface kan användas till mer än bara skrivare, mus, modem och joystick! Medan det var ganska lätt att direkt komma åt PC-interface med en DOS-dator är det inte lika enkelt under Windows. Denna bok visar hur detta kan göras. Konstruktion och körning av program och kretsar för följande applikationer beskrivs utförligt: kontrollkretsar, mätinstrument, analog/digitalomvandlare, EPROM-programmerare, lagringsoscilloskop, I2C-applikationer, göra mätningar med ljudkortet, göra mätningar med videofångningskort, asynkron seriell dataöverföring och mycket mer. Förutom exakta, praktiskt

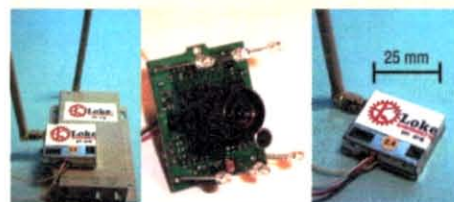
orienterade beskrivningar av traditionella PC-interface (vad de kan göra och hur de adresseras via mjukvara) beskriver författarna gör-det-självt konstruktioner och programmering av ett antal mycket intressanta kretsar, som alla kan kopplas till PC-portarna. Det är inte nödvändigt att öppna PC:n för något av dessa projekt. Applikationer som beskrivs i denna bok baseras på 32-bitars operativsystem (Windows 95/98 och senare). Programmeringsspråken är Visual Basic och Delphi. CD-skivan innehåller alla programexemplen i VB5, Delphi 3 eller 4 med alla källfiler i körbar form. Port.DLL finns också med. Dessutom finns det två kompletta applikationsprogram på CD-skivan, COMPUNI.EXE, ett universellt interfaceapplikationsprogram och SSCAN.EXE, en komplett oscilloskopapplikation för ljudkortet.

296 sidor, engelska. Pris: 390:- inkl. moms. Porto tillkommer.

Allt om Elektronik, Box 178, 444 22 Stenungsund
Tel: 0303-770 490, Fax 0303-770 490
order@alltomelektronik.com, www.alltomelektronik.com

Kamerasystem

flyg, bil, u-båt, tåg, spion etc



Ultralätta kretskortskameror och videosändare med lång räckvidd för bl.a. elflyg och helikopter. Vi för allt i branschen: Mörkerseende, ljusförstärkande och vattentäta kameror, Flata TFT-skärmar, Inspelningsutrustningar mm. Egen import = bra priser. Kolla in www.loke.nu för erbjudanden och komplett sortiment.



Loke

LOKE Getingtåkten 739 91 Skinnkatteberg
Tel: 0222-152 00 order@loke.nu

Flygbandsmottagare

en dubbelomvandlande superhet
för 108-137 MHz NAV och KOM-mottagning

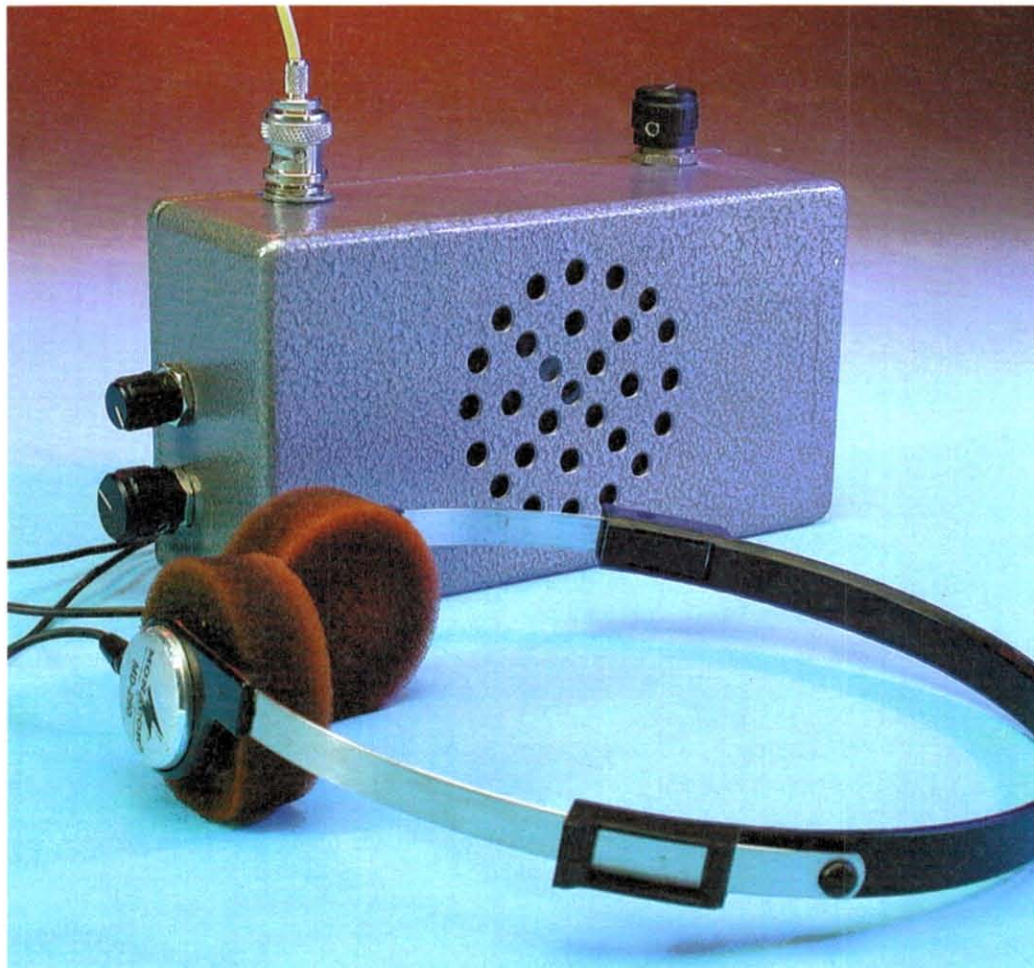
Design av G. Baars

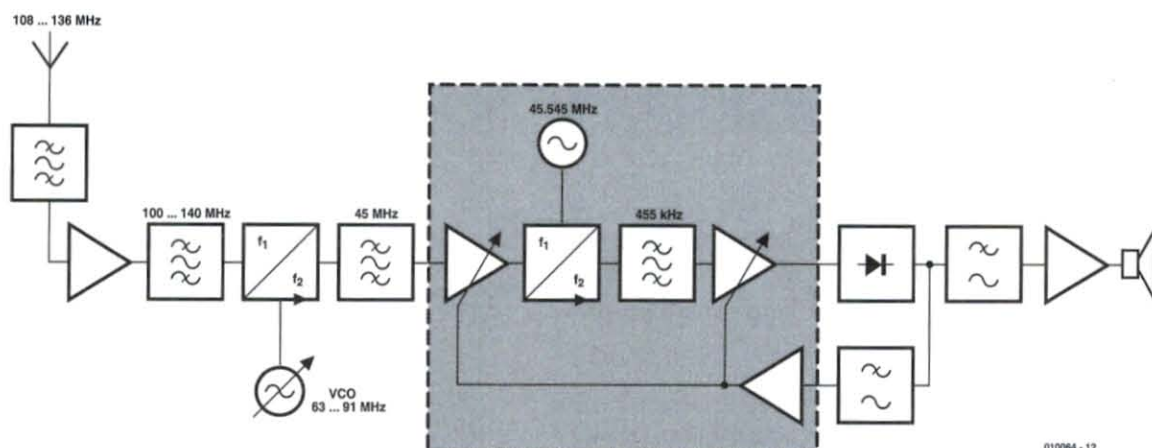
gert_baars@hetnet.nl

Denna mottagare, speciellt designad för VHF-flygbandet, kombinerar bra prestanda med enkel konstruktion, allt till ett överkomligt pris. Kretsen innehåller inga exotiska kretsar och kan justeras utan specialinstrument. Vi tycker därför att detta är en idealisk nybörjarmottagare för flygentusiaster med båda fötterna fast placerade på marken.

Att tjuvlyssna på kommunikation inom polis, ambulans och brandkår, för att nämna några få exempel, är en attraktiv hobby för många människor. Och så har det varit ända sedan man började använda öppen kommunikation. Varför det är så roligt att lyssna är svårare att förstå. Nyfikenhet naturligtvis, men det är inte den enda orsaken. En möjlig faktor kan vara att tjuvlyssning ligger någonstans i gråzonen mellan vad som är lagligt och vad som är olagligt (i många länder är det olagligt), vilket utan tvekan ger en extra krydda åt lyssnandet.

Ett av de mer populära frekvensområdena att lyssna på är känt som VHF-flygbandet. Här hör man i stort sett all kommunikation mellan flygledarna, piloterna och markpersonalen. Detta band kopplar ihop spänningen med att avlyssna med den spänning som omger flygning över huvudtaget och detta lockar många. VHF-flygbandet definieras generellt som frekvensområdet mellan 108 MHz och 137 MHz, vilket indikerar att det är avsett att bilda en sömlös länk med VHF FM-bandet, 87,5 MHz till 108 MHz. Detta kan leda oss till antagandet (eller hoppet) att med





Figur 1. Mottagaren är en dubbelomvandlande superheterodyn med mellanfrekvenser vid 45 MHz och 455 kHz.

lite intelligent modifiering kan en befintlig FM-mottagare klämmas in i den nedre delen av VHF-flygbandet. Men detta är inte så enkelt som det först verkar. Först och främst är den bandbredd som används för FM-rundradio mycket större än i VHF-flygbandet och detsamma gäller kanalsepareringen (100 kHz i motsats till 25 kHz). Resultatet av detta är att känsligheten hos en FM-radio är helt otillräcklig. För det andra säger reglerna att all kommunikation på flygbandet skall använda amplitudmodulering (AM), vilket skulle betyda att den befintliga FM-demodulatorn måste tas bort och ersättas med en AM-motsvarighet. För att göra en lång historia kort, låt oss glömma FM-radion och i stället koncentrera oss på en dedikerad VHF flygbandsmottagare.

Överväganden

För att göra helt klart vad det är för slags mottagare vi kommer att diskutera, så kan det vara på sin plats med en lista över viktiga egenskaper. I stort sett allt vi nämner nedan kommer vi att gå igenom mer i detalj längre fram i artikeln när vi fördjupar oss mer i elektroniken.

- Vad som kanske är den viktigaste egenskapen, den här mottagaren är en **dubbelomvandlande superheterodyn** design, omfattande två blandare, två lokaloscillatorer (LO) och två mellanfrekvens (MF) förstärkare. Superhetprincipen ger mycket bra mottagaregenskaper när det gäller spegelfrekvensdämpning och känslighet.
- Den första LO är en VCO (spänningsstyrd oscillator) med kapacitansavstämning (tuning), **grov** och **fin**.
- Eftersom projektet använder **färdigköpta spolar**, så behöver du inte vara någon HF-

specialist, som radioamatör. Det är bara en enda spole du behöver linda själv, en mycket enkel luftspole.

- Justeringarna **kräver inte någon specialutrustning** och kan utföras genom att bara lyssna.
- Eftersom hela mottagaren, inklusive audioförstärkaren och strömförsörjningens stabilisering, finns på **ett enda kretskort**, finns det knappast någon ledningsdragning alls.
- Den **mottagna bandbredden** är lätt att välja genom att montera ett keramiskt filter med en bandbredd på 6 kHz eller 15 kHz.
- Mottagaren har utrymme för expansion med en räknare för frekvensvisning och en extern PLL för avstämning. Vi har dock för närvarande inga planer på att konstruera dessa expansioner.

Blockschema

Mottagarens kompletta struktur visas i figur 1.

Den HF-signal som plockas upp av antennen (ca 60 cm lång) filtreras först för att dämpa komponenter som ligger utanför bandet. Därefter följer en 20 dB förstärkare och ett filter med ett passband på ca 100-140 MHz. Huvudfunktionen för detta filter är att hålla signalerna vid spegelfrekvenserna borta från ingången till HF-förstärkaren.

I den första blandaren blandas den filtrerade och förstärkta antenssignalen med utgångssignalen från en VCO (spänningsstyrd oscillator). VCO:n har ett frekvensområde på 63

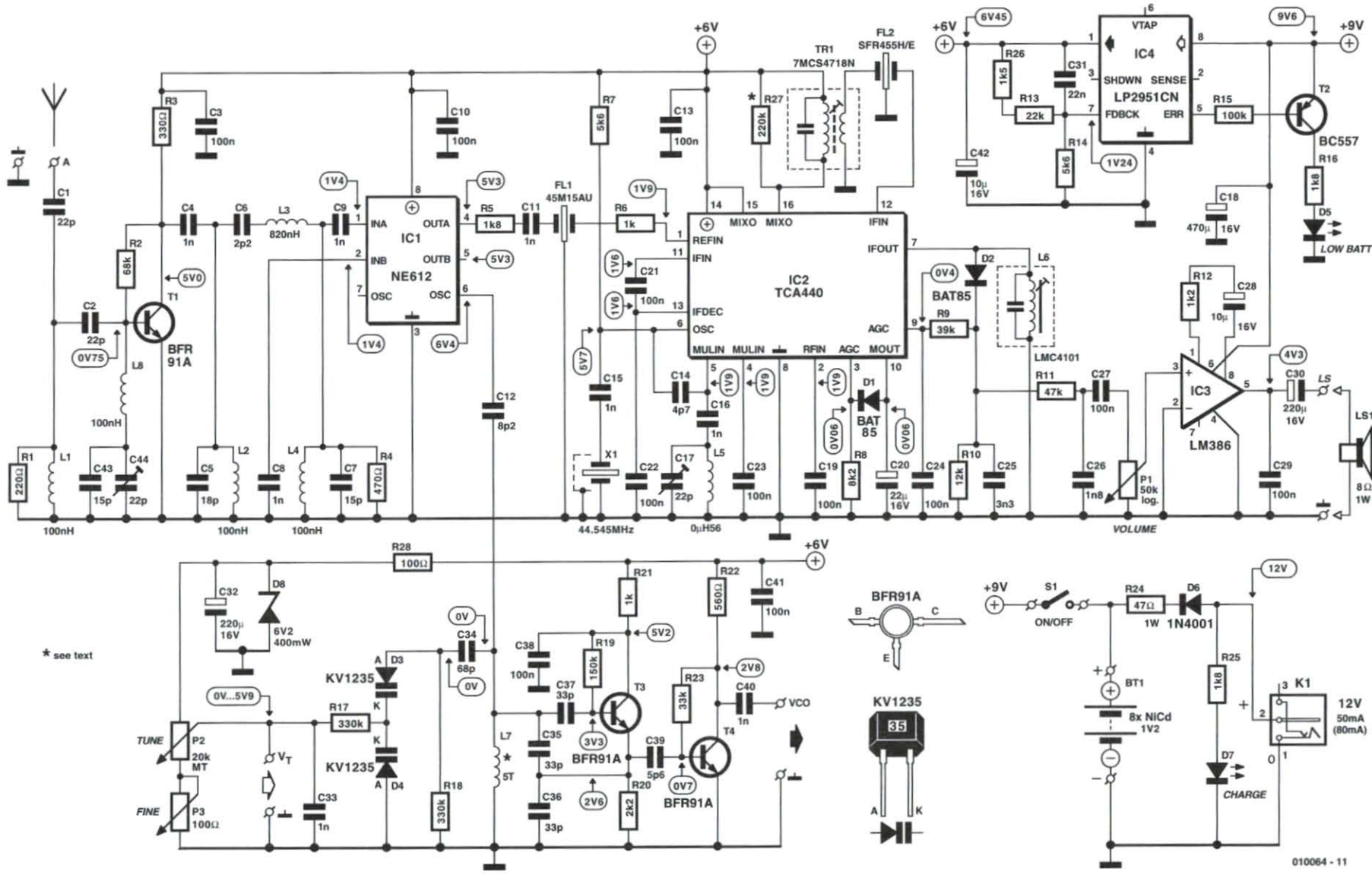
MHz till 91 MHz och används för att stämma av mottagaren. Den skillnadssignal som uppstår som en följd av att blanda HF och VCO-signaler har en fast frekvens på 45 MHz. Denna kallas för den första MF (mellanfrekvensen). Med hjälp av ett 45-MHz filter tas alla störningar bort från den första MF.

Den första MF-signalen förstärks sedan innan den läggs till den andra blandaren där den blandas med en 44.545 MHz signal från en fast oscillator. Den resulterande skillnadssignalen på 455 kHz filteras igen och förstärks sedan. Därefter kommer AM-demodulatorn. Bandbredden på 455-kHz filtret bestämmer mottagarens totala selektivitet.

Efter demodulatorn passerar signalen genom en buffert innan den läggs till förstärkningsstegen före och efter den andra blandaren. Detta är den automatiska förstärkningskontrollen (AGC), som används för att minska den totala förstärkningen i mottagaren när extremt starka signaler tas emot. AGC-n jämnar ut större variationer i signalstyrkan och på så sätt slipper du att hela tiden justera volymen när du vrider in en annan signal.

Som visas av den streckade linjen i blockschemat så finns den andra blandaren, 44.545-MHz oscillatorn, de två justerbara förstärkarna och AGC-n i en enda integrerad krets. Detta gör det betydligt enklare att bygga projektet än vad det varit med diskreta komponenter.

Efter AM-demodulatorn hittar vi ett enkelt lågpasfilter följt av en liten



Figur 2. Tack vare den integrerade blandar/oscillator/MF-förstärkaren TCA440 är kretsschemat lättöverskådligt.

audioförstärkare och, naturligtvis, den vanliga högtalaren.

Praktiskt utförande

Kretsschemat för VHF flygbandsmottagaren visas i figur 2. Låt oss nu titta på hur de funktioner vi gått igenom ovan kan utföras rent praktiskt.

Antennsignalen kommer in på L1, med ett notchfilter bestående av L8-C43-C44 för dämpning av oönskade signaler. Ingångsförstärkaren för HF, T1, är en bipolär transistor av typ BFR91. Denna komponent ger en hygglig förstärkning med ett accepterbart brustal. Bandpasset på 100-140 MHz är ett 3-poligt Butterworthfilter bestående av L2-L3-L4-C5-C6-C7. Detta filter, hjälpt av 'grov' filtret vid ingången, ger ungefär 50 dB spegel-frekvensdämpning.

Den första blandaren använder den välkända IC-kretsen NE612, som tar emot VCO-signalen vid ben 6 via kopplingskondensatorn C12.

VCO:n är byggd runt transistor T3, en annan BFR91 i en modifierad Colpittskonfiguration, som är en klassisk krets inom HF-tekniken och känd för sin goda stabilitet. Oscillatorns resonanskrets avstämms med två kapacitansdioder (varicaps), D3 och D4, vars kapacitans är en (icke-linjär) inverterad funktion av den avstämningsspänning som läggs över dem via deras gemensamma katod. Avstämningsspänningen kan justeras mellan 0,5 V och ca 6 V med potentiometerna P2 (grov) och P3 (fin). Filtret R28-D8 agerar som en extra stabilisator för avstämningsspänningen och hjälper till att motverka frekvensdrift hos VCO:n, vilket annars kan få mottagaren att tappa inställningen.

Via anslutningen VT görs kapacitansdiodelnas styrspänning tillgänglig externt om man (senare) bestämmer sig för att använda en PLL-synthesiser för att stämma av mottagaren.

På samma sätt görs utgångssignalen från VCO:n tillgänglig via bufferten T4 så att en frekvensdisplay skall kunna anslutas. Om du inte planerar att använda denna expansion kan du utelämnat T4, C39, R22 och R23 när du bygger kretsen på kortet.

Filtret vid utgången från den första blandaren är en 45-MHz keramisk typ med en nominell bandbredd på 15 kHz. Detta filter följs av sektionen inom den streckade linjen i block-schemat. Alla dessa funktioner (förförstärkare, blandare, oscillator, MF-förstärkare och AGC) finns i kretsen TCA440, som (i stort sett) bildar en enchips radiomottagare. Av de mer eller mindre standardkomponenterna runt TCA440 är utan tvekan de viktigaste kristallen på 44.545 MHz, X1. LC-kombinatio-

nen L5-C17 för den interna oscillatorn och 455-kHz bandpassfiltret bestående av transformatorn Tr1 och det keramiska filtret FL2. Spolen L6 fungerar som en avstämd utgångskrets.

Om vi rör oss mot kretsens utgång hittar vi en enkel dioddetektor, D2, för AM-demodulering, ett lågpassfilter, R10-R11-C25-C26, och slutligen, en integrerad audioförstärkare av typ LM386, IC3.

Strömförsörjning

Mottagaren konstruerades att arbeta från en ostabiliserad 9 V matning. Matningsspänningen försörjer audioförstärkaren IC3 direkt, såväl som spänningsregulatorn IC4, som ger en stabiliserad spänning på 6 V (egentligen 6,45 V) för resten av kretsen. Eftersom 'error' utgången från IC4 (ben 5) går låg när ingångsspänningen faller under miniminivån för korrekt stabilisering så används denna för att styra en 'LowBatt' indikator-LED via transistor T2. Då minimispänningsfallet över IC4 endast är 0,1 V kan batteriet 'mjölkas ur' innan LED D5 tänds för att indikera att detta verkligen är tomt.

Mottagaren drar ca 60 mA med en högtalare inkopplad och ca 35 mA med ett par hörlurar på 32-Ω och där båda öronpropparna är parallellkopplade. Följaktligen kommer ett 9-V batteri att räcka 5 respektive 10 timmar. Om du behöver mer batterikapacitet kan du använda åtta 1,2-V NiCd-batterier (typ AA) såsom visas i kretsschemat. Dessa batterier kan laddas genom att ansluta en 12-V nätadapter till K1. LED D7 fungerar då som laddningsindikator, medan motståndet R24 bestämmer nivån på laddningsströmmen. Det visade värdet på 47 Ω ger en (i allmänhet säker) laddningsström på ca 50 mA. Detta betyder att du kan lämna nätadaptern inkopplad hela tiden utan problem, oavsett vilken exakt typ av batteri som används.

Om mottagaren används utan återuppladdningsbara batterier kan komponenterna R24, D6, R25, D7 och K1 utelämnas.

Tuning och selektivitet

Som vi redan nämnt bestämmer det keramiska filtret FL2 selektiviteten

hos mottagaren. Det finns två alternativ, ett filter med en bandbredd på 6 kHz (SFR455H eller CFW455H) eller 15 kHz (SFR455E eller CFW455E). Även om du kanske vill välja den högsta selektiviteten omedelbart, så råder vi dig att använda 15-kHz versionen, åtminstone från början. Radioutrustning som är anpassad till den standard för kanalseparation på 8,33 kHz som introducerades under 1999 för ATC-kommunikation (ATC = flygledare) är fortfarande rätt tunnådd och 25 kHz är fortfarande den vanligaste kanalseparationen. Dessutom är det betydligt svårare att stämma av mottagaren med ett filter på 6-kHz. Trots användningen av en mångvarvig potentiometer för P2 är det lätt att missa stationer. Det finns naturligtvis en fininställningen, men den gör inte stor nytta om du redan vridit förbi signalen.

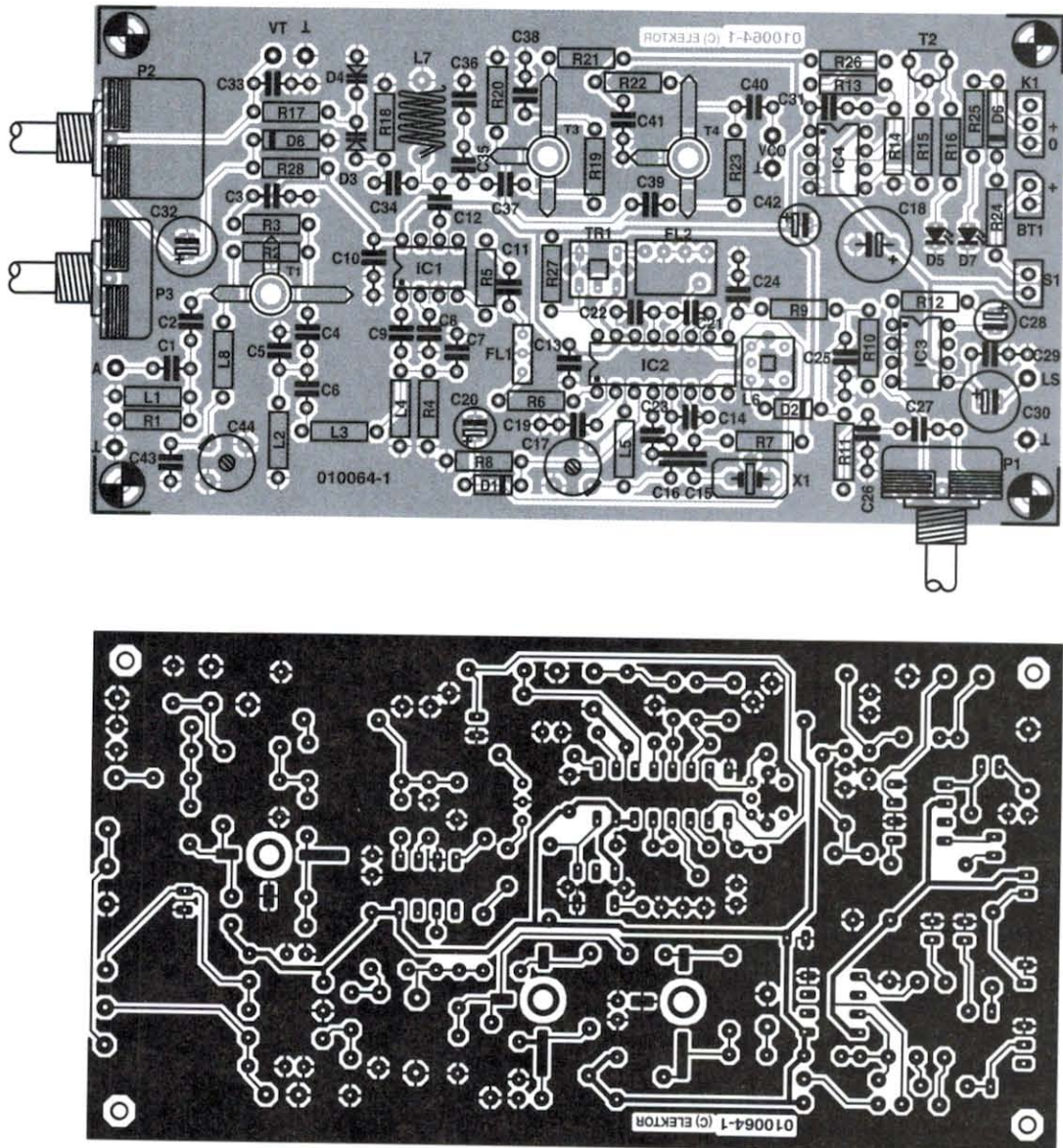
Om du emellertid använder en extern PLL-synthesiser för att avstäm-ma mottagaren är det bättre att använda det smalare filtret, om inte annat så för att minska brusnivån.

En sista anmärkning när det gäller avstämningen, en viss drivning kan märkas precis när mottagaren har slagits på. Denna effekt kommer att försvinna efter ca 5 minuters uppvärmning.

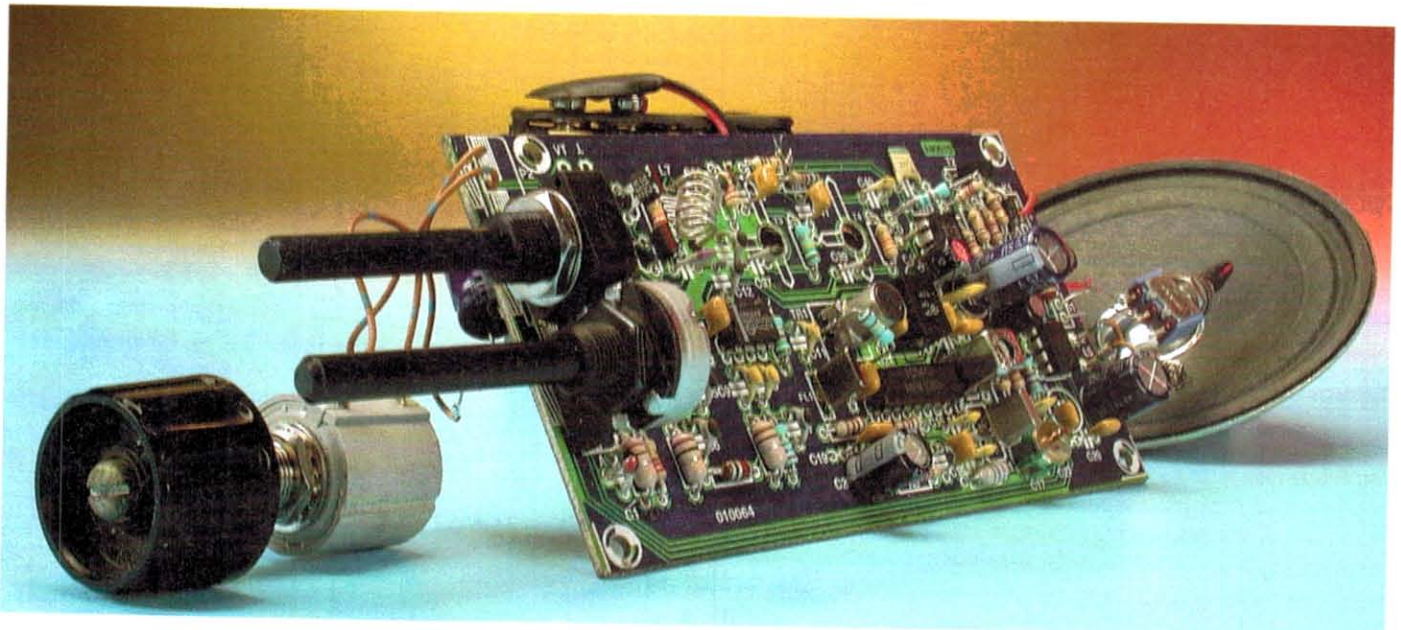
Själva bygget

Figur 3 visar kretskortslayout och komponentplacering för det kort som tagits fram för detta projekt. På kortet finns den krets som visas i figur 2, dvs inkluderande audioförstärkaren IC3, regulatorn IC4 samt NiCd-laddarkretsen bestående av R24, R25, D6, D7 och K1.

Trots det stora antalet komponenter på kortet är själva bygget tämligen enkelt. Som vanligt skall du kolla noga att de polariserade komponenterna vänds åt rätt håll, alltså IC-kretsarna (titta efter det lilla 'hacket'), elektrolytkondensatorerna, transistorerna och dioderna. Kapacitansdiodelna D3 och D4 kräver speciell uppmärksamhet eftersom de inte har någon tydlig markering. Om du håller dioden på så sätt att typkoden är läsbar med benen nedåt så är det vänstra benet anoden och det högra benet katoden. På kortet sitter inte D3 och D4 åt samma håll så var uppmärksam på detta!



Figur 3. Kretskortslayout och komponentplacering. Kortet finns att köpa från oss.



Det är bäst att börja med de låga komponenterna eftersom detta är mest bekvämt. Först motstånd, därefter de små kondensatorerna, elektroliterna osv. Du kan använda hållare för IC3 och IC4, medan IC1 och IC2 kan lödas direkt på kortet.

HF-transistorerna T1, T3 och T4 (T4 endast om en buffrad VCO-utgång behövs) löds på kortets undersida med benen direkt på de relevanta kopparbanorna. De passar bara på ett sätt och det finns hål i kretskortet för deras runda små kapslar.

Sedan kommer spolarna. L1-L5 och L8 är färdiga miniatyrdrosslar som ser ut som precisionsmotstånd, inklusive färgade band som visar deras värde. MF-transformatorn Tr1 och den avstämde kretsen L6 är också färdigköpta komponenter. Båda finns i metallhöljen som bara kan vändas åt ett håll. Den enda hemlindade spolen är L7. Denna är lätt att linda och består av fem varv 1 mm silverpläterad tråd. Den inre diametern är 5 mm och spolen kan därför lindas kring ett 5 mm borr. När spolen är färdig drar du ut den jämnt till en längd på ca 12 mm.

Några få detaljer när det gäller bygget. R24 skall vara på 1 W och monteras något ovanför kortet eftersom det kan bli lite varmt. Motståndet R27 används inte eftersom vårt test visade att detta inte behövs. Lysdioderna D5 och D7 skall monteras så att de kan synas från utsidan. Anslut dem med mjuk kopplingstråd till kortet. Metallhöljet på kristallen X1 skall (snabbt) lödas till jord med en mycket kort bit kopplingstråd (avklipp från komponentben).

Du ser att potentiometrarna P1, P2 och P3 passar direkt på kortet. Om du monterar dem där eller inte beror på vilken låda du tänkt använda. P2 och P3 kan anslutas till kortet med vanlig kopplingstråd medan P1 skall ha en kort bit skärmd audiokabel.

När du monterat alla komponenter på kortet är det en god idé att använda en multimeter för att kolla att det finns rätt spänning vid de indikerade mätpunkterna. Om de är (i stort sett) korrekta kan du utgå från att kretsen är rätt hopbyggd.

Som en ytterligare hjälp att få igång projektet utan allt för mycket felsökning så visar figur 4 ledningsdragningen för den kompletta mottagaren, med kretskortet som centralpunkt.

Mekaniskt arbete

Eftersom det är ganska litet så passar kretskortet in i en relativt liten låda, tillsammans med batterierna och högtalaren. Även om vi inte har några större invändningar mot en ABS-låda (plast) så rekommenderar vi dock starkt en metalllåda eftersom detta minimerar riskerna för VCO-drivning på grund av den

så kallade 'handeffekten'. Vår prototyp byggdes in i en aluminiumlåda av typ BIM5005, som har de yttre måtten 15x8x5 cm. Även om kortet passar in bra i denna låda så blir det trångt om NiCd-batterierna och högtalaren också skall klämmas in. Vi var, till exempel, tvungna att ta bort lite aluminium från lockets insida in till potentiometer P2.

Figur 5 visar prototypen från insidan.

KOMPONENTLISTA

Motstånd:

R1 = 220Ω
R2 = 68kΩ
R3 = 330Ω
R4 = 470Ω
R5, R16, R25 = 1kΩ28
R6, R21 = 1kΩ
R7, R14 = 5kΩ26
R8 = 8kΩ22
R9 = 39kΩ
R10 = 12kΩ
R11 = 47kΩ
R12 = 1kΩ22
R13 = 22kΩ
R15 = 100kΩ
R17, R18 = 330kΩ
R19 = 150kΩ
R20 = 2kΩ22
R22 = 560Ω
R23 = 33kΩ
R24 = 47Ω (1W)
R26 = 1kΩ25
R27 = monteras ej
R28 = 100Ω
P1 = 50kΩ logaritmisk pot
P2 = 20kΩ flervarvig
P3 = 100Ω linjär pot

Kondensatorer:

C1, C2 = 22pF
C3, C10, C13, C19, C21-
C24, C27, C29, C38, C41 = 100nF
C4, C8, C9, C11, C15, C16, C33, C40 = 1nF
C5 = 18pF
C6 = 2pF2
C7, C43 = 15pF
C12 = 8pF2
C14 = 4pF7
C17, C44 = 22pF justerbar (trimmer)
C18 = 470μF 16V radial
C20 = 22μF 16V radial
C25 = 3nF3
C26 = 1nF8
C28, C42 = 10μF 16V radial
C30, C32 = 220μF 16V radial
C31 = 22nF
C34 = 68pF
C35, C36, C37 = 33pF
C39 = 5pF6

Antennen vi använde var en vanlig teleskopantenn. Du kan också använda en bit styv kopplinstråd på 60 cm försedd med en banankontakt.

Justering

Det finns fyra justeringspunkter i mottagaren. Kärnorna hos Tr1 och L6, samt trimmer C17, justeras helt enkelt för maximalt utgångsbrus.

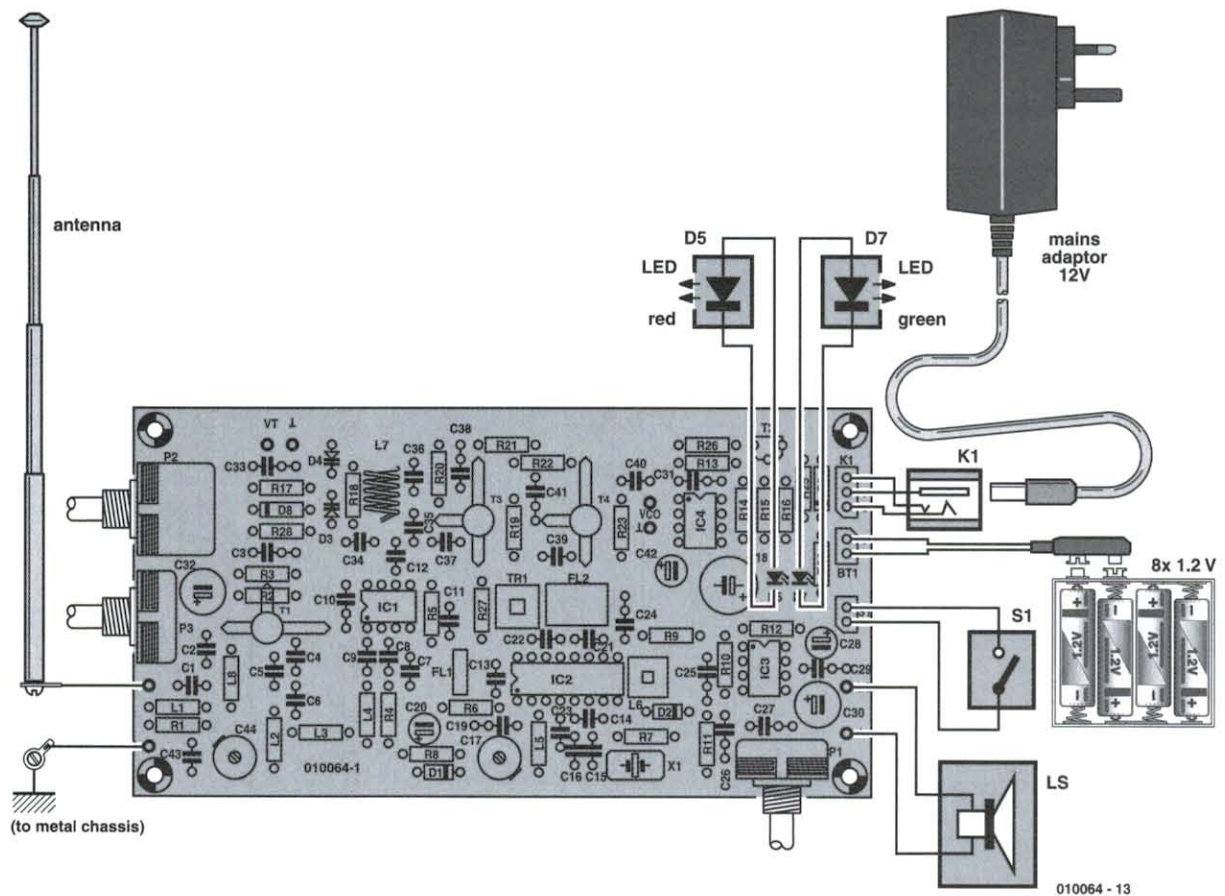
Halvledare:

D1, D2 = BAT85
D3, D4 = KV1235
D5 = LED, röd, dagsljus
D6 = 1N4001
D7 = LED, grön, dagsljus
D8 = zenerdiod 6.2V, 0.4W
IC1 = SA612AN eller NE612
IC2 = TCA440
IC3 = LM386
IC4 = LP2951CN
T1, T3, T4 = BFR91A
T2 = BC557

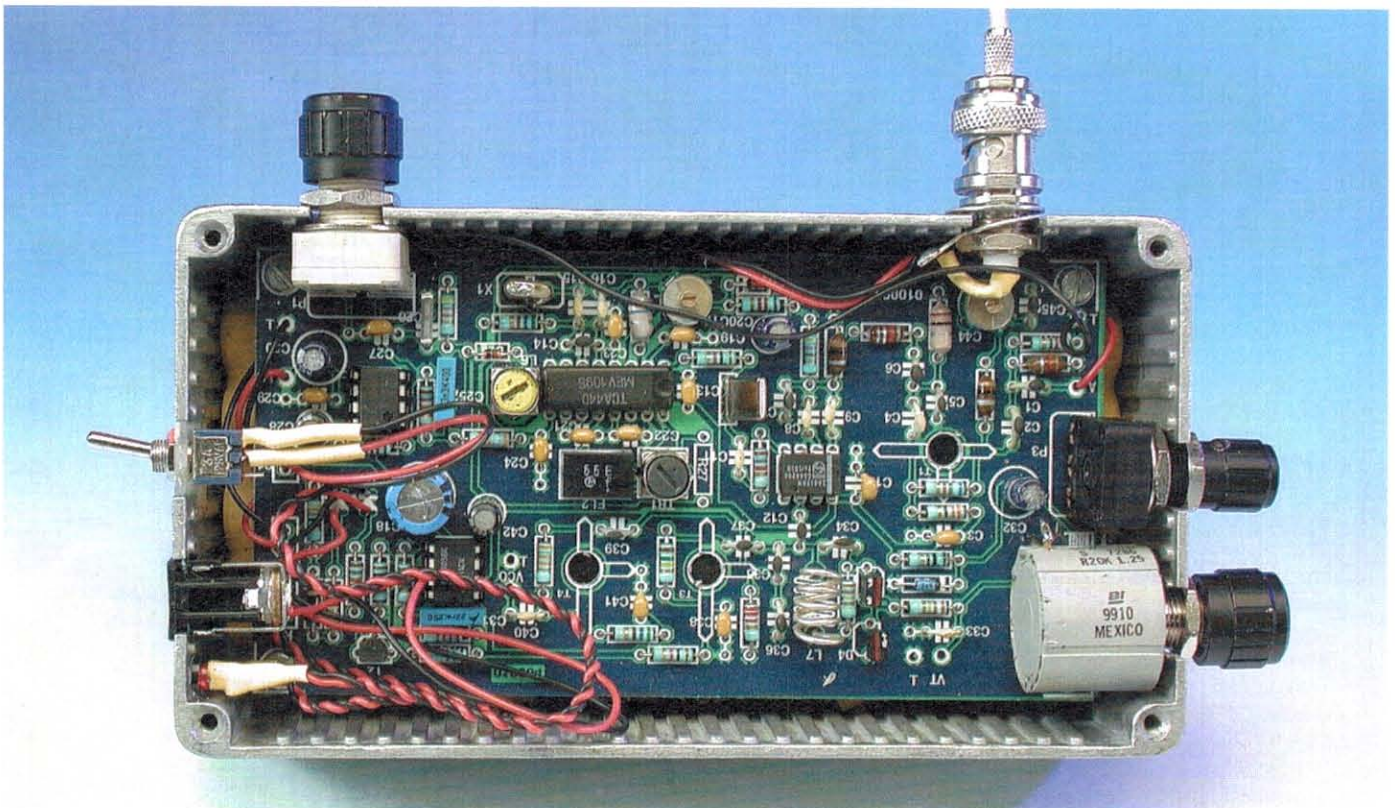
Övrigt:

BT1 = 9V batter (PP3) eller 8 NiCd batterier (1.2V)
FL1 = 45M15AU
FL2 = SFR455H eller -E (CFW455H eller -E)
K1 = jack för nätadapter, för kretskort
L1, L2, L4, L8 = 100nH
L3 = 820nH
L5 = 560nH
L6 = LMC4101 (Toko)
L7 = 5 varv 1 mm silverpläterad tråd på en 5 mm formare (ingen kärna)
S1 = switch, 1 slutande kontakt
TR1 = 7MCS4718N (Toko)
X1 = 44.545MHz kristall, hölje anslutet till jord
LS1 = högtalare 8Ω 1W
Kretskort 010064-1 (se Läsaerservice)
Låda, t.ex. BIM, dim. 150x80x50 mm,

Många HF-delare för detta projekt, inkl. spolar, varicaps, keramiska filter och trimrar finns att köpa från Barend Hendriksen HF Elektronika BV, PO Box 66, NL-6970-AB, Brummen, Nederländerna.
Tel. (+31) 575 561866,
Fax (+31) 575 565012.
Websida www.xs4all.nl/~barendh/
email barendh@xs4all.nl



Figur 4. Översikt över de externa kontrollerna och andra element kopplade till kretskortet.



Figur 5. Som man kan se av denna bild är det ganska trångt !

Trimmern C44 sätts i mittläge och kan sedan justeras för att ta bort starka signaler från närliggande FM-stationer. Det är allt!

Om du noga följt lindningsanvisningarna för polsen L7 så skall nu VCO:n fungera med det korrekta avstämningssområdet, vilket du kan verifiera om du har en frekvensräknare tillgänglig. Anslut denna till VCO-utgången och vrid P2 för att kolla om VCO:n kan justeras mellan 63 MHz och 91 MHz. Om så är nödvändigt kan du dra isär eller trycka ihop lindningarna på L7, små ändringar åt gången!

Mottagning

Den mesta kommunikationen kan plockas upp inom den så kallade KOM-sektionen (kommunikation) av bandet, mellan 117 och 137 MHz. Den undre delen 108-117 MHz, är reserverad för flygfyrar, instrumentlandningssystem (ILS), navigationsfyrar och an-

dra hjälpsystem och det är därför de ofta kallas för NAV-system. Det bästa sättet att hitta de frekvenser som används vid en flygplats i närheten är att söka på flygplatsens namn eller ICAO-nummer (Göteborg/Landvetter har benämningen ESGG t.ex.), och det finns en sida på Internet som visar alla flygplatser i Sverige, och för Landvetter är den

www.geocities.com/swedishairports/ap/goteborg-landvetter.html

flygplatsen. Som du märker är signalen från flygplanen ofta starkare än den från flygplatsen, men det är inte så konstigt om man tänker på att flygplanet beinner sig högt över marken och knappast har några hinder i vägen för signalen.

Slutligen, genom att låta mottagaren ta in svaga signaler från navigationsfyrar kan den användas som en excellent propageringsmonitor för att förutsäga sporadiska E-öppningar i VHF-bandet.

(010064-1)

Vi mätte upp känsligheten hos mottagaren till ungefär 0,5 μ V för 12 dB (S+N/N). Detta är tillräckligt för att kunna plocka upp kommunikation mellan flygledaren och piloterna på ett avstånd av ca 25 kilometer från

Spegelfrekvensdämpning

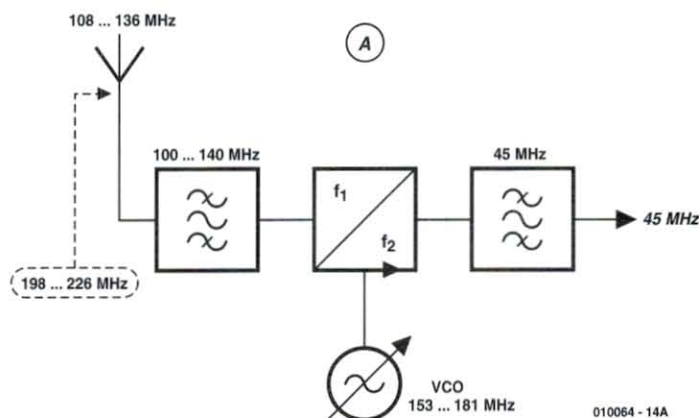
Inbyggt i dess design är alla superheterodyna mottagare (enkel eller dubbelomvandlande) i princip öppna för två band, det önskade bandet och 'spegelfrekvens' bandet. Avståndet mellan dessa band är två gånger den första mellanfrekvensen. Spegelfrekvenser orsakas av oönskade utgångsprodukter från de blandare som används.

I en superheterodyn mottagare blandas den mottagna signalen (HF) med en lokalscillator (LO) på sådant sätt att blandarens utgång producerar en mellanfrekvens (MF) som är konstant över hela frekvensområdet. I den mottagare som visas i figur A finns HF-signalerna inom det önskade bandet mellan 108 MHz och 136 MHz och LO-signalen kan avstämmas mellan 153 MHz och 181 MHz. Detta kallas för högfrekvensinjektion (high-side injection). Skillnadsfrekvensen är enligt $LO - RF = 45$ MHz, vilket är mittfrekvensen hos MF-passbandet.

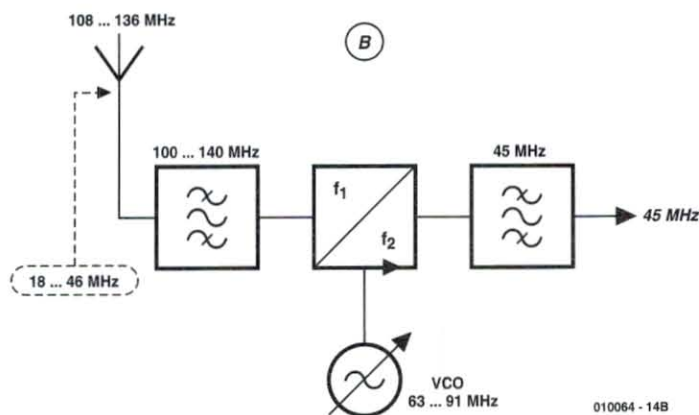
Enkel matematik säger dock att en identisk 45 MHz signal produceras av HF-signaler mellan 198 MHz och 226 MHz, såsom visas i figur A. Det filter som är monterat före blandaren har ett passband som motsvarar det önskade frekvensområdet, dvs. 100-140 MHz och används således för att undertrycka signaler som plockas upp i 'spegelbandet'.

I detta fall, med en mellanfrekvens på 45 MHz, ligger spegelbandet mindre än en oktav från det önskade bandet. Följaktligen måste passbandfiltret vara ganska brant. Alternativt måste dess tuning 'spåra' VCO:n. Båda lösningarna är ganska svåra att implementera vilket inte är vad vi är ute efter.

I detta exempel är bästa sättet att uppnå en bra spegelfrekvensdämpning att använda lågfrekvensinjektion. Att använda ett tuningområde för VCO:n på 63-91 MHz resulterar även detta i en fast MF på 45 MHz (RF-LO). Som visas i figur B ligger spegelbandet då mellan 18 MHz och 46 MHz, vilket är två oktaver från ingångsfilterets passband. Därmed kan dessa spegelfrekvenser enkelt undertryckas med hjälp av ett relativt enkelt passbandsfilter.



010064 - 14A



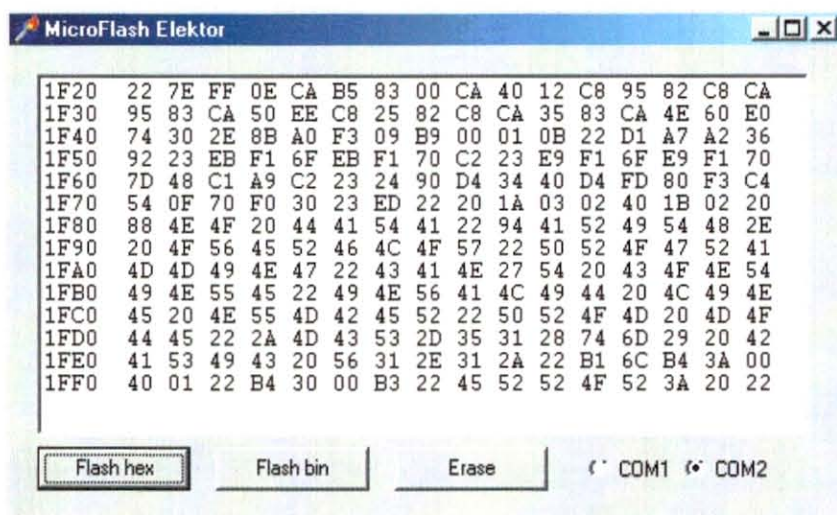
010064 - 14B

Grundkurs i mikro-kontrollerteknik (3)

Del 3: BASIC-52

Av B. Kainka

I de första två artiklarna i denna kurs arbetade vi med assembler, men nu är det dags för ett högnivåspråk: BASIC-52.



Figur 1. Nedladdning av Basic-tolken.

Denna tolk för 8052-kontroller är välkänd för att alla mikrokontrolleranvändare och universellt populär. Den finns på programdisketten för Flash-kortet i form av en programfil i Intel hexformat med namnet BASIC-52.hex. Allt du behöver göra är att använda programmet Micro Flash.exe för att ladda denna fil in i mikrokontrollern (figur 1). Tolken är 8 kB stor och tar ca 1 minut att föra över. För att kunna använda tolken be-

höver du ett terminalprogram som kan skicka instruktioner och program till mikrokontrollern via serieinterfacet. Här använder vi programmet BASIC.EXE från H.J. Bernt. När du först startar programmet måste du konfigurera den port som skall användas (COM1 eller COM2). Kommunikationen mellan mikrokontrollersystemet och PC:n sker sedan via kontakten K1. Som visas i figur 2 har programmet två textfönster. Det övre redigeringsfönstret används för

att redigera källtexten i Basic. Varje rad kan redigeras så mycket du vill och skickas inte till mikrokontrollern innan du trycker på <Return>. Det undre fönstret är en direktterminal. Alla tecken som läggs in i detta fönster skickas direkt till mikrokontrollern. Respon- sen från systemet visas också i detta fönster. Efter en Reset initialiserar först BASIC-52 sig själv och kollar tillgängligt RAM i systemet. Efter detta väntar tolken på 'mellanslags- tecknet' från terminalen. Detta används för att automatiskt bestämma den baudrate som skall användas och för att konfigurera det seriella interfacet hos mikrokontrollern i enlighet med detta. Detta betyder att du inte får trycka på någon annan tangent än mellanslag (ASCII 20h). BASIC-52 svarar med följande startmeddelande:

```
*MCS-51 (tm) BASIC V1.1*
READY
>
```

Nu kan du ange direkta kommandon eller programrader. Frågan Print MTOP returnerar den högsta adressen hos det anslutna minnet (Memory Top), vilket i detta fallet är '32767' för 32 kB RAM. När du anger kända nyckelord behöver du inte bry dig om stora eller små bokstäver. Tolken omvandlar dock allting in- ternt till stora bokstäver och programlist- ningarna visas också alltid på detta sätt. Du kan prova detta med ett litet testprogram som heter Test1.bas (se figur 3).

```
10 for n= 1 to 10
20 print n
30 next n
```

Kommandot LIST får programmet att skickas ut via serieinterfacet och visas i terminalfönstret. Det kan startas med hjälp av RUN. Vårt första program genererar ett ökande antal tal i terminalfönstret.

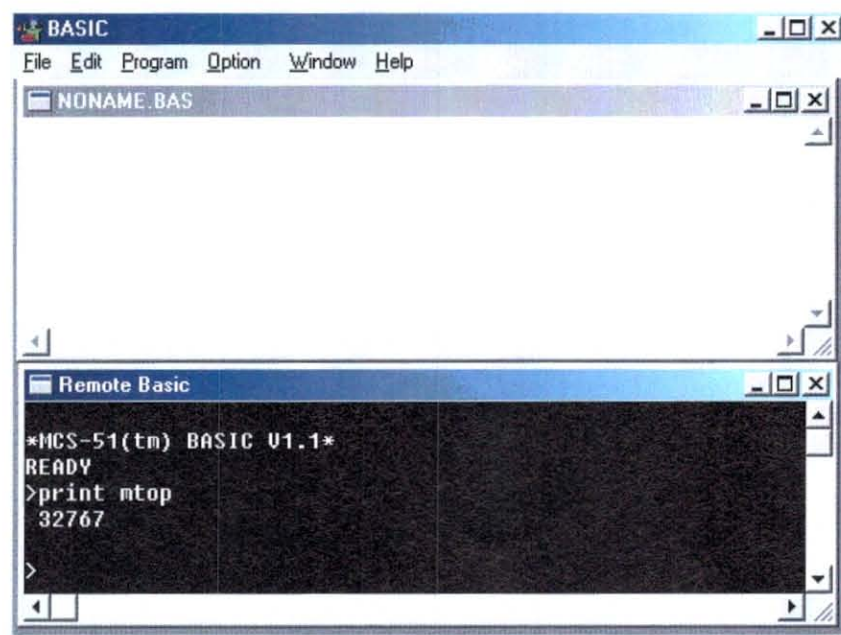
Kommandon som LIST, RUN och NEW kan direkt läggas in som text. Du kan emellertid också använda de inbyggda funktionerna i Programmenyn. Program/List kopierar listorna till redigeringsfönstret. När de finns i detta fönster kan de antingen redigeras eller sparas på hårddisken.

Nu är det dags för en kort förklaring av vårt programexempel. I rad 10 definieras variabeln N. Denna inkrementeras (ökas) sedan från 1 till 10 i tio steg. FOR... TO... NEXT bildar en loop (slinga) som körs tills värdet 10 uppnås. Detta betyder att N får värdet '1' vid första passeringen genom loopen, '2' vid andra passeringen osv. upp till 10 vid den sista passeringen. I rad 20 skickas det aktuella värdet ut via serieinterfacet och kan skrivas till skärmen.

En räknande loop kan också byggas i assembler utan problem. Kommandot PRINT är dock så komplext att det skulle behövas ett ganska omfattande assemblerprogram för att uppnå samma funktion. BASIC-52 använder inte bara byte (området 0-255), vilka direkt kan förstås av mikrokontrollern, utan också riktiga tal, som vart och ett består av sex byte som representerar åtta signifikanta siffror, ett förtecken (+ eller -) och en exponent. PRINT-kommandot måste omvandla dessa siffror till text, tecken för tecken, via serieinterfacet. Detta kräver en initialisering av interfacet, vilket utförs av tolken när den startas.

Nu är det inte meningen att vi skall gå ge en omfattande, och långtråkig, förklaring av tolkens alla kommandon och nyckelord. En kort kommandosumming kan hittas i redigeringsprogrammets Hjälppil. En omfattande summing av kommandona, komplett med en användarmanual, kan du hämta gratis från vår websida www.alltomelektronik.com under Download. I de följande experimenten kommer du att stöta på en hel del nya nyckelord och lära dig hur du skall använda dem. Vi börjar med mycket enkla exempel som funktionellt är helt kompatibla med de assemblerprogram som du redan har sett. Det är mycket intressant att jämföra hastigheten hos dessa funktionellt ekvivalenta program. Vi börjar med Test2.bas:

```
10 FOR N=0 TO 255
20 PORT1=N
30 NEXT N
40 GOTO 10
```

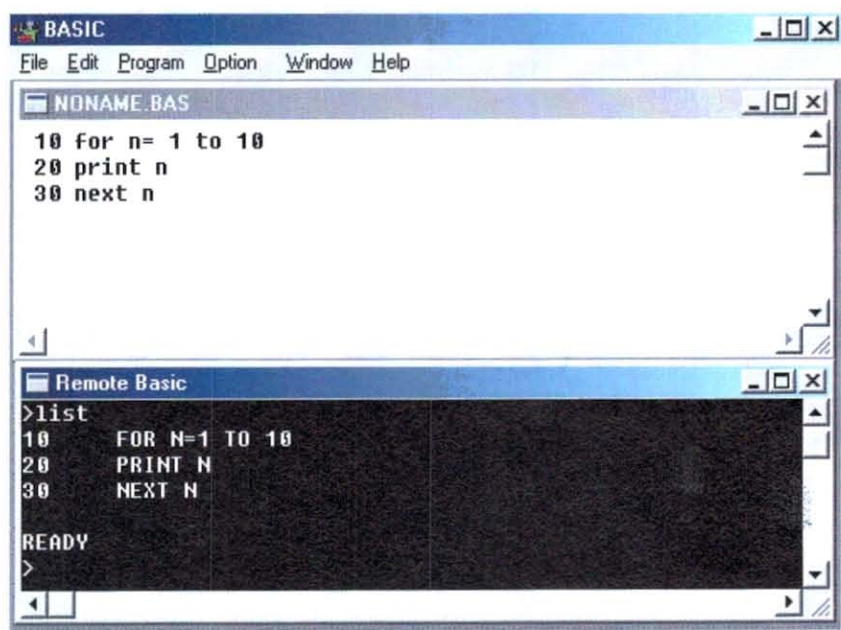


Figur 2. Det terminalprogram som används i kursen.

Här använder vi igen vår välkända räkneloop, denna gång med området 0-255, vilket täcker det numeriska området för ett enkelbyte. I BASIC-52 kan ett tal skickas ut till Port 1 lika enkelt som till skärmen. PORT1=N överför värdet på variabeln N till porten. I motsatt riktning kan tillstånd hos alla åtta portarna läsas med hjälp av instruktionen N=PORT1.

Så fort som du startat programmet

med hjälp av RUN kan du se hur snabbt det är. En fyrkantsignal med en frekvens på ca 200 Hz kommer att ligga på P1.0. Vid P1.7 kommer signalnivån är växla ungefär en gång per sekund och kan användas för att driva en lysdiod direkt. Det jämförbara assemblerprogrammet var något snabbare, även med en extra fördröjningsloop. Basicprogram går mellan 100 och 1000 gånger långsammare än ett jämförbart assem-



Figur 3. Att lägga in ett nytt program.

blerprogram. Detta är inte överraskande eftersom BASIC-52 är en tolk, vilket betyder att det är ett ganska komplicerat program som skall tolka och exekvera (utföra) källtext i Basic ord för ord. Att det är en komplicerad uppgift kan man se på storleken hos tolken, 8 kB. Alla våra tidigare programexempel i assembler i de två första artiklarna var, tillsammans, mindre än 100 byte. I dessa exempel har vi endast skrivit enkla, linjära, program. Tolken måste, i motsats till detta, utföra en mycket omfattande uppdragslista. När man kör ett Basicprogram skall nyckelorden kännas igen och utvärderas. Dessutom behandlar tolken alla numeriska värden som riktiga tal, vilket kräver betydligt mer arbete än att bara processa bytes.

En frekvensdelare i Basic

Nu är det dags att ge BASIC-52 en konkret uppgift: En ingång skall konstant övervakas för pulser. Efter 10 pulser skall en utgång vipa. Efter totalt 20 pulser skall utgången återgå till sitt originaltillstånd. Programmet bildar således en frekvensdelare med en delningsfaktor på 20. Här använder vi ett program för att utföra en uppgift som normalt utförs av några IC-kretsar. En intressant fråga är att avgöra de gränser inom vilka detta är möjligt.

I BASIC-52 finns det inget enkelt sätt att adressera individuella portbitar. Följaktligen måste vi alltså läsa från, eller skriva till, hela porten. Information om tillståndet hos en individuell bit kan erhållas genom att utföra en AND-operation på det totala tillståndet hos porten (masking). BASIC-52 har .AND. byteoperatör för detta ändamål. Punkterna före och efter 'AND' indikerar bitbehandlingen, i motsats till den logiska länkningen av uttryck (IF villkor 1 AND villkor 2 ...). I följande listning anges maskingen i raderna 110 och 130. Praktiskt sett utför detta kommando åtta samtidiga AND-operationer mellan två olika satser om åtta bit-tillstånd. Värde på varje resulterande bit kommer att vara '1' bara om båda bitarna i samma position har värdet '1'. Detta kan illustreras med följande exempel:

MCS BASIC-52 VI.3

Den senaste revisionen av tolkens originalprogram, som utfördes av H.-J. Böhlings och D. Wulf, har vi redan presenterat i nummer 2/2001 av *Allt om Elektronik*. Denna version av tolken har vi också testat framgångsrikt med Flashkortet. Programmet har specialkommandon för programmering och radering av EEPROM. Flera olika Basicprogram kan lagras i ett enda EEPROM och startas selektivt. De individuella stegen, som visas nedan, har testats med hjälp av ett 8-kB EEPROM:

1. Skriv ett program.
2. Använd XFER, och programmera ett EEPROM med detta program.
3. Använd XFER, och ladda ytterligare program.
4. PROG returnerar antalet lagrade program (t.ex. 12).
5. Använd ROM för att switcha till EEPROM.
6. Använd ROM2 för att aktivera det andra programmet.
7. Använd PROG2 för att göra det första programmet i EEPROM ett autostartprogram.

Nu kommer det första programmet i EEPROM att starta automatiskt efter en reset, utan behov av någon terminal.

```
10101010 .AND.
00000001 =
-----
00000000

11110001 .AND.
00000001 =
-----
00000001
```

Som du kan se ger en .AND. operation med 00000001b = 01h = 1 endast värdena '0' och '1'. Detta betyder, att med full avsikt, har endast bit 0 på porten tagits hänsyn till medan uppträdandet hos de övriga bitarna har maskats ut. Vi har alltså täckt porten med en mask som endast tillåter att en bit kan synas, vilket är den bit som har valts som ingång. Samma sak kan också göras i assembler, även om det i detta fall inte är nödvändigt eftersom det där är möjligt att direkt polla individuella bitar.

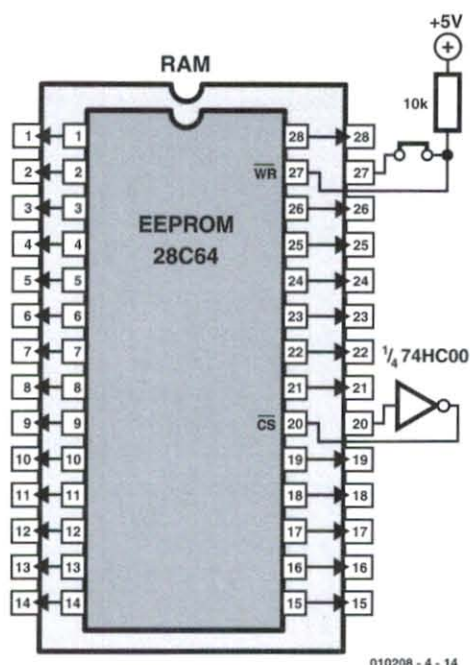
Vårt frekvensdelarprogram (divide.bas) ser ut så här::

```
100 COUNT=0
110 INP=PORT1.AND.1
120 IF INP=1 THEN GOTO 110
130 INP=PORT1.AND.1
140 IF INP=0 THEN GOTO 130
150 COUNT=COUNT+1 :
    REM PRINT COUNT
160 IF COUNT=10 THEN
    GOTO 200
170 IF COUNT=20 THEN
    GOTO 250
180 GOTO 110
```

```
200 REM Output low
210 PORT1=253:REM P1.1 = 0
220 GOTO 110
250 REM Output high
260 PORT1=255:REM P1.1 = 1
270 COUNT=0
280 GOTO 110
```

Basicprogram innehåller ofta många GOTO hopp, vilket minskar deras läsbarhet, speciellt när det gäller stora projekt. Detta kallas ofta för 'spagettikod', vilket betyder att koden är kaotisk och svår att reda ut. Frekvensdelarprogrammet är ett bra exempel på detta. Det visar dock också att denna stil är ganska bra för relativt små program. Större program å andra sidan, kräver definitivt en bättre struktur, vilken man kan, till exempel, uppnå med subrutiner.

Programmet använder en variabel (COUNT) som en pulsräknare och en annan variabel (INP) för ingångstillståndet på Port 1.0. I början av programmet är COUNT satt till noll. Programmet passerar därefter genom två loopar i vilka porttillståndet (statusen) kontinuerligt läses och utvärderas. Så länge som ett Högt tillstånd ses förblir programmet i raderna 110 och 120. Raderna 130 och 140 hanterar Lågt tillstånd på exakt samma sätt. Loopen avslutas endast när ett nytt Högt tillstånd uppstår, vilket ger ett hopp från rad 180 tillbaka till 110, där hela processen påbörjas på nytt. Emellertid, på vägen till rad 180 inkrementeras värdet i rad 150. Raderna 160 och 170 kollar om ett visst räknartillstånd har uppnåtts (10 och 20) och exekverar hopp till andra delar av programmet, som switchar utgångens tillstånd om detta är fallet. Koden för Lågt tillstånd börjar i rad 200, medan den för Högt tillstånd börjar vid rad 250. Förutom det-



Figur 4. Koppling av EEPROM.

ta nollställs räknaren vid dessa platser. Den viktiga egenskapen hos detta program är att testa de villkor som antingen gör att vissa hopp exekveras eller undviker att göra dessa hopp. IF... THEN... ELSE... betyder således att om det specificerade villkoret är uppfyllt så hoppa då till rad den eller den och om det inte är uppfyllt så exekvera helt enkelt nästa rad. Villkoret ges i rad 120 som `INP=1`. Detta betyder att här görs en jämförelse, med resultatet att jämförelsen är sann om ingången hade ett Högt tillstånd vid exakt samma tid som den samplades i föregående rad. Programmet innehåller flera kommentarer, vilka föregås av 'REM' och som används för att öka läsbarheten hos programmet. En rad kan innehålla mer än en instruktion, men dessa instruktioner måste skiljas åt med hjälp av ett kolon (:). Basic gör det lättare att leta efter fel eftersom det är möjligt att 'tillfälligt' infoga en PRINT instruktion var någonstans som helst för att kolla vilka mellanliggande resultat som kan finnas. I rad 150 ser du resterna av ett sådant test. För att kunna se om den egentliga räknaren arbetade så lades instruktionen PRINT COUNT till. Detta gör det mycket enkelt att kolla om ingångspulserna utvärderades korrekt eller inte. Vår första version av exempelprogrammet hade ett fel som inte längre kunde döljas när denna instruktion adderades. Felet uppstod eftersom, under programutvecklingen, vi glömde att nollställa räknaren (rad 270), med resultatet att jämförelser mellan specifika räkningar (rader 160 och 170) inte ledde någonstans. Nä

allting rättats till kunde vi 'kommentera ut' PRINT instruktionen med hjälp av REM, vilket rent praktiskt betyder att den togs bort. Naturligtvis kunde vi också ha raderat den, men om exakt samma utgångsinstruktion någon gång i framtiden skulle visa sig användbar vid testning kan vi helt enkelt ta bort 'REM' och på så sätt återställa det tidigare tillståndet.

Den intressanta frågan nu är, vilket är den övre frekvensgräns som räknaren kan uppnå? För att svara på denna fråga ansluter vi en funktionsgenerator till Port P1.0 och tittar på P1.1 utgången med hjälp av ett oscilloskop. Vårt första intryck är ganska nedslående. Trots att enkla LS TTL-IC lätt kan hantera denna uppgift med en ingångsfrekvens på 50 MHz så nåddes gränsen med denna Basicräknare redan vid 50 Hz. Programmet är således ungefär en miljon gånger långsammare. Om ingångsfrekvensen höjs över 50 Hz tappas ingångspulser bort eftersom vid den tid som pulsen skulle ha samplats så hade programmet ännu inte återvänt till den rad som samplar ingången. Programmet kan dock sampla pulser något fortare än vad en människa kan.

Att jämföra programmet med en rent elektronisk lösning, eller en manuell lösning, är inte riktigt rättvist. Den korrekta frågan är i stället, i vilka fall kan Basiclösningen vara den rätta lösningen. Det finns potentiella applikationer där relativt långsamma resultat skall räknas och utvärderas. Vi kan, till exempel, bygga en maskin som lindar spolar och där vi vill att motorn skall stängas av efter exakt 1650 varv. I detta fall har Basicprogrammet en fördel eftersom det snabbt och enkelt kan modifieras och räkneområdet begränsas inte av hårdvara.

Autostart för BASIC-52

Assemblerprogram har den fördelen att de är autostartkapabla med Flashkortet. Detta betyder att de kan automatiskt börja arbeta av sig själva varje gång strömmen slås på. Detta är tyvärr inte direkt möjligt med BASIC-52 eftersom det egentliga programmet är lagrat som databyte i RAM-et. När strömmen stängs av förloras allting som finns i RAM.

Även om vi använder ett RAM med batteriback-up startar det tidigare programmet i alla fall inte automatiskt. I stället skulle det helt raderas av Basicolken eftersom BASIC-52 helst vill börja med ett helt tomt RAM.

Utvecklingarna av tolken försåg den dock från början med en funktion som låter programmet ligga i EPROM i den minnesregion som börjar vid 8000h. Denna möjlighet kan också användas med ett supplementärt EEPROM placerat i samma minnesregion. Det finns två möjliga sätt att implementera denna expansion. Det första är att använda systembussen, som finns tillgänglig på kontakten K8, medan det andra är att löda en annan hållare 'piggyback' ovanpå RAM-et. Nästan alla ledare kan anslutas en-till-en, och det är endast CS (ben 20) och WR (ben 27) som behöver speciell behandling (se figur 4).

RAM-et ockuperar adressområdet 0-8FFFh. Adressavkodningen är mycket enkel eftersom det bara handlar om att ansluta A15 till RAM-ets CS-ben. I enlighet med detta enkla arrangemang måste adresssignal A15 inverteras innan den läggs till CS-ingången på det supplementära EEPROM-et. På detta sätt blockeras EEPROM-et i det lägre minnesområdet och ockuperar området över 8000h.

Från experiment med EEPROM verkar det som om det finns en risk för att deras innehåll förändras när strömmen slås på. Detta beror förmodligen på en kort puls på WR-ingången. Om detta skulle hända är EEPROM-et inte till stor nytta eftersom autostarten endast fungerar om det lagrade programmet är felfritt. Lösningen på detta är en bygel som kan spärra den oavsiktliga programmeringen av EEPROM-et. Figur 4 visar den kompletta modifieringen av ett 28C64 8-kB EEPROM, inklusive inverterare och skrivskydd.

Ursprungsversionen av BASIC-52 kan programmera sina egna EPROM men detta kräver speciell hårdvara som vi inte går in på här. Detta är emellertid inte något problem eftersom programmeringsprocessen för EEPROM-et är så enkel att den kan utföras med hjälp av ett litet Basicprogram. Det hjälpprogram som listas nedan kan utföra detta jobb.

Det läggs på toppen av ett befintligt program (i detta fall programmet i raderna 10-40) och börjar med hjälp av GOTO 9000. Det kopierar helt enkelt att programbyte från RAM till EEPROM-et, med början från adress 200h i RAM och adress 8011h i EEPROM-et. Tre ytterligare byte ger en problemfri autostart genom att lämna viktigt information, såsom baudrate inställning, till operativsystemet. För korrekt funktion är det också viktigt att använda MTOP för att begränsa den övre delen av RAM-et.

```
10FOR N=0 TO 255
20PORT1=N
30NEXT N
40GOTO 10

9000 N=200h : E=8011h
9010 D=XBY(N)
9020 PRINT N,D
9025 XBY(E)=D
9030 N=N+1 : E=E+1
9040 FOR T=1 TO 10 :
      NEXT T
9050 IF D<>1 THEN
      GOTO 9010
```

```
9100 XBY(8000H)=32H
9105 FOR T=1 TO 10 :
      NEXT T
9110 XBY(8001H)=0FFH
9115 FOR T=1 TO 10 :
      NEXT T
9120 XBY(8002H)=0DCH
9125 FOR T=1 TO 10 :
      NEXT T
9130 XBY(8010H)=055H
mtop = 8191
goto 9000
```

Så fort som EEPROM-et har programmerats kan du använda kommandot ROM för att switcha till den övre adressregionen. Det lagrade programmet kan nu startas på normalt sätt med RUN och stoppas med Ctrl-C, men det kan inte redigeras. Du kan använda RAM för att gå tillbaka till den normala adressregionen. På detta sätt är det möjligt att ha två separata program tillgängliga i systemet. Varje gång matningsspänningen slås på startas det program som ligger i EEPROM-et automatiskt. Om detta program skall användas permanent skall programmeringsbygeln tas bort

så att du inte modifierar programmet av misstag.

Detta avslutar vår korta introduktion till att arbeta med BASIC-52. I nästa artikel kommer vi att arbeta med språket C. Detta kommer då att fullborda vår jämförelse mellan de tre mest viktiga programmeringsspråken.

(010208-4)

RIKTLINJER VID BYGGEN

Allt om Elektronik säljer inte några andra komponenter än kretskort, folier till frontpaneler samt mjukvara på diskett eller IC (inte nödvändigtvis för alla projekt). Komponenter finns att köpa från ett antal återförsäljare, se länkar på vår websida.

Stora och små komponentvärden indikeras med hjälp av något av följande prefix:

E (exa) = 10^{18}	a (atto) = 10^{-18}
P (peta) = 10^{15}	f (femto) = 10^{-15}
T (tera) = 10^{12}	p (pico) = 10^{-12}
G (giga) = 10^9	n (nano) = 10^{-9}
M (mega) = 10^6	μ (micro) = 10^{-6}
k (kilo) = 10^3	m (milli) = 10^{-3}
h (hekto) = 10^2	c (centi) = 10^{-2}
da (deka) = 10^1	d (deci) = 10^{-1}

I vissa kretsschema, för att undvika missförstånd, men i motsats till IEC-rekommendationerna, ges komponentvärdena genom att ersätta det relevanta prefixet med ett decimalkomma. Till exempel,

3k9 = 3,9 k Ω

4 μ 7 = 4,7 μ F

Om inte annat anges är toleransen hos motstånden $\pm 5\%$ och effektmärkningen 1/3-1/4 watt. Arbetsspänningen hos kondensatorer är ≥ 50 V.

När du bestyr kretskortet skall du alltid börja med de minsta passiva komponenterna, dvs trådbygglar, motstånd och små kondensatorer. Därefter IC-hållare, reläer, elektrolyter och andra större kondensatorer samt kontakter. Känsliga halvledare och IC-kretsar monteras sist.

Vid lödning skall du använda en 15-30 W lödpenna med fin spets och lödtenn med flussmedel (60/40). Sätt in komponentbenen i kortet, böj dem lite, klipp av dem och löd. Vänta 1-2 sekunder på att tennet skall flyta ut och ta bort lödpennan. Överhetta inte, speciellt inte med IC och halvledare. Avlödning görs bäst med en avlödningssläta eller en tennsug.

Felsökning. Om kretsen inte fungerar så jämför noga kortet med komponentplaceringsskizzen och komponentlistan. Sitter komponenterna på rätt plats. Är de vända åt rätt håll? Har du vänt matningsledningarna fel? Är alla

lödningar rätt gjorda? Har du glömt någon trådbygel?

Om det finns spänningar angivna i kretsschemat se då till att spänningarna på kortet matchar dessa, avvikelser på upp till $\pm 10\%$ är acceptabla.

Eventuella rättelser publiceras i tidningen.

Motståndsvärden indikeras av färgband på följande sätt



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

Exempel:

brun-röd-brun-guld = 120 Ω , 5%

gul-violett-orange-guld = 47 k Ω , 5%

ISAK (4)

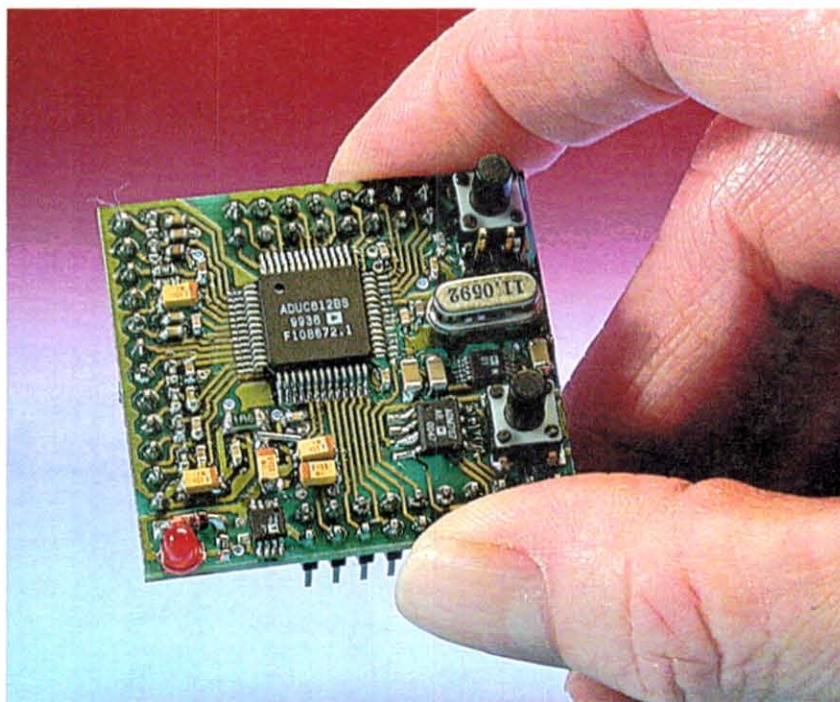
Del 4: Basapplikationer

Av Prof. B. vom Berg, P. Groppe och M. Müller-Aulmann

I den fjärde artikeln i denna serie skall vi gå igenom de första applikationerna för ISAK-kuben och det enkla moderkortet, vilket ger en bra introduktion till kretsen möjligheter. Kompletta programexempel som utvecklats för dessa applikationer, tillsammans med extra information, finns att hämta från vår websida.

Innan vi presenterar de individuella applikationsexemplen vill vi ge några kommentarer angående programexemplen och deras användning.

- Applikationerna och demonstrationsprogrammen är skrivna i C51-kod på så sätt att de (med ett undtag) kan användas med den begränsade versionen av μ Vision2 (max kodstorlek 2 kByte, ingen flytpunktsmatematik). För större applikationsprogram måste fullversionen av μ Vision2 användas, eller något annat 8051-verktyg.
- För att kunna producera någonting 'synligt' guidar alla program användaren via PC-skärmen och hit produceras också utgångarna. Detta betyder att du måste köra ett terminalprogram på din PC. Vi har utgått från att du använder HyperTerminal eftersom detta program ingår som en del av Windows. Du kan dock använda något annat terminalprogram, kommunikationsparametrarna skall vara 9600 baud, 8 databitar, ingen paritet, 1 stoppbit.
- Innan du startar ett program på ISAK-kuben skall du starta ditt terminalprogram och sedan trycka på resetknappen på kuben.
- **Mycket viktigt:** Om du har använt HyperTerminal och sedan vill ladda ner ett nytt program till ISAK-kuben med den seriella nedladdaren måste du först rensa den befintliga anslutningen med HyperTerminalen ('Koppla från' på HyperTerminalen). I annat fall behåller HyperTerminalen kontrollen över COM-porten och den seriella nedladdaren kommer inte åt den. Du får ett meningslöst felmeddelande om att den seriella nedladdaren inte kan hitta ADuC812. I



detta fall måste du avbryta felmeddelandet, frigöra anslutningen i HyperTerminalen och sedan starta om nedladdaren.

- Kommunikationen mellan mikrokontrollerns kärna och kringkretsarna på chipet använder SFR (Special Function Register) av 8051-typ (figur 1). Ytterligare information om de individuella bitarna i SFR kan hittas i databladet för ADuC812, eller i 8051-böcker.

D/A-omvandlare

Skicka ut olika signaler:

Med hjälp av programmet `dac_1.c` kan två olika signaler (en sågtag och en fyrkant) skickas ut med hjälp av D/A-omvandlaren med två olika upplösningar (8 bitar och 12 bitar). Anslut ett oscilloskop till de relevanta utgångarna för att verifiera resultaten.

Den referensspänning som används här är den digitala matningsspän-

ningen V_{DD} ($= +5\text{ V}$), och de producerade utgångsspänningarna ligger i området 0 till $+5\text{ V}$.

A/D-omvandlare

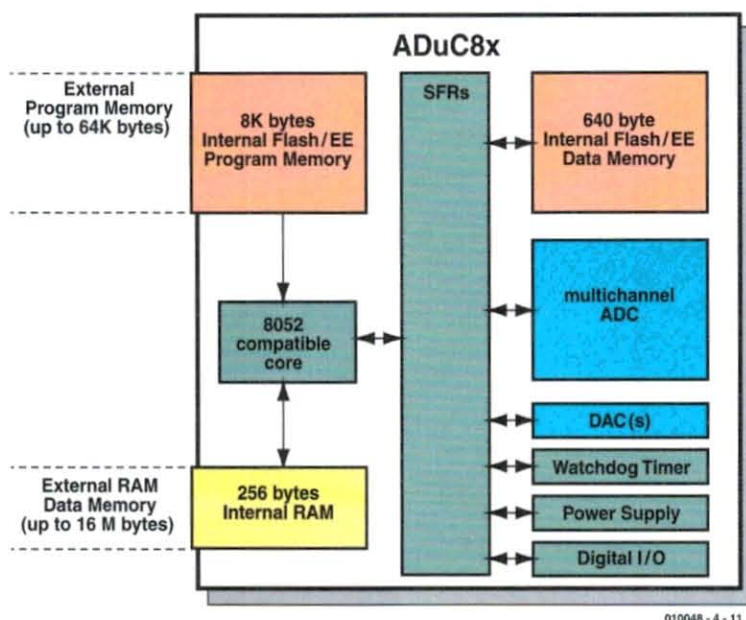
Läsning av spänningar på ingångskanalerna:

När du startar programmet `adc_1.c` blir du uppmanad att uppges det antal kanaler som skall läsas, och du har följande möjligheter:

- nummer 0-7: externa mätkanalerna ADC 0 till ADC 7.
- nummer 8: kanal ansluten till den inbyggda temperatursensorn (se nästa exempel).

området $+2.3\text{ V}$ till $+5.0\text{ V}$, ansluten till V_{Ref} benet. Med tanke på 12-bitarsupplösningen skall denna spänning filtreras och stabiliseras. Om, t.ex., V_{Ref} är kopplad till AV_{DD} matningen ($+5\text{ V}$), motsvarar 1 LSB ett spänningssteg på 1.2 mV .

Den ingångsspänning som skall mätas måste **alltid** ligga mellan 0 V och den referensspänning som används (antingen intern eller extern), dvs. i området 0 till $+2.5\text{ V}$ med den interna referensspänningen eller i området 0 till $+5\text{ V}$ om en extern referens på $+5\text{ V}$ används.



Figur 1. SFR-registret.

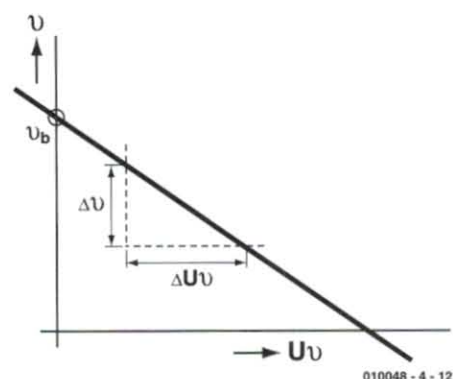
När du har angett en siffra omvandlas den spänning som finns vid den valda ingångskanalen en gång per sekund och resultatet visas som ett decimalt och ett hexadecimalt värde på skärmen. Läsningar stoppas med resetknappen och en ny ingångskanal kan sedan väljas.

Notera: A/D-omvandlarens upplösning är på 12 bitar och kan arbeta med två olika referensspänningar:

1. Med den interna 2.5 V referensspänningen. I detta fall motsvarar 1 LSB ett spänningssteg på $610\text{ }\mu\text{V}$.
2. Med en extern referensspänning i

Mäta den interna matristemperaturen och den omgivande temperaturen:

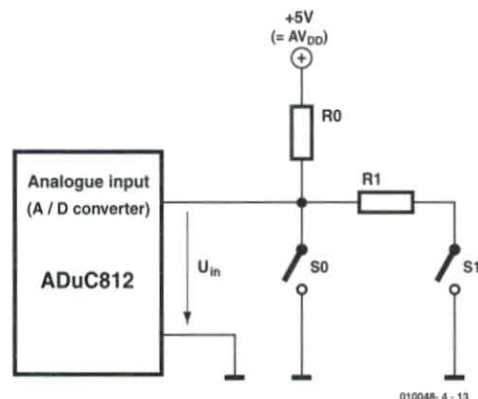
ADuC812 har en inbyggd temperatursensor som producerar en temperaturberoende spänning som är kopplad till den nionde kanalen (dvs. kanal 8) hos A/D-omvandlaren. Data-bladet ger följande information om denna sensor: Utgångsspänningen är 600 mV vid en temperatur på $\vartheta = 25^\circ\text{C}$, och temperaturberoendet är $-3\text{ mV}/^\circ\text{C}$. Om temperaturen stiger med 1°C kommer den temperaturberoende spänningen att falla



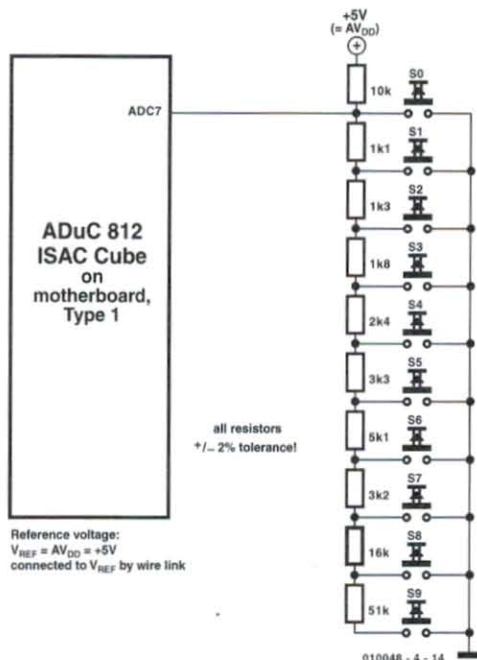
Figur 2. Det linjär förhållandet mellan spänningen U_0 och temperaturen ϑ .

med 3 mV och tvärtom. Men det finns en **mycket viktig punkt** att notera här: Dessa siffror är endast **typiska värden** och varierar från komponent till komponent. Om du jämför ADuC812 från olika produktionssatser hittar du ganska olika värden. Analog Devices garanterar dock att den temperaturberoende spänningen är mycket linjär i matristemperaturen.

Detta förhållande karaktäriseras av en (linjär) ekvation $U_0 = f_1(\vartheta)$ (or $\vartheta = f_2(U_0)$), figur 2. För användaren betyder detta att två kalibreringsavläsningar måste göras för att bestämma parametrarna för den linjära ekvationen innan några meningsfulla temperaturvärden kan erhållas. Om du då har en ADuC812 från samma produktionssats (vilket vanligtvis är fallet om de kommer från samma rör) behöver ingenting ändras. Om du emellertid köper ADuC812 från olika satser måste du upprepa denna kalibrering flera gånger. I värsta fall måste olika parameterinställningar för det linjär förhållandet erhållas för varje ADuC812. Programmet `temp812.c` har utvecklats för att hjälpa till med beräkningen. När programmet laddats ner hittar du



Figur 3. Spänningsdelaren vid ingången till A/D-omvandlaren switchas med tryckknappar.



Figur 4. En serie tryckknappar anslutna till ett analogt ingångsben.

också instruktioner hur du bestämmer parametrarna för det linjär förhållandet på **measuring the ambient temperature**, och hur du kommer åt de inställningar som programmerats in i temp812.c. Följ instruktionerna om hur programmet skall användas.

Tryckknappar anslutna till en analog ingång:

Grunden för denna metod för anslutning av tryckknappar, vilket gör att ett helt tangentbord kan använda ett analogt ingångsben på A/D-omvandlaren, är en 'multiply switched voltage divider, eller flerfaldigt switchad spänningsdelare'. Vi börjar med en enkel spänningsdelare, figur 3.

Med denna krets har vi:

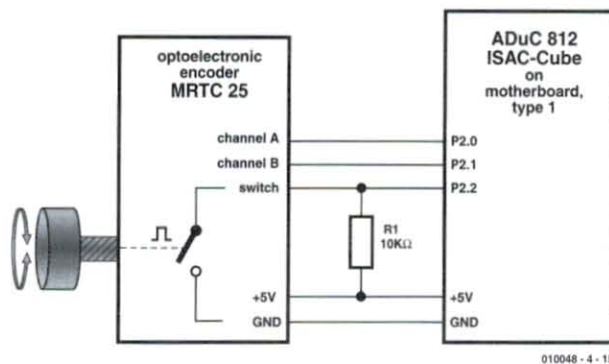
Ingen knapp intryckt: $U_{in} = 5\text{ V}$

Knapp 0 intryckt: $U_{in} = 0\text{ V}$

Knapp 1 intryckt: $U_{in} = 5 [R1/(R0+R1)]\text{ V}$

Om de två motstånd R1 och R0 har samma värde, t.ex. $1\text{ k}\Omega$, har vi $U_{in} = 2.5\text{ V}$.

För varje knapp som trycks in kommer en annan, skiljbar, spänning på den analoga ingången och omvandlas av A/D-omvandlaren. För varje knapp som trycks in erhåller vi på detta sätt ett annat digitalt värde och därför kan vi identifiera vilken knapp som tryckts in. Det finns ytterligare en fördel med denna krets. Om mer än en knapp trycks in samtidigt är det bara knappen närmast A/D-omvandlaren som räknas. Vi kan med andra ord bygga ett tangentbord där de olika tangen-



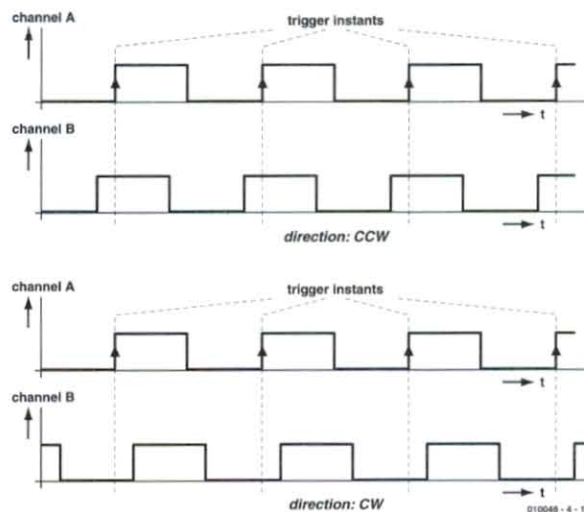
Figur 5. Inkoppling av en opto-elektronisk shaft encoder.

Tabell I

Egenskaper hos MRTC25 opto-elektronisk shaft encoder

Elektriska egenskaper:

Upplösning (pulser/360°)	25
Kanaler	2 (A och B; 90° fasoffset)
Matningsspänning	5 V DC $\pm 0.5\text{ V}$
Strömförbrukning	20 mA
Utgångssignaler	TTL-nivå fyrkant; intern $10\text{ k}\Omega$ pull-up
Tryckknapp	5 V, 10 mA (icke-induktiv last)
Arbetsliv	> 1,000,000 cykler



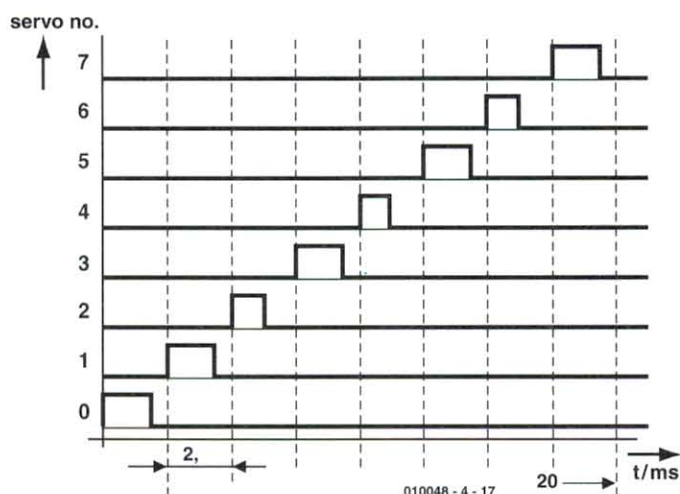
Figur 6. Timingdiagram för de två rotationsriktningarna.

terna prioriteras. Om, t.ex., knapp 1 redan är intryckt och knapp 0 därefter trycks in kommer U_{in} att direkt dras till den spänning som motsvarar knapp 0 (0 V). Knapp 1 körs alltså över och erkänns inte längre eftersom knapp 0 har den högsta prioritet och alltid kommer att erkännas.

För att kunna använda ett stort an-

tal knappar krävs motstånd med snäv tolerans, såväl som en välstabiliserad matningsspänning till spänningsdelaren. Figur 4 visar en praktisk krets för ett tangentbord med 10 knappar.

Vi skall inte bortse från en liten nackdel med denna idé. Knapparna kan naturligtvis bara läsas när de trycks in, det finns ingen automatisk



Figur 7. Timingdiagram för styrning av ett servo.

tangentbordsbuffert. Detta kan man dock lösa med hjälp av mjukvara. Ett lämpligt program för att läsa ett 10-knappars tangentbord anslutet till det analoga ingångsbenet ADC7 på ADuC812 finns i **ana_tast.c**. När du laddat ner detta program hittar du också en tabell som visar förhållandet mellan de intryckta knapparna, ingångsspänningen U_{in} och det konverterade digitala värdet, säkert läsintervall för knapparna och knapparnas prioritering (0=högsta prioritet).

Digital I/O-port

Att använda en opto-elektronisk

shaft encoder (pulsgivare):

En optoelektronisk shaft encoder är en elegant och enkel ingångskomponent. Här använder vi en MRT25 från Megatron, som har två utgångskanaler (A och B) för räkning och riktningvisning (medurs/moturs). Den har också en kontakt som sluts när axeln trycks in.

Anslutning till ISAK-kuben eller till det enkla moderkortet (typ 1) utförs såsom visas i figur 5. Timingdiagrammet för medurs och moturs rotation visas i figur 6. Elektriska egenskaper för shaft encodern visas i tabell 1. Programmet **drehge_1.c** vis-

ar hur du arbetar med denna intressanta ingångskomponent.

Styrning av åtta servon:

Med hjälp av åtta digitala utgångar hos port 2 och lite mjukvara, i detta fall skriven i assembler, är det lätt att kontrollera åtta servon. För varje servo måste en styrepuls med en längd mellan 0,9 ms och 2,1 ms genereras var 20 ms, figur 7. Den enklaste lösningen är att dela 20 ms perioden i åtta delintervall och sedan driva varje servo i tur och ordning från den lämpliga utgången hos port 2 till det önskade läget. Programmet **servo.asm** genererar intervallen på 2,5 ms med Timer 2 i autoreload mod och genererar de behövliga servopuls längderna med Timer 0 som en 16-bitars timer.

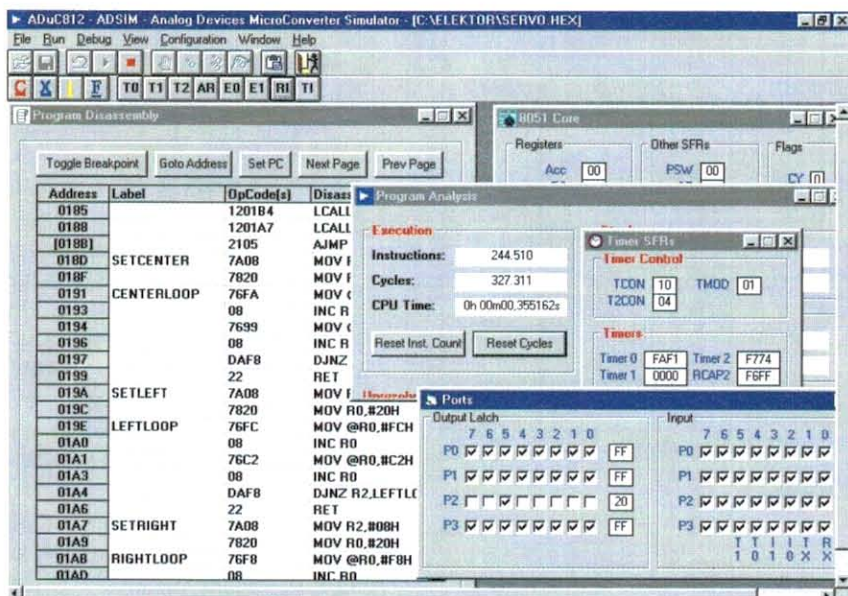
Intervall: $20 \text{ ms} / 8 = 2.5 \text{ ms}$ (Timer 2)

Servopulsbredd: 0.9 ms till 2.1 ms (Timer 0)

I interruptservicerutinen för Timer 2 startas Timer 0 med den pulsbredd som behövs för servot och den lämpliga pinnen på port 2 sätts. I interruptservicerutinen för Timer 0 rensas alla port 2 pinnar och timern stoppas. För att uppnå exakt timing i huvudprogrammet används en extra mjukvarufördröjning i Timer 2 interruptservicerutin. Detta korta program har testats med ADuC812 simulatören, figur 8.

Applikationer för ADuC812 begränsas endast av din egen fantasi. Med lite utbyggd mjukvara kan upp till 24 servon styras med de 24 digitala utgångarna. När det används i kombination med I²C interfacet kan vi bygga en ersättare för 'I²C servointerface' (Allt om Elektronik, 8/2001) — och vår ersättning erbjuder en upplösning på mer än 1000 steg.

Servona kan också styras av sensorer via de åtta analoga ingångarna, vilket är exakt vad denna serie (ISAK, Intelligent Sensor/Aktiva-tor Kontroller) handlar om.



Figur 8. servo.hex i ADuC812 simulatören (pulser skickas för servo 5 på ben P2.5).

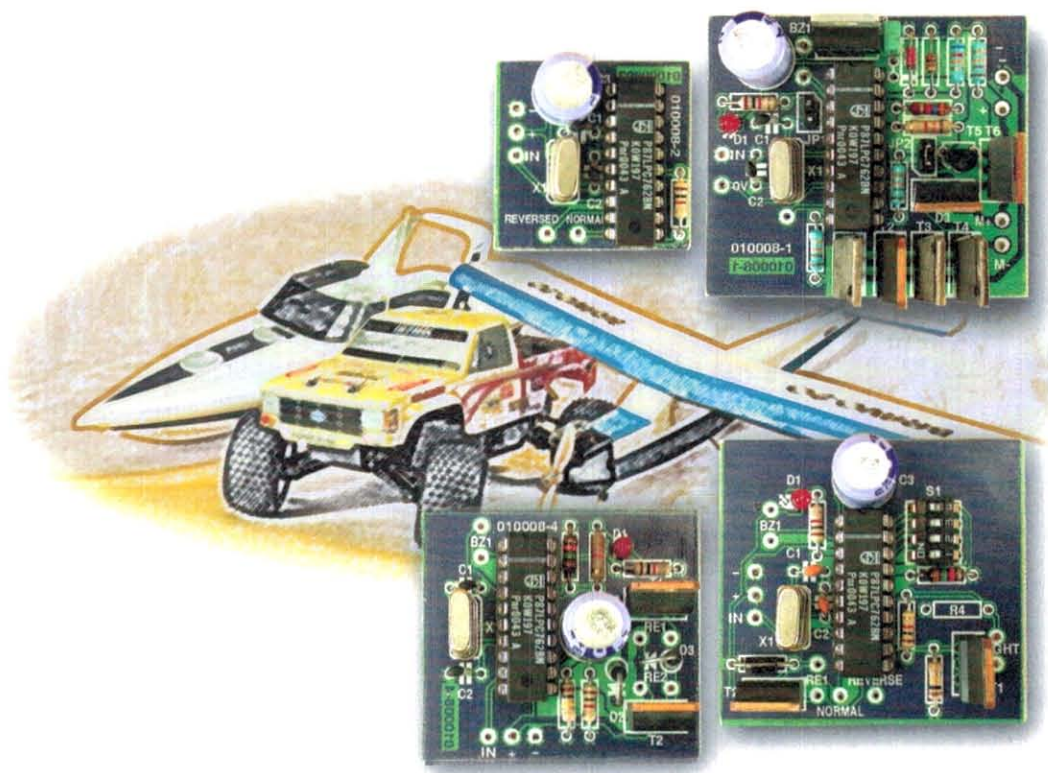
Komponentkälla:

MEGATRON Bauelemente,
Hermann-Oberth-Strasse 7,
D-85640 Putzbrunn, München, Tyskland.
Tel.: +49 89 46094 146

Mångsidig IC för modellbyggare

Del 2: två kretskort och nio funktioner

Den mångsidiga IC-kretsen för modellbyggare, som erbjuder totalt fjorton funktioner, kan verkligen stå för sig själv med de två kretskort som vi presenterar här. Dessa kort ger fartkontroll, mjukstart, övervakning av motorspänning, BEC, antikollisionsljus, idiotsäkerhet (fail-safe), hot glow, servobalansering och go-slow funktioner.



I föregående artikel i denna serie presenterades den mångsidiga IC-kretsen för modellbyggare med två applikationskort. I denna andra, och sista, artikel presenterar vi ytterligare

två kort. Totalt erbjuder dessa fyra kretskort fjorton olika funktioner och alla använder samma IC, en 87LPC 762 mikrokontroller från Philips, som

har programmerats speciellt för detta projekt. De två nya applikationskorten är 'motor-kontroll' och ett kort för go-slow/hot glow'.

En översikt av de fyra korten och de funktioner de erbjuder visas i tabell 1.

Övervakning av motorspänning

I fjärrstyrda flygplan och fordon används normalt batterier för både drivningen och, samtidigt via ett BEC-system (batterieliminatorkrets), för att strömförsörja mottagaren. Eftersom inte batterierna får laddas ur för mycket, och då BEC kräver en viss lägsta ingångsspänning, är det vettigt att övervaka batterispänningen och vidta lämpliga åtgärder när denna når en kritisk nivå.

Systemet måste designas på så sätt att dessa funktioner är tydliga för användaren och att de inte stör operationen av modellen. Som en permanent optisk indikering är en LED ansluten till portpinne

Tabell I

Kretskort och funktionsöversikt

Kretskort 010008-1, 'fartkontroll',

Allt om Elektronik 4/2002:

Fartkontroll, mjukstart, BEC, modellfinnare, spänningsövervakare för mottagare, motorspänningsövervakare

Kretskort 010008-2, 'servo reverse',

Allt om Elektronik 3/2002:

Servo reverse, Y-kabelersättning, fail-safe, servofilter

Kretskort 010008-3, "hot glow/go-slow",

Allt om Elektronik 4/2002:

Servo reverse, Y-kabelersättning, fail-safe, servofilter, anti-kollisionsljus, go-slow, skevroderbalansering, hot glow, spänningsövervakare för mottagare, modellfinnare

Kretskort 010008-4, "2-kanalswitch",

Allt om Elektronik 3/2002:

Fail-safe, spänningsövervakare för mottagare, modellfinnare, 2-kanalswitch

P1.6. Tar man med i beräkningen att motorspänningsövervakningen endast kommer att användas tillsammans med mjukstarten eller fartkontrollfunktionerna i kretsen i figur 1 kan vi också få motorn att 'hacka', en

indikation som är mycket tydlig för alla modellbyggare. Detta tillstånd kan man koppla bort permanent genom att kort koppla bort motorn och sedan koppla på den igen med en liten knuff på styrspeken. Detta

ser till att vi kan utföra en säker landning med full motoreffekt.

Ovanstående aktioner sker när mikrokontrollern upptäcker att spänningen är för låg tio gånger i rad, uppmätt med intervaller på 20 ms. Om du inte vill ha spänningsövervakningsfunktionen skall du inte montera motståndet R2 på portpinne P0.2. Zenerdioden **måste** monteras på kretskortet (se figur 2). Notera också att zenerdioden skall vara en 0,25-W typ.

De behövliga värdena på motstånden för att erhålla den önskade tröskelspänningen U_S kan beräknas med hjälp av formeln för spänningsdelning. Den inre referensen, som spänningen på port P0.2 jämförs med, är 1,23 V. Om, till exempel, R3 är satt till 10 k Ω , får vi för R2:

$$R2 = 10 \text{ k}\Omega * (U_S - 1.23 \text{ V}) / 1.23 \text{ V}$$

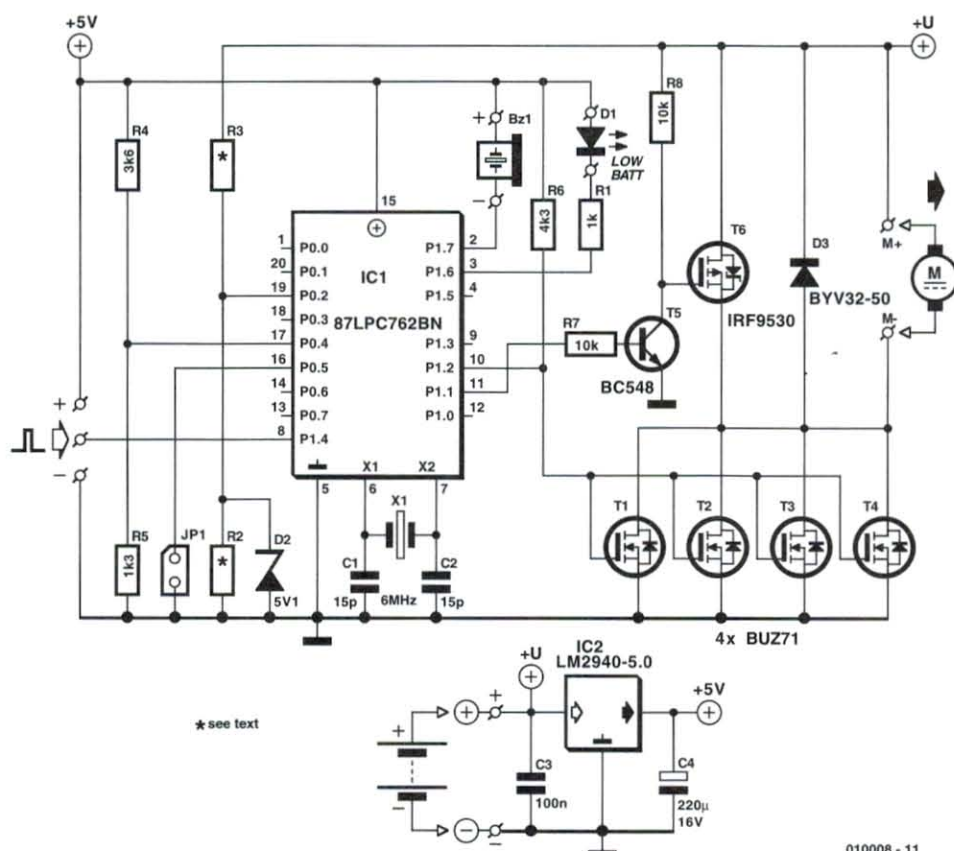
Till exempel: 8 celler, slutspänning 8.8 V:

$$R2 = 10 \text{ k}\Omega * (8.8 \text{ V} - 1.23 \text{ V}) / 1.23 \text{ V}$$

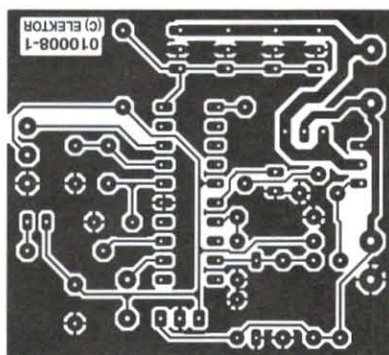
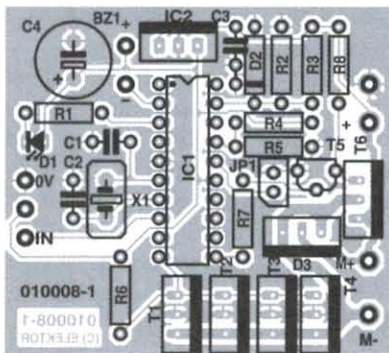
$$R2 = 61.54 \text{ k}\Omega \text{ (närmaste värde: 62 k}\Omega\text{)}$$

BEC-systemet

En BEC (Battery Eliminator Circuit) är ett system som ersätter mottagarens batteri med en matning som tas från huvudbatteriet. BEC i



Figur 1. Kretsschemat för motorstyrningskretsen.



Figur 2. Kretskortet för motorstyrning.

KOMPONENTLISTA

Kretskort 010008-1 (Motorstyrning, Fig. 2)

Motstånd:

- R1 = 1kΩ
- R2, R3 = se text
- R4 = 3kΩ26
- R5 = 1kΩ3
- R6 = 3kΩ26
- R7 = 10kΩ
- R8 = 10kΩ

Kondensatorer:

- C1, C2 = 15pF
- C3 = 100nF
- C4 = 220μF 16V radial

Halvledare:

- D1 = LED, röd, lågström
- D2 = zenerdiod 5V

D3 = BYV32-50

IC1 = 87LPC762BN (programmerad, ordernr **010008-41**)*

IC2 = LM2940-5.0

T1-T4 = BUZ71/IRLZ34N

T5 = BC548

T6 = IRF9530

Övrigt:

BZ1 = DC-summer, 5V

X1 = 6MHz kristall

2-pol stiftlist med bygel

Kretskort **010008-1***

* För programmerad mikrocontroller, kretskort och diskett med mjukvara, se Läsaerservice eller vår websida www.alltomelektronik.com

$$P = (U_{\text{Bat}} - U_{\text{BEC}}) I_{\text{BEC}}$$

Denna effekt omvandlas helt till värme. Ju större spänningsskillnad och ju högre ström desto större är effektförlusten och den avgivna värmen. Detta kan leda till en kraftig temperaturhöjning i IC-kretsen om du inte ser till att avleda denna, till exempel med ett kylelement. Ett kylelement rekommenderas också för spänningsregulatorn för att avleda värme om du använder högeffektsservon. I motsats till tidigare BEC-designer som använde en 7805-regulator kan vi med den låga drop-out-spänningen hos LM2940-regulatorn arbeta med en ingångsspänning ner till 6 V. De huvudsakliga tekniska egenskaperna hos LM2940 är:

- 1 V drop-outsänning
- Max utström = 1 A vid $T_J = 25^\circ\text{C}$
- Skydd mot omvänd polaritet
- Maximal ingångsspänning 26 V

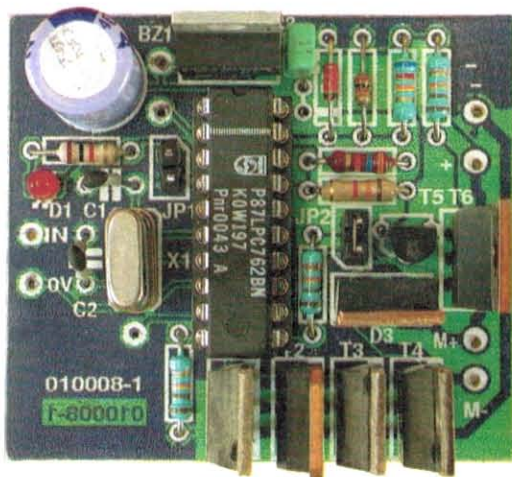
Fartkontroll

Moderna fartkontroller för modellplan är små, lätta, kraftfulla och drivs av en mikrocontroller. Den kontrollalgoritm som är implementerad i den mångsidiga IC-kretsen ger en mycket effektiv styrning av hastigheten hos DC-motorer med strömmar upp till ca 40 A med hjälp av pulsbredsmulering. Motorströmmen switchas av effekt-MOSFETarna T1-T4, vilka delar på lasten och switchar

med en frekvens på ca 2 kHz. Dioden D3 skyddar transistorerna från spänningsspicar när motorn switchas från.

När motorn kopplas från kan rotationen av propellern i luftströmmen kraftigt påverka modellens glidegenskaper. Av detta skäl finns det bromsfunktion implementerad, som använder T6 för att kortsluta motorn. Denna broms kan kopplas från genom att hålla styrspaken i läge 'fullt till' med sändaren påkopplad men mottagaren fränkopplad och sedan koppla på mottagaren. Helikopterpiloter skall vara medvetna om denna funktion, som låter rotorbladen snurra fritt. En inbyggd säkerhetsfunktion ser till att motorn inte kopplas på omedelbart om styrspaken inte från början är i frånläge, innan motorn kan startas måste styrspaken först föras till frånläget. Området för accepterade pulsbredder från mottagaren är 1,2 ms till 1,9 ms.

Om batteriövervakningsfunktionen används och spänningen sjunker under det förinställda minimivärdet börjar motorn att hacka rytmiskt. Detta hackande varnar piloten för att batteriet håller på att ta slut. Detta hackande kan du stoppa genom att kort stänga av motorn och sedan sätta på den igen med styrspaken. Eftersom den mottagna pulsbredden mäts med en mikrosekunds noggrannhet är också upplösningen hos mjukstarten och fartkontrollen en mikrosekund. Detta ger en mycket



Prototypen (skiljer sig lite från slutversionen).

kretsen i figur 1 består av lågfällsregulatorn IC2, som kan lämna en stabiliserad utgång på 5 V för mottagare från ett lämpligt huvudbatteri. Om denna funktion inte behövs behöver du inte montera IC2 på kretskortet (figur 2) och strömförsörjningen till kretsen dras direkt till den punkt som är markerad '+U'.

Om IC2 är monterad finns det en effektförlust som beror på skillnaden mellan huvudbatteriets spänning och spänningen vid BEC-utgången samt BEC-utgångens ström:

fin kontroll som många andra kontroller med grövre PWM-kvantisering inte kan matcha.

Mjukstart

Switchen för mjukstarten är konstruerad på samma sätt som fartkontrollen. Den enda skillnaden är att motorn automatiskt ökas upp till full fart under en tid på 2 s. Switchnings-tröskeln sätts när den mångsidiga IC-kretsen switchas till. Pulsbreddsmätningen vid denna punkt, plus extra 400 μ s (som praktiskt motsvarar ett utslag hos styrspaken på halva dessa område) sätter in det värde över vilket motorn kommer att starta. Denna säkerhetsfunktion ser till att motorn inte startas oavsiktligt när sändaren slås på med styrspaken i maxposition. Bromsfunktionen kan också avaktiveras på samma sätt som beskrivs ovan för fartkontrollen.

Hot glow

När mjukstarts eller fartkontrollsfunktionerna aktiveras i hot glow/goslow kretsen som visas i figur 3 (med det kretskort som visas i figur 4) lämnar portpinne P1.0 en signal som switchar för att visa när längden på den puls som kommer från mottagaren är kortare än 1,2 ms. Utgången

switchas till när kontrollinställningen befinner sig inom det givna området och switchas från när den lämnar detta. En intressant applikation för denna funktion är automatisk switchning av matningen till glödstiften i fyrtaktsmotorer, vilka har betydligt jämnare tomgång när glödstiftet är inkopplat. Om vikten på modellen inte är kritisk är det i allmänhet bättre att montera ett batteri för matningen till glödstiftet. Då behöver du inte hålla på med kontakten till glödstiftet, vilket kan vara farligt om detta ligger nära propellern. MOSFET T2 kan enkelt hantera strömmar upp till ca 10 A.

Anti-kollisionsljus

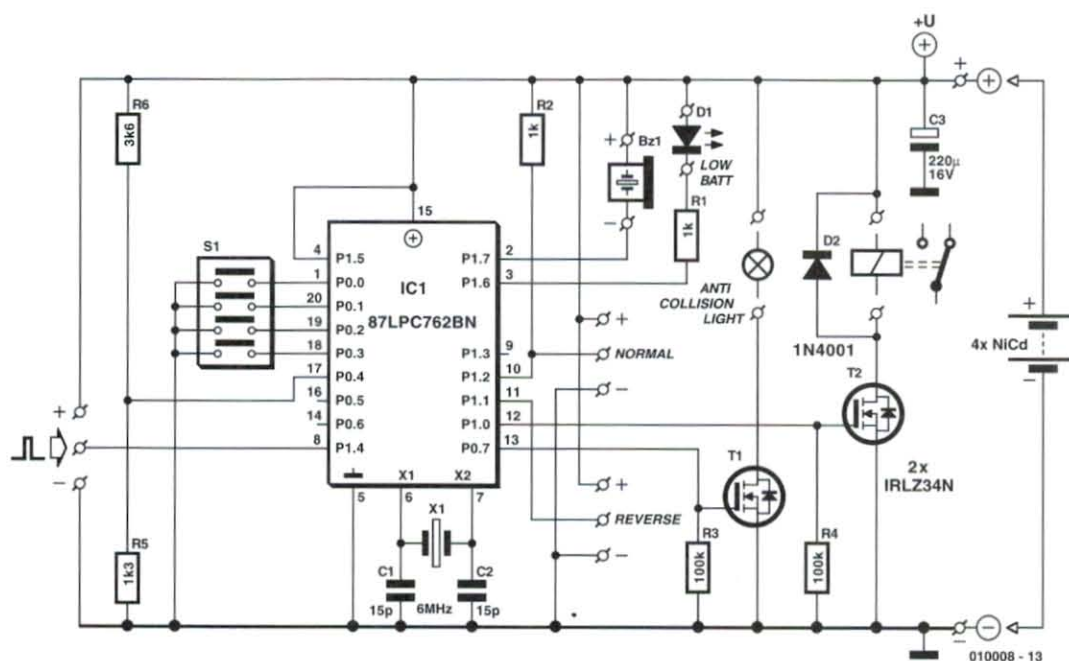
Ljuseffekter är slående och viss belysning gör modellerna intressanta. Mikrokontrollernas mjukvara genererar en signal på portpinne P0.7 som kan användas för anti-kollisionsljus (ACL). Signalen driver ljusen via effektristor T1, som kan switcha strömmar upp till 10 A utan problem. Blinkhastigheten beror på pulsrepetitionshastigheten vid mottagar servos signalutgångar och kan således variera något från tillverkare till tillverkare. Vid en frekvens på 50 Hz, till exempel, kommer ljusen att switchas till under 0,5 s och från under 1,5 s.

Som vi redan sagt är det viktigt att övervaka mottagarens matningsspänning för modellens 'överlevnad'. För att visa när huvud eller extrabatteriet nästan är tomt modifieras blinkhastigheten hos anti-kollisionsljusen i matningsspänningens larmkrets så att problemet kan upptäckas även på långt håll. Blinkhastigheten ändras till 4 Hz och håller denna frekvens tills mikrokontrollern nollställs och matningsspänningen återställs till ett säkert värde.

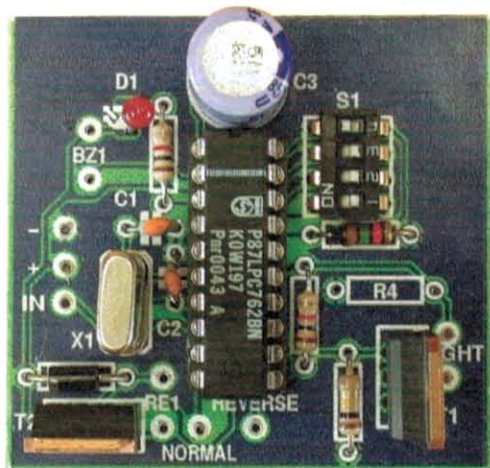
Balansering av skevningsroder

'Skevrodersbalansering' (aileron balancing) refererar till olika utslag uppåt och nedåt av skevroderen. Målet är att kompensera för det så kallade negativa momentet (tendensen för modellen att falla ur svängen). Modeller med ett stort vingspann och symmetriskt utslag på skevroderen utsätts för olika drag på de två vingarna, vilket tenderar att gira modellen runt den vertikala axeln mot svängriktningen. Detta kompenseras normalt med ökat roderutslag, vilket dock slösar effekt. En bättre lösning är att ombalansera skevroderen (figur 5), vilka normalt är positionerade antingen 100% uppåt eller 50% till 70% nedåt. Dessa värden varierar beroende på modellens geometri.

Graden av balanseringseffekt ställs in med DIP-switchen S1 på kretskortet. De tre switcharna S1-1 till S1-3 ställer in ett värde i åtta steg (om 10%) mellan 20% och 90%. Den fjärde switchen, S1-4 väljer mellan skevroderbalansering (switchen till) och go-slow funk-



Figur 3. Kretsschemat för 'hot glow/go-slow'.



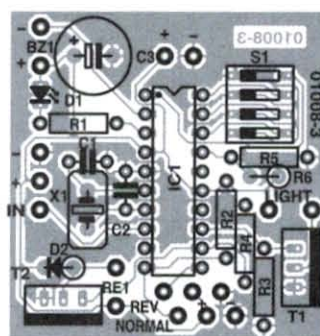
Prototypen (skiljer sig lite från slutversionen).

tionen (switchen från). Tabell 2 visar förhållandet mellan switchens inställningar och det differensvärde för skevrodet som erhålls. Balanseringseffekten fungerar bara ovanför neutalläget, dvs när pulsbredden hos signalen från mottagaren ligger inom området 1,5 ms till 2 ms. Båda servoutgångarna påverkas av denna funktion, reverseutgången påverkas när pulsbredden ligger inom området 1m5 ms ner till 1 ms. Detta betyder att två servon som arbetar i motsatt riktning kan användas även om bara en mottagarutgång används.

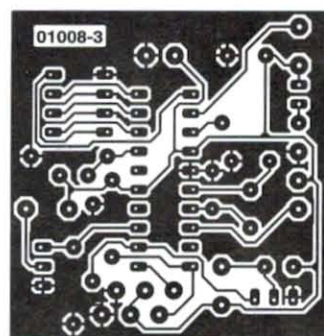
Go-slow

Moderna servon är pålitliga, kraftfulla och snabba. Fart kan, å ena sidan, vara mycket önskvärd, men å andra sidan finns det tillfällena när inte servot kan röra sig tillräckligt sakta. Om man, till exempel, flyttar vingklaffarna till landningsläge snabbt kan modellen stegra sig. En mycket snabb rörelse hos landningsstället verkar också orealistiskt. Naturligtvis kan varje servo styras för sig med en separat kontroll på sändaren, men det kan vara bra att ha en kontroll mindre att hålla koll på under start och landning. Det är viktigt att notera att go-slow funktionen inte fungerar ihop med specialservon som bara kan gå från det ena ytterläget till det andra. Dessa servon kan inte styras proportionerligt och kan därför inte användas med denna funktion.

Vad servot anbelangar så är go-slow enheten bara en svart låda som sitter mellan mottagaren och servo, mäter de inkommande pulserna och sedan antingen ökar eller minskar längden på utgångspulserna tills servot hamnar i det läge som motsvarar den nya pulsbredden vid mottagaren. Kretsen simulerar en gradvis rörelse hos styrspeken på sän-



Figur 4. Kretskortet för 'hot glow/go-slow'.



KOMPONENTLISTA

kRETSKORT 010008-3 (hot glow/go-slow, Fig. 4)

Motstånd:

R1,R2 = 1kΩ
R3,R4 = 100kΩ
R5 = 1kΩ3
R6 = 3kΩ26

Kondensatorer:

C1,C2 = 15pF
C3 = 220μF 16V radial

Halvledare:

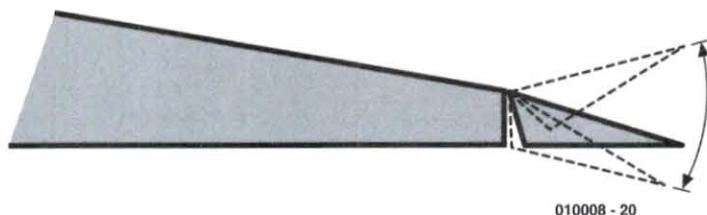
D1 = LED, röd, lågström
D2 = 1N4001

IC1 = 87LPC762BN (ordernr
010008-41)*
T1,T2 = BUZ71/IRLZ34N

Miscellaneous:

Bz1 = DC buzzer, 5 V
S1 = 4-way DIP switch
X1 = 6MHz quartz crystal
PCB, order code 010008-3*

* För programmerad mikrokontroller, kretskort och diskett med mjukvara, se Läsaerservice eller vår websida www.alltomelektronik.com



Figur 5. Principen för skevroderbalansering.

Tabell 2

Förhållande mellan switchinställningar och skevroderbalans

Switchinställningar				Skillnad
1	2	3	4	
on	on	on	on	20%
off	on	on	on	30%
on	off	on	on	40%
off	off	on	on	50%
on	on	off	on	60%
off	on	off	on	70%
on	off	off	on	80%
off	off	off	on	90%
off	off	off	off	0% ('go slow' funktion med nollfördröjning)

Tabell 3

Förhållande mellan switchinställningar och fördröjning

Switchinställningar				fördröjning (s)
1	2	3	4	
on	off	off	off	1
off	on	off	off	2
on	on	off	off	3
off	off	on	off	4
on	off	on	off	5
off	on	on	off	6
on	on	on	off	10
off	off	off	off	noll fördröjning

Notera: switch 1 till switch 4 är kopplade till portpinnarna P0.0 till P0.3
on = switch sluten = portpinsingången ser jordnivå
off = switch öppen = portpinne dras upp av internt motstånd

daren, och hastigheten kan väljas i sju steg. Förutom de applikationer vi nämnde ovan, landingsställ och vingklaffar, finns det också andra möjligheter som får folk att titta lite extra på din modell, till exempel öppna huven över cockpit eller att öppna en dörr. Förhållandet mellan switchens inställningar och servots hastighet visas i tabell 3 och där switchen S1-4 aktiverar go-slow funktionen (när switchen är i läge 'off').

Med go-slow funktionen kan två servon också kopplas på så sätt att de arbetar i motsatta riktningar men med samma hastighet. Notera att de givna tiderna bara är ungefärliga värden och är beroende på en 25 Hz pulsrepetition hos mottagaren. Här kan det skilja sig åt mellan tillverkarna.

Fail-safe (idiotsäkerhet)

Fail-safefunktionen, som hittills endast funnits tillgänglig för fjärrstyrningssystem med PCM, tvingar in servona i fördefinierade positioner om den mottagna signalen skulle utsättas för kraftiga störningar. Om en modell kan landa säkert med rodet låst i mittläge är kanske tveksamt, men att stänga av motorn och elektriska funktioner är ett säkert sätt att minimera eventuella skador. Så kan, till exempel, en motor som låst sig på full matningsspänning lätt förstöra drivelektroniken med för hög ström. Det är också förnuftigare att föra tillbaka rodet i mittläge i händelse av störningar, än att låta det ligga kvar med fullt utslag. Fail-safe funktionen aktiveras när störningarna på den mottagna signalen är så kraftiga att mikrokontrollern inte längre kan avkoda en giltig signal eller

när det inte finns någon signal alls (eftersom modellen ligger utanför sändarens räckvidd eller om sändaren stängts av). Mikrokontrollern kommer då att:

1. aktivera summern;
2. avaktivera alla switchutgångar;
3. när det gäller mjukstarten och fartkontrollen, stänga av motorn;
4. driva båda servoutgångarna med en pulsbredd på 1.5 ms.

Dessa förhållande kommer att kvarstå ända tills det finns en giltig signal tillgänglig igen.

MOSFET

Slutligen några ord om de MOSFET som används. Moderna exemplar av dessa transistorer har praktiskt taget förlustfri switchning och en låg 'till' resistens. Det är dock viktigt att se till att styret (gate) alltid drivs fullt så att transistorn inte arbetar inom det resistiva området av sin karaktäristik. Eftersom det endast finns 5 V tillgängligt för att driva transistorerna måste vi använda typer med så kallade 'logiska nivåer'. Med dessa kan höga strömmar switchas utan att de blir för varma. De IRLZ34N från International Rectifier som vi föreslår finns lätt tillgängliga och har bra förhållande pris/prestanda. De har följande egenskaper:

- I_D : 27 A
- V_{DS} : 55 V
- P_D : 56 W
- $R_{DS(on)}$: 0.035 Ω

Modellbyggare som vill krama den sista droppen prestanda ur sina system kan överväga att använda SUP75N03-04, vilka, med en $R_{DS(on)}$ på 0.004 Ω , ger ett ännu lägre spänningsfall.

(010008-2)

Internetadresser

LM2940:

<http://www.national.com/pdf/LM/LM2940.html>

N-channel MOSFET:

<http://www.irf.com/product-info/datasheets/data/irlz34n.pdf>

P-channel MOSFET:

<http://www.irf.com/product-info/datasheets/data/irf9530.pdf>

BYV32-50 :

<http://www.mobilesemi.com/us/products/fer.asp>

SUP75N03 :

http://www.nessel-elektronik.de/FET_Bauteile/fet_bauteile.html

8051-kompilerare

freeware, shareware och demoversioner

Av Harry Baggen

8051-an, och dess många kompatibla derivat, är den mest använda mikrokontrollern inom industrin. En av fördelarna med att arbeta med denna mikrokontroller är den enorma mängden information angående associerad hårdvara och mjukvara. Internet är utan tvekan den största kunskapskällan för användare av 8051-derivat. I denna artikel fokuserar vi på kompillerare för 8051-an.

De gamla bekantingarna 8051 och 8052 kan tacka sin ryktbarhet för att de är så lätta att programmera via den världsberömda BASIC-52 tolken (som *Allt om Elektronik* var en av de första tidningarna att beskriva och använda i praktiska projekt). Idag producerar halvledartillverkare kontroller vars kärna baseras på 8051-arkitekturen, även om dagens chip-teknik är betydligt snabbare och mer utbyggd än originalet, som introducerades för många år sedan. Den utbyggda arkitekturen kan inkludera mer minne, en watchdog, ett LCD-interface, D/A och A/D-omvandlare och flera andra element.

Den 89S8252 som du hittar på det utvecklingskort som vi använder i vår grundkurs i mikrokontrollerteknik är ett bra exempel på ett avancerat 8051-derivat.

För många applikationer är BASIC-52 helt enkelt för långsam och det uppenbara alternativet är att skriva koden i assembler. Det kan vara uppenbart, men det är inte enkelt. En kompillerare representerar ett mellanalternativ som översätter ett antal 'läsliga' kommandon till maskinkod som kan kopieras in i mikrokontrollern.

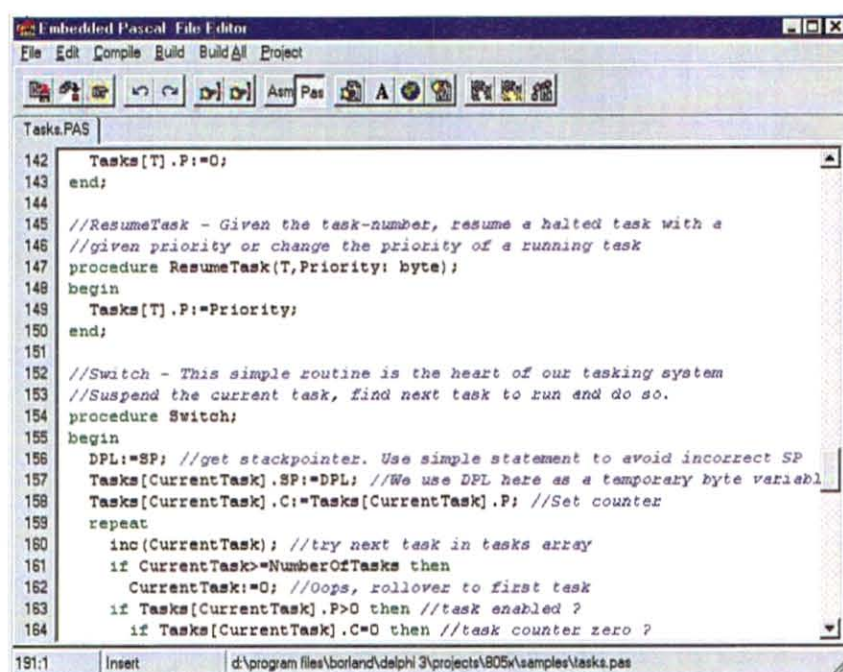
Även om det finns ett stort antal 8051-kompillerare på marknaden så kostar de flesta av dem runt en tusenlapp. Lite väl häftigt tycker vi om du bara är nyfiken, eller bara programmerar lite då och då. Nu finns det tack och lov även gratiskompillerare och demoversioner, vilka är begränsade när det gäller storleken på programkoden. Dessa program är perfekta om du begränsar dig själv till relativt små

program. Och tro oss, du upphör aldrig att förvånas över kraften hos intelligent designad maskinkod på bara 1 eller 2 kByte.

En av de mest kända gratis C-kompillerarna för 8051 kallas **SDCC** [1] vilken är en förkortning för *Small Device C Compiler*, även om vissa källor hävdar att det står för *Sandeep Dutta's C Compiler* efter namnet på den ursprungliga utvecklaren. SDCC

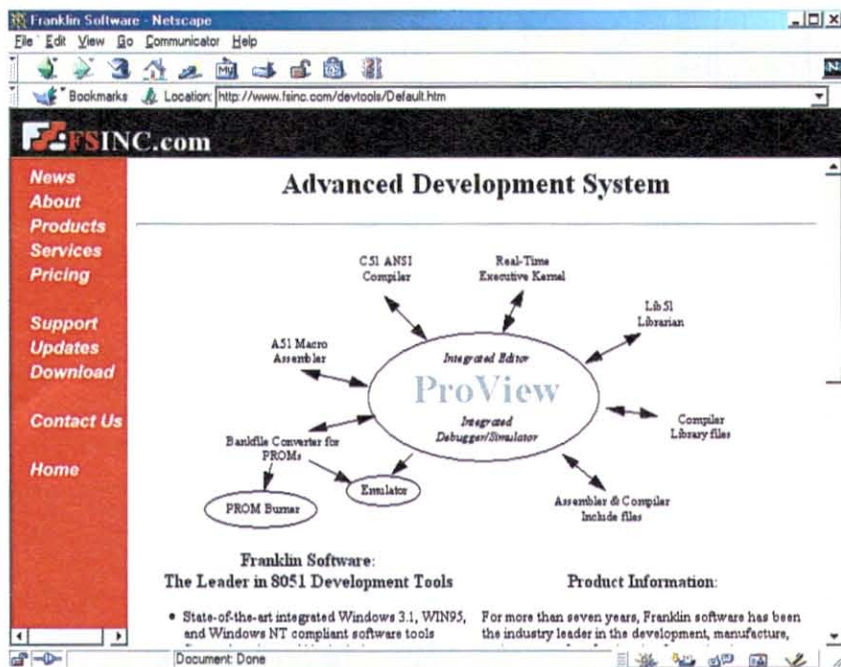
är en kod som optimerar ANSI-C kompillerare för 8051 och Z80 mikrokontroller. Versioner håller också på att utvecklas för andra populära mikrokontroller. SDCC finns tillgänglig för Windows och Linux. Programmet håller hela tiden på att utvecklas av ett antal programmerare.

En annan kompillerare som är gratis (vilket vi redan nämnt i Mikrokontrollerkursen) är **Reads51** from Rigel



```

Embedded Pascal File Editor
File Edit Compile Build BuildAll Project
[Icons] Asm Pas [Icons]
Tasks.PAS
142   Tasks[T].P:=0;
143 end;
144
145 //ResumeTask - Given the task-number, resume a halted task with a
146 //given priority or change the priority of a running task
147 procedure ResumeTask(T:Priority; byte);
148 begin
149   Tasks[T].P:=Priority;
150 end;
151
152 //Switch - This simple routine is the heart of our tasking system
153 //Suspend the current task, find next task to run and do so.
154 procedure Switch;
155 begin
156   DPL:=SP; //get stackpointer. Use simple statement to avoid incorrect SP
157   Tasks[CurrentTask].SP:=DPL; //We use DPL here as a temporary byte variable
158   Tasks[CurrentTask].C:=Tasks[CurrentTask].P; //Set counter
159   repeat
160     inc(CurrentTask); //try next task in tasks array
161     if CurrentTask=NumberOfTasks then
162       CurrentTask:=0; //oops, rollover to first task
163     if Tasks[CurrentTask].P>0 then //task enabled?
164       if Tasks[CurrentTask].C=0 then //task counter zero?
165
191:1 Insert d:\program files\borland\delphi 3\projects\8051\samples\tasks.pas
  
```



Corp. [2]. Även om denna kompilator egentligen är avsedd för Rigels egna mikrokontrollerkort kan *Allt om Elektronik* läsare ladda ner det gratis för privat användning tillsammans med Grundkursen i mikrokontrollerteknik. Reads51 är en kombination av en small-C kompatibel 8051-kompilator, en relativassembler, linker/locator, redigerare, simulator samt avbuggare/monitor. Ett fantastiskt mjukvarupaket som man vänjer sig vid ganska fort. Om du känner dig mer hemma med

Pascal än med programmeringsspråket C kan du prova med **Pascal51**. Detta populära program har vi redan nämnt i samband med ett antal projekt i *Allt om Elektronik*. Även om de olika downloadplatserna för detta program (som från början utvecklades av en rysk programmerare) har växlar en hel del under senare tid, kan vi nämna två platser som verkar relativt pålitliga [3]. Derek Kennedy har tagit fram några expansioner för Pascal51, kallade Prepo, vilka kan hittas på sam-

ma sida.

Embedded Pascal för 805x [4] är en Pascalkompilator som har ett 'device definition system' som gör att kompilatorn kan anpassas till i stort sett vilket 8051-derivat som helst på marknaden. Embedded Pascal är ett sharewareprogram med vilket man kan producera objektкод med begränsad storlek.

Nu när vi har kommit över till shareware och demoversioner är det läge att återvända till C-kompilatorer. En användbar C-kompilator finns i utvärderingsversionen av **C51 Tools** från Keil [5]. Kompilatorn har en kodstorleksbegränsning på 1 kByte. Tyvärr tillåter demoversionen inte att programmet startar vid 0000H, vilket betyder att det inte kan användas för single-chip kontroller.

Franklin Software har en omfattande programmeringsmiljö för 8051, med **Proview** redigerare/avbuggare/simulator i centrum. Man erbjuder också en C51 ANSI kompilator (version 6). Programpaketet kommer som ett utvärderingspaket med en storlek på 12 MB [5]. Begränsningen är storleken på objektkodsfilen, 4 kByte.

Raisonance har ett antal programmeringsverktyg för olika mikrokontroller, inklusive 8051, Philips XA och ST6-serierna. En demoversion för 8051-an finns tillgänglig, max kodstorlek är 4 kByte [7].

Som du kan se finns det en hel del att välja på för er som vill ge er i kast med 8051-kompilatorer och många av dem är gratis eller utvärderingsversioner. Så det är bara att börja tanka hem.

(025019-1)



Internetadresser:

- [1] SDCC: <http://sdcc.sourceforge.net>
- [2] READS51: www.rigelcorp.com/reads51.htm
- [3] Pascal51: www.eedevl.com/files.html
Pascal51, Prepo51: www.geocities.com/SiliconValley/Campus/9592/
- [4] Embedded Pascal: <http://users.iafrica.com/r/ra/rainier/p8052.htm>
- [5] Keil C51 evaluation tools: www.keil.com/demo/
- [6] Franklin Development Tools: www.fsinc.com/devtools/Default.htm
- [7] Raisonance 8051-compiler: www.raisonance.com/download/index.php