BYGGE

BYGG EGET KODLÅS!

Detta är ett elektronikbygge för dig som behöver ett enkelt men tillförlitligt kodlås. Det kan användas för att styra en låskolv, för att aktivera och stänga av ett elektroniskt system och det larmar direkt vid felaktig kod.

Av Nils-Åke Petersson

Det är nog så att varje ändamål kräver sin lösning på kodlås. Induktiva nycklar, radiosändare eller IR-sändare utan fysisk kontakt med kodmottagaren är praktiska och betydligt säkrare än till exempel kodlås med knappsats.

Men färdiga system är inte billiga och som användare vet man mycket lite om funktionen. Man måste helt enkelt lita på att det fungerar och är säkert. Man kan oftast inte själv snabbt ändra koden om tex en nyckel kommer bort. Trots avancerade kodkretsar är de inte utan säkerhetsluckor.

Ett hembyggt kodlås har knappast någon inkräktare lärt sig i förväg.

Till och med om man vet hur nyckeln ser ut och hur larmet är konstruerat är ett kodlås svårforcerat om det, som detta bygge, endast tillåter ett enda försök. Det här kodlåset har prövats under flera år och dessutom stått emot sabotageförsök vid ett inbrott.

Det är en synnerligen enkel konstruktion. Det använder en billig nyckel och det är lätt att ändra koden vid behov. Från början var tanken att det skulle ersättas av ett mer avancerat men det behovet uppstod aldrig eftersom det visade sig fungera utmärkt.

Så här fungerar kodlåset

Kodlåset består i själva verket av tre delar: nyckel, "sändare" och mottagare. Sändaren placeras åtkomlig för nyckeln och mottagaren placeras skyddad, ihop med själva larmet eller den enhet som kodlåset skall styra.

Själva nyckeln består av ett byglingsfält som motsvarar ett antal ettor och nollor, maximalt 12. Jag har använt en 9-polig D-subkontakt som nyckel men det finns andra möjligheter.

Sändaren skickar data motsvarande den kod, dvs ettor och nollor, som nycken byglats till. Via kabel når datat mottagaren. Inget hindrar att överföringen i stället sker med IR-ljus, ultraljud eller radio men i mitt fall var kabel det bästa.

Om strömmen av data motsvarar den programmerade koden i mottagaren ger denna en klarsignal. Mottagaren kan ge larm direkt, om en felaktig kodnyckel används. Ett försök med felaktig nyckel räcker.

UM3750

Det finns en hel del kodkretsar och jag har valt en vanlig typ: UM3750 från UMC. Det är en 18-bens IC om kan fungera antingen ➤

som mottagare eller sändare. Det är en MOS/LSI-krets som arbetar med 3 till 11V och drar lite ström. Byglad som sändare skickar den data som består av 12 bitar (tar ca 11,5 ms), ett uppehåll, samma 12 bitar igen, uppehåll osv. Frekvens dvs bithastigheten bestäms av den oscillator som finns i kretsen. Ett motstånd **R** och en kondensator **C** bestämmer frekvensen enligt formeln **f= 2 / R * C**.

Tillverkaren anger 100 kHz som typfrekvens vilket man kommer nära med R = 100 kohm och C = 180 pF. Jag har dock valt en lägre frekvens. R = 100 kohm och C = 470 pF ger f = 2/R * C = 43 kHz. Mottagaren arbetar asynkront och dess frekvens bör ligga så nära "sändarens" som möjligt men det verkar inte alltför kritiskt.

Med ett oscilloscope kan man se hur bitströmmen ser ut **(figur 1)**:

Synkronisering av varje databit sker på föregående puls bakflank. I figuren ser du att pausen avslutas med en puls (0,50 ms). Dennas bakflank är startsignalen (synk) för identifiering av värdet på den första biten. Antingen följer ett pulsuppehåll på 0,50 ms (+puls 2 x 50 ms) vilket står för en 0:a (som i figur 1) eller ett pulsuppehåll på 2 * 0,50 ms (+ puls 50 ms) vilket står för en 1:a.

Därefter kommer nästa bit, i det här fallet en 1:a! Detta är en typ av pulskodmodulering (PCM):

*Mottagaren jämför de 12 bitarna med sin inställda kod (som du bestämt). Om koden stämmer **fyra gånger i rad** betraktas den som godkänd och då går kretsens utgång låg. Endast så länge det fortsätter att komma korrekta 12-bitars ord (med max 256 ms uppehåll) förblir utgången låg.*

Sändaren

Låt oss nu gå över till schemat för sändaren **(figur 2)**.

Till **VCC** (**U1:18**) kopplas positiv matningsspänning. Ingången **U1:15** som kallas MODE kopplas till **VCC** och gör att kretsen fungerar som sändare. R_1 och C_1 bestämmer frekvensen. Dataingångarna **D0 - D11** har intern pull-up och ligger därför höga om de inte kopplas låga det vill säga till 0V. När **VSS** (**U1:14**) kopplas till **0V** (**GND**) kommer kretsen att börja skicka PCM data på utgången TX/OUT. Detta är det hela!

Om du utnyttjar alla dataingångarna finns det 4096 olika byglingsmöjligheter. Den första koden får du om du lämnar alla ingångarna orörda. Då skickas data 1111 1111 1111. Om du byglar **D0** till **VSS** (**0V**) skickas i stället 1111 1111 1110 osv.

I schemat finns en lysdiod D_2 som lyser när sändaren kopplas in.

Lysdioden D_1 är en återkoppling från mottagaren som indikerar att nyckeln fungerar.

En praktisk nyckel

Den kod som skall skickas kopplas alltså in på sändarens dataingångar.

I mitt fall består själva nyckeln av en byglad 9-polig D-subkontakt (hylsdon).

En bygel är absolut nödvändig! Den skall sitta mellan **P1:5** och **P1:1**.

Om du studerar schemat (figur 1) ser du att denna bygel kopplar **VSS** till **GND** (**P3:4**). Detta ger spänning till sändaren. Så här kan en nyckel se ut **(figur 3)**:

Denna kommer att skicka koden 1111 1011 1110.

De första fem bitarna 1111 1 finns inte på nyckel och utnyttjas inte (obyglade på kortet). I själva verket utnyttjas i det här fallet endast 7 bitar vilket ger 128 möjliga koder.

Mottagaren ger endast en chans till rätt kod men om du tycker 7 bitar ger för svagt skydd kan du använda en 15-polig D-sub istället och koppla alla 12 dataingångarna till nyckeln.

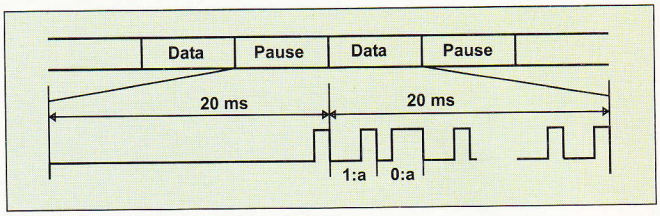

Fig 1. Pulskodmodulerat data.

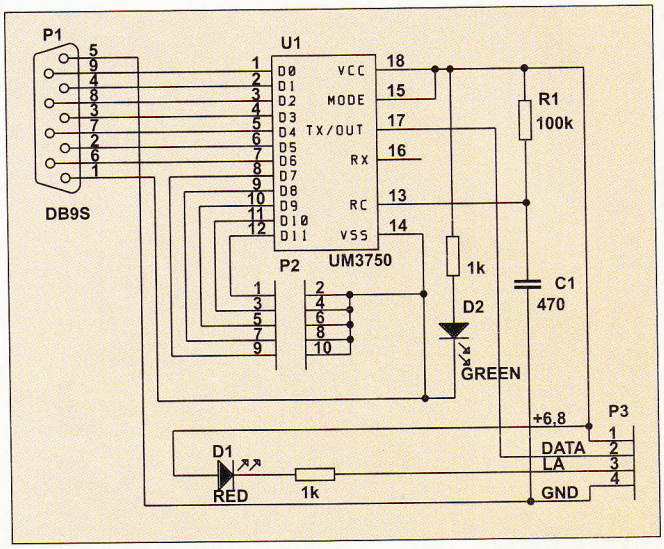

Fig 2. Schema på kodsändaren.

Mottagaren

I sin enklaste form ser mottagaren ut på detta sättet **(figur 4)**:

Samma typ av krets, UM3750, används även i mottagaren. Nu har dock ingång **U2:15** kopplats till **0V** (**GND**) vilket gör att kretsen nu arbetar som mottagare. R_9 och C_2 bestämmer oscillatorfrekvensen. Till **D0 - D11** finns ett bygelfält där du byglar rätt kod dvs den enda dataström som kretsen skall godkänna. Här byglar du exakt samma som i sändaren.

Till nyckel ovan (figur 3) byglas **P6:1-2** och **13-14**. När rätt data tas emot kommer utgången **RX** (**U2:16**) att gå låg. Den ligger låg så länge rätt kod skickas och går hög om/när kodnyckeln dras ut.

Själv har jag nyckeln (D-sub kontakten) bland mina vanliga nycklar och vill därför att kodmottagaren skall stanna kvar i aktiverat eller deaktiverat tillstånd när jag drar ut nyckeln. Detta kan lösas med en enkel vippkoppling och jag gjorde en variant med hjälp av en räknare **(figur 5)**.

Kopplingen ser mer invecklad ut än den är! Utsignalen från kodmottagaren (**U2:17**) kopplas till **IN** (**IC2:15**) på räknaren 4516. Rätt kod ger låg utsignal men då händer inget!

Däremot när nyckeln dras ut och signalen **IN** återgår till hög stegar räknaren upp. Om utgången **IC2:6** tidigare varit låg går den nu hög. Om den tidigare var hög, går den låg. Varje gång jag skickar rätt kod till mottagaren kommer alltså utgång **IC2:6** att inverteras (skifta mellan 0:a och 1:a).

C_3 och R_8 kopplade till **IC2:9** har endast funktionen att säkerställa att utgången är låg från början när spänningen kopplas in.

Som du ser i schemat driver utgången (**IC2:6**) två transistorer T_3 och T_4. Aktiverad utgång (hög) gör att T_3 leder som då i sin tur ger ström till lysdioden i optokopplaren **U3**.

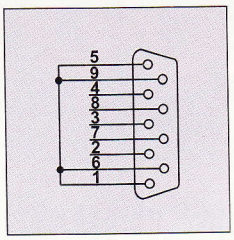

Fig 3. Exempel på bygling av nyckel.

Optokopplarens utgång **U3:4** är den återkopplingssignal (LA) till sändaren som jag nämnde ovan. som gör att D_1 (figur 2) lyser och visar att mottagaren är aktiverad. Optokopplaren skyddar elektroniken i mottagardelen och främst då **IC2** som inte får påverkas av vilken som helst koppling i sändardelen som ju är åtkomlig för sabotageförsök.

Transistorn T_4 ger den utsignal som används för att styra den apparat , låskolv eller vad det nu är som skall aktiveras av kodlåset. Denna kopplingen med BC337 klarar ett relä eller annat som drar max 40 mA.

Larma kodlåset

Så här långt är det inte alltför svårt att ta sig förbi kodlåset.

Lite kunskap om elektronik i allmänhet och om UM3750 i synnerhet räcker för att testa olika koder (byglingar) och förr eller

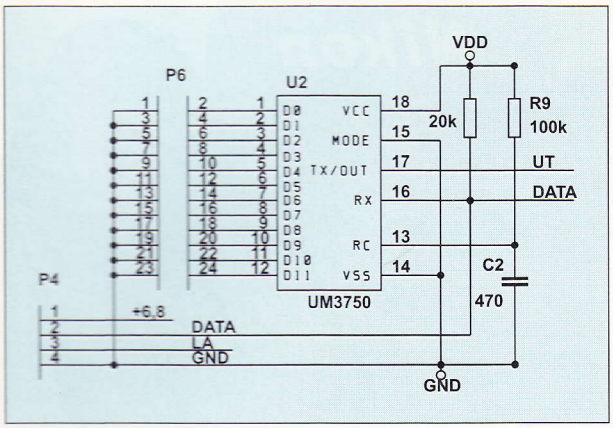

Fig 4. Kodmottagaren.

senare träffa rätt. I verkligheten är väl det mest sannolikt en skruvmejsel eller kofot som kommer till användning men faktum är att "mina inkräktare" försökte bygla in rätt kod genom att bocka ihop stiften och troligtvis kortsluta med skruvmejsel innan de slet ner hela kodlåset (sändardelen) och gav upp den delen.

Om kodlåset skall användas för att sätta på och stäng av ett larm bör man ha en sabotagebrytare som aktiverar larmet om låset slits ner från väggen och/eller bryts upp.

I vilket fall som helst, det skall inte vara möjligt att testa sig fram till rätt kod! Rätt kod direkt, annars larm! Detta löser man med elektronik i mottagardelen **(figur 6)**.

Principen är enkel men den förutsätter att sändaren aktiveras av kodnyckeln och inte ligger inkopplad hela tiden.

Räknarens ingång **IC1:10** kopplas in på dataströmmen från sändaren (som också går till **U2:16** se figur 4). Så fort nyckel stoppas i och sändaren börjar skicka data kommer **IC1** att räkna upp. Det vill säga om ingången **IC1:11** (resetingång) ligger låg.

Transistorn T_1 som styr denna ingång är kopplad till kodmottagarens utgång (**U2:17**) det vill säga den utgång som går låg när rätt kod skickas.

Funktionen blir därför denna: så länge inte rätt kod skickas kommer signalen på **IN** (från **U2:17**) att ligga hög. Det innebär att transistorn T_1 leder och resetingången **IC1:11** ligger låg vilket i sin tur innebär att räknaren räknar upp för varje puls som kommer från sändaren.

Efter 256 pulser kommer utgången **IC1:13** att gå hög , T_2 leder och detta betyder larm - någon skickar felaktig kod!

Om rätt kod skickas kommer det att gå iväg maximalt 65 pulser sedan kommer utgången på kodmottagaren att gå låg vilket gör att T_1 öppnar och räknaren **IC1** nollställs (reset) det vill säga en bra bit innan den kommer till 256. Något larm kommer inte att utlösas vid rätt kod.

Varför blir det just 65 pulser. Jo, mottagarkretsen kräver 4 korrekt mottagna ord och det innebär 4 x 13 pulser (se figur 1).

Dessutom är det ganska troligt att det första pulståget inte är rätt eftersom sändaren kopplas in samtidigt med koden. Minst 5 x 13 pulser = 65 således för att ge OK ut från mottagaren.

Om det däremot är fel kod blir det ingen nollställning utan räknaren når snabbt (under 0,5 sek) 256 pulser och därmed larm. Det är dessutom endast rätt kod som nollställer kretsen.

Vid snabba försök med felaktig kod (endast i teorin eller med avancerad utrustning) ackumuleras pulserna av räknaren.

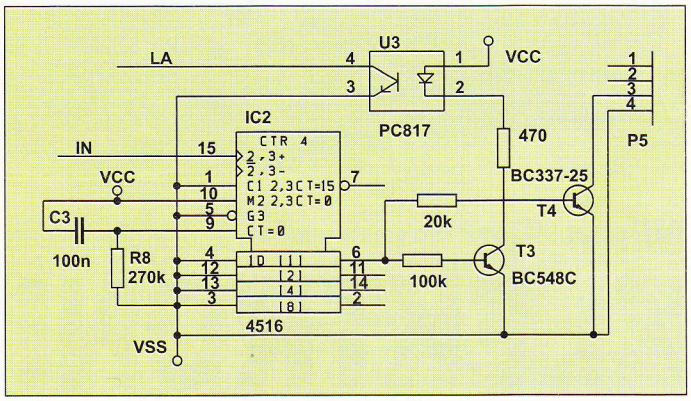

Fig 5. Vippkoppling i kodmottagaren.

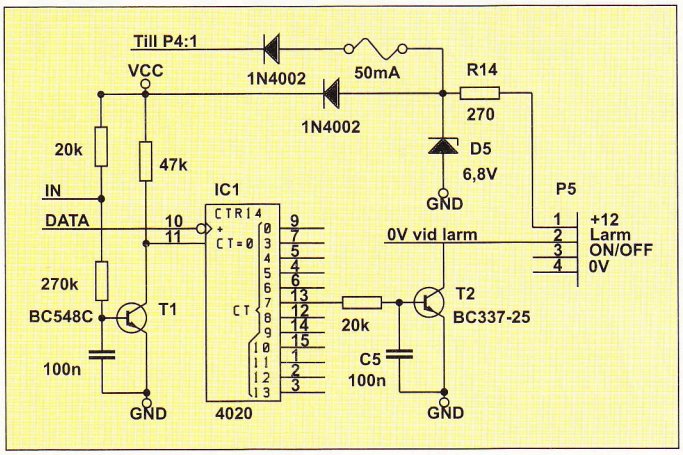

Fig 6. Larm vid felaktig kod.

En svaghet med den här tekniken skall dock noteras. Spänningsinkopplingen av sändaren och byglarna, som sker genom nyckeln, måste ske samtidigt och någorlunda distinkt.

Det är möjligt att få en puls på larmutgången om man vickar och trixar med nyckeln istället för att trycka in den. För att undertrycka denna möjlighet sker spänningsinkopplingen med hjälp av ytterstiften i nyckeln. Kondensatorn C_5 (se figur 4) har också tillkommit för att sätta P för snabba korta på larmutgången som har upptäckts vid trixande med rätt kodnyckel.

I schemat (figur 6) kan du lägga märke till zenerdioden D_5 på 6,8V. Denna ger den drivspänning som valts till kodlåset. De två andra dioderna och säkringens funktion är att skydda bakomliggande elektronik i händelse av sabotage i sändardelen.

Tips för användning

Det kontaktdon (**P1**) i sändaren som passar mot nyckeln (stiftkontakten) bör kunna bytas på enkelt sätt. Det är i regel mycket tunn guldplättering på dessa kontakter av billig typ och tillverkarna garanterar inte många in- och urjackningar. Jag bytte dock första gången först efter ca 2 års daglig användning!

Jag har inte monterat själva kontakten **P1** på kretskortet eftersom den mekaniska påfrestningen vid in- och ur-jackning av denna typ av kontakter är stor och riskerar att leda till foliebrott. Sätt istället kontakten på frontpanel och mjuk kabel ner till kortet.

Eftersom sändarens koppling är så pass enkel går det bra att bygga denna del på experimentkort. Samma gäller för övrigt mottagaren i dess grundutförande. Komplett mottagare med två räknare kan dock vara svårt att få tillförlitlig utan ett riktigt mönsterkort.

Jag har valt frekvensen ca 40 kHz i mitt kodlås. Både för att få säkrare funktion och med hänsyn till störningsproblematiken - EMC. Även 40 kHz är en tillräckligt hög frekvens för att kunna störa annan elektronisk utrustning.

Med riktigt konstruerade mönsterkort med jordplan, en skärmad kabel mellan sändare och mottagare (rekommenderas) och helst skärmade lådor för inbyggnad, är dock risken minimerad.

Bygg nu själv

Kodlåset finns att köpa som komplett komponentsats inklusive två mönsterkort, samtliga komponenter på kretskorten, 9-polig D-sub (nyckel) samt schema och kort byggbeskrivning. (245:-) Synpunkter och frågor som rör artikeln är välkomna. Skriv eller ring till författaren.

Belganet Dataelektronik , 370 12 HALLABRO. Tel 0457-52175 eller e-post: nilsake@bde.se. Eventuella korrigeringar, tillägg och tips på användning publiceras på hemsidan: **www.bde.se.**