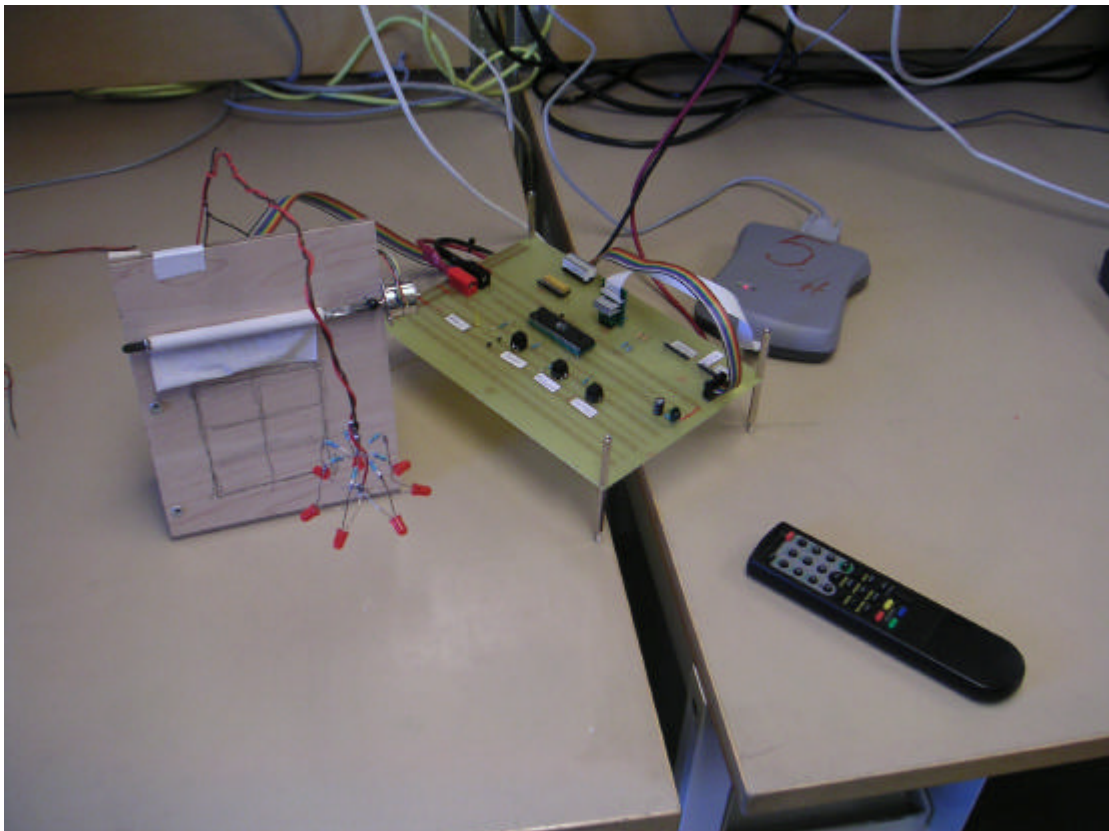




LUNDS TEKNISKA  
HÖGSKOLA  
Lunds universitet

# RemoteBud



Inlämnas: 2005-02-01

Patrik Johnsson, e01pjo  
Viktor Karlsson, e01vk

## **Abstract**

Skulle du också vilja styra dina lampor och rulla ner dina persienner med hjälp av din TV-fjärrkontroll? Remotebud är konstruktionen som låter dig styra diverse vardagsbestyr med din redan befintliga fjärrkontroll. Har du knappar på din fjärrkontroll som saknar funktion eller vill du kanske att dina persienner ska rullas ner när du slår på teven? I så fall har vi lösningen på ditt problem. Remotebud passar de flesta fjärrkontroller och är perfekt för dig som är en äkta latmask.

# Innehållsförteckning

Beskrivning av problemet .....	3
Utvecklingsprocessen.....	4
Hårdvaran.....	6
Mjukvaran.....	7
Förbättringar.....	8
Slutsats .....	9
Referenser .....	10
Bilaga 1 .....	11

## Beskrivning av problemet

Vi hade från början tänkt oss att konstruera något som på ett bra vis kan spela in en godtycklig signal från vilken fjärrkontroll som helst inom en viss standard. Den inspelade signalen ska sedan användas för att t.ex. slå på en lampa eller styra en stegmotor. Fokus på uppgiften ligger på att få fram en konstruktion som går att använda tillsammans med så många olika fjärrkontroller som möjligt. Hela syftet med vår konstruktion är att ge din befintliga fjärrkontroll nya användningsområden. Vi satte från början upp en kravspecifikation som såg ut så här:

Grundspecifikation:

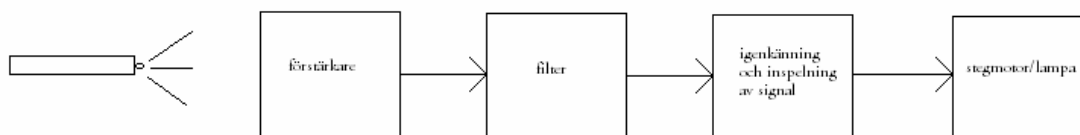
- Konstruktionen skall klara av att spela in godtycklig signal från en i förhand utvald fjärrkontroll. Naturligtvis är det någon form av IR-standard som kommer att användas.
- Signalen som spelats in används för att känna igen en likadan signal. När signalen dyker upp igen skall en utsignal genereras. Utsignalen ska kopplas till ett relä eller något liknande.

Specifikation för saker som implementeras i mån av tid:

- Möjlighet att kunna spela in flera olika signaler.
- Klara av mer än en standard.
- Kopplas till fler och mer avancerad applikationer. T.ex. dimmer eller stegmotor.

## Utvecklingsprocessen

Vi började med att försöka tänka ut ett blockschema som på ett bra vis beskrev uppgiften vi tänkt lösa. Efter en del funderande kom vi fram till blockschemat i figur 1.



figur 1. blockschema över problemet.

Vi började sedan fundera på hur vi skulle implementera vår konstruktion och vilka komponenter vi behövde. Eftersom vi redan från början hade våra aningar om att det skulle komma att behövas en processor gällde vårt första beslut val av processor. Vi ansåg att en AVR-processor var precis det vi behövde för inspelning och igenkänning av signalen och funderade vidare på de två första bitarna av blockschemat. Att bygga filter och förstärkare är ju inte vidare digitala uppgifter och framför allt inte det vi hoppades få syssla med i den här kursen men eftersom det är två mycket viktiga bitar för vår konstruktion är det inget man kan ignorera eller skjuta åt sidan. Dock visade det sig senare att det fanns en alldeles ypperlig komponent som löste bägge dessa problem mycket enkelt. IS1U60 är en komponent som tar emot, förstärker och filtrerar en signal från alla fjärrkontroller som har bärfrekvensen 38 kHz. Tack vare denna lilla komponent sparade vi mycket arbete. Efter ytterligare lite letande bland databladerna på kurshemsidan hittade vi en komponent till som kunde spara mycket arbete. En decoder från Philips som klarade av att koda av just den typen av signal som vi får som utsignal från vår mottagare/förstärkare. Man alltid har hört att Philips dominerar stora delar av TV-marknaden och att många olika andra TV-tillverkare använder sig av Philips standard. Visserligen skulle vi nu vara begränsade till Philips standard men om man tvunget måste välja en standard så varför inte ta den som är störst?

Nu kunde vi göra en enkel uppkoppling för att undersöka fjärrkontroller. Vi provade den ena fjärrkontrollen efter den andra och hur vi än försökte kunde vi inte få decodern att fungera på någon fjärrkontroll. Detta var en besvikelse eftersom vi trodde att många olika fjärrkontroller skulle passa bra. Vi kontrollerade mottagaren för säkerhets skull men det verkade inte vara något fel på den. Vi valde nu att försöka hitta information på internet om exakt vilka olika fjärrkontroller som kunde tänkas följa vår valda standard och då fann vi förklaringen till att vi inte lyckats. Standarden vi valt var gammal och används inte längre. Detta var ett bakslag och vi fick nu på allvar fundera ut ett bra sätt att spela in och jämföra olika signaler med varandra.

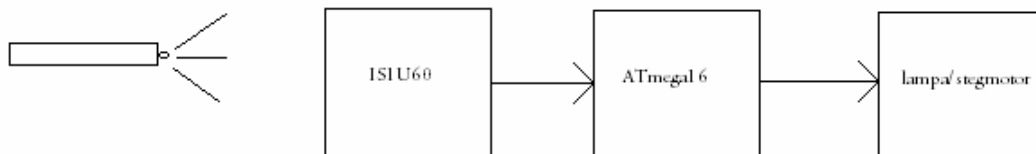
När vi funderat på problemet ett tag och diskuterat fram och tillbaka kom vi fram till att en ren mjukvarulösning av problemet var den bästa lösningen. Idén var att koppla signalen till ett av benen på processorn som används som inport. Därefter skulle vi generera ett periodiskt avbrott och i avbrottsrutinen läsa av om vi hade en etta eller

nolla på benet vid just den tidpunkten. Vi tänkte sedan räkna antalet ettor och nollor som kommer i följd och på det viset avgöra pulslängderna i signalen. Alla mätningar sparas sedan i en vektor vilket gör det lätt att jämföra två signaler med varandra. För att hålla reda på om en signal ska sparas som referenssignal eller om den ska jämföras med den befintliga referenssignalen fick vi införa en inspelningsknapp. När knappen är nertryckt vet vi att det är dags att spela in en ny referenssignal och om knappen inte är nedtryckt så jämför vi den inkommande signalen med referenssignalen. Det visade sig att det var just denna biten som var den svåraste delen av hela projektet. Vi fick lägga ner ganska många timmar på att få ordning på avbrottsrutinen och på att få vårt program att fungera felfritt men när vi löst detta problemet var det dags att gå vidare till stegmotorn.

Att få igång stegmotorn visade sig vara en betydligt lättare uppgift än vi trodde trots att vi fick leta en del på internet innan vi fick igång den. Eftersom det gick så pass smärtfritt att få igång stegmotorn fick vi lite tid över att utvidga vår konstruktion så att den har möjlighet att känna igen flera olika signaler.

## Hårdvaran

Remotebud består av relativt få komponenter men här i figur 2 kommer ett blockschema som visa hur komponenterna hänger ihop med varandra och efter det kommer en beskrivning av de intressanta komponenterna. Vill du se hela kopplingsschemat finns det bifogat i bilaga 1



figur 2. Blockschema över hårdvaran.

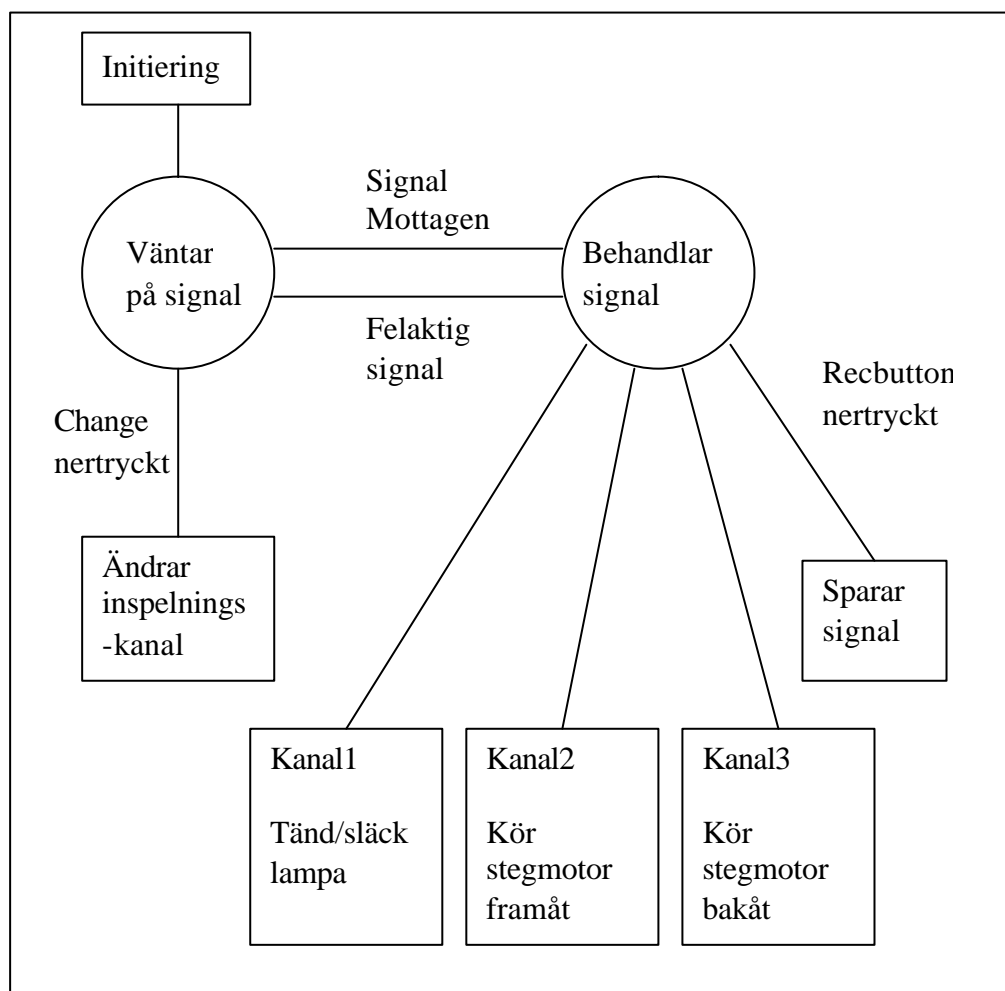
IS1U60: En mycket praktisk liten komponent som tar emot, förstärker och filtrerar signalen från vilken fjärrkontroll som helst med bärfrekvensen 38 kHz. Utsignalen från denna komponent är perfekt som insignal till en processor.

ATmega16: Processorn som vi använder för att utföra det mesta av arbetet. Denna processor innehåller det mesta och därför behövs väldigt få övriga komponenter i vår konstruktion. Processorn har 40 pinnar och en maximal klockfrekvens på 8 MHz. Stöd för både interna och externa avbrott finns naturligtvis också.

Sm40: Detta är en mycket liten stegmotor som vi använder i rent demonstrativt syfte. När vår konstruktion skall användas på riktigt måste man såklart ha en motor anpassad till ändamålet.

## Mjukvaran

Som sagt har vi valt att implementera det mesta av vår konstruktion i mjukvara. Vi har ju trots allt tillgång till en rätt kraftfull processor så varför inte använda den. I figur 3 visas en abstrakt modell av mjukvarans funktion. Efter initieringsfasen inväntas en signal eller knapptryckning. Om tex. change-knappen trycks ner så byts inspelningskanal och det visas med feedback lysdioden. När en signal tagits emot jämförs den och om den stämmer överens med någon inspelad signal görs den fördefinierade händelsen. Är istället inspelningsknappen nertryckt sparas signalen undan.



figur3. modell över hur mjukvaran är uppbyggd.



## Förbättringar

När man byggt en prototyp kommer man alltid på saker som kan förbättras eller som borde implementerats annorlunda. Vi anser att vår konstruktion kunde förbättrats på följande punkter:

- Den första och enklaste förbättringen är att lagra informationen i minnet så att den inte försvinner vid ett eventuellt strömavbrott.
- Genom att optimera avbrottstider och gränsvärden skulle man eventuellt få konstruktionen fungera tillsammans med fler fjärrkontroller.
- Ett lite större ingrepp vore att grundläggande ändra inläsningsmetod och istället för att generera ett periodiskt avbrott hela tiden vänta med detta tills det dyker upp en insignal. När det väl är en insignal på porten skulle man mycket väl kunna ha samma strategi som vi tidigare haft. Om man gör dessa ändringar får man en mer effektiv konstruktion och resurser över till eventuella andra uppgifter som skulle kunna implementeras.
- Att få användargränssnittet snabbare och enklare är också en sak att tänka på. Ett användargränssnitt som är både enkelt uppbyggt och lätt att förstå är viktigt för den här typen av produkter.
- Att byta processor till en lite mindre variant t.ex. Tiny AVR är ett sätt att få ner kostnaden på konstruktionen.
- Man kan även tänka sig att det kunde vara bra att försöka få ner effektförbrukningen så pass mycket att det räcker med ett batteri för att driva konstruktionen. Om man kan göra detta slipper man t.ex. ha en transformator och man har möjlighet att göra produkten riktigt liten och nätt.

## Slutsats

Kursen har ett format som är väldigt intressant. Nästan alla andra kurser består enbart av teori. Här får man gå från ide till prototyp och försöka använda sig av allt man har lärt sig. Kursen är uppbyggd på det viset att det inte finns någon övre gräns för hur mycket man får ut av den. Den blir helt och hållet vad man gör den till och ju mer tid man kan lägga ner på den ju mer intressant blir den. Kursen fungerar också som en bra värdemätare för hur mycket man hittills lärt sig på den här utbildningen och man får en liten hint om vilka områden man saknar kunskap inom och vilka områden man behärskar bra.

## Referenser

För att det skulle vara möjligt att bygga vår prototyp har vi naturligtvis fått leta efter en del fakta och här kommer våra källor.

1. Datablad till alla kretsar finns på :  
[http://www.it.lth.se/digp/datablad/datablad\\_index.asp](http://www.it.lth.se/digp/datablad/datablad_index.asp)
2. B.W. Kernighan , D.M. Ritchie, the C programming language second edition, 1988.
3. Information om Stegmotorn:  
<http://www.isk.kth.se/kursinfo/mekatronik/komponentkurs/klab/stegmotor/>

# Bilaga 1

