

# **TSIU51 Mikrodatorprojekt**

—

## **Projektkompendium Konstruktionsmetodik Projektkrav**

Micke Josefsson

23 januari 2020

# Innehåll

<b>1</b>	<b>Inledning</b>	<b>3</b>
1.1	Litteratur och komponenter . . . . .	3
1.2	Gruppindelning . . . . .	3
1.3	Lokaler . . . . .	3
1.4	Information under kursens gång . . . . .	3
1.5	Handledarens roll . . . . .	4
1.5.1	Ordningsregler för Muxen . . . . .	5
1.5.2	Skåp . . . . .	5
1.5.3	Tillgängliga komponenter . . . . .	5
1.5.4	Virning . . . . .	6
<b>2</b>	<b>Konstruktionsmetodik</b>	<b>8</b>
2.1	Allmänt . . . . .	8
2.2	Konstruktionstips . . . . .	8
2.2.1	Synkron klockning . . . . .	9
2.2.2	Avkoppling . . . . .	11
2.2.3	Slingor . . . . .	11
2.2.4	Läs databladen . . . . .	12
2.2.5	Fallgropar . . . . .	12
2.2.6	Felsökning . . . . .	12
2.2.7	Dokumentation och rapport . . . . .	13
<b>3</b>	<b>Examination</b>	<b>14</b>
3.1	Presentation . . . . .	14
3.2	Demonstration . . . . .	14
3.3	Rapport . . . . .	14
<b>4</b>	<b>Projektkrav</b>	<b>18</b>

# 1 Inledning

Allmän information om kursen och projektet. Läs detta.

## 1.1 Litteratur och komponenter

Som kurslitteratur i kursen används dels detta kompendium, dels av examinator angiven litteratur enligt kurshemsidan samt de datablad som återfinns på databladsservern *vanheden*, [vanheden.isy.liu.se](http://vanheden.isy.liu.se). Vissa udda komponenter utöver databladsservern finns dock. Datablad för dessa komponenter kan delas ut av handledaren beroende på projektets art. För utdelade dverktyg och komponenter att de måste vara återlämnade vid projektdelens slut.

## 1.2 Gruppindelning

Projektet utförs i grupper om fyra studenter. Gruppindelningen sköts av studenterna själva för att få grupper med gemensamt intresse för det valda projektet. Kontakta examinator om du är grupp-lös så kan grupper bildas även på detta sätt.

## 1.3 Lokaler

Projekten genomförs i ämnesområdet Datortekniks kurslaboratorier, kollektivt benämnda *Muxen*. Dessa är belägna i Datortekniks lokaler i B-huset, ingång 27, en trappa, C-korridoren, se skyltar. Varje grupp har ett nummer och ett förvaringsskåp med samma nummer.

Antalet labplatser är begränsade. Förvaringsskåpet är tänkt för förvaring av projektet när din grupp inte är i labbet. Är ditt projekt skrymmande kan det förvaras även ovanpå skådpen. Håll rent på labplatsen så blir det trevligare att komma dit. Städa bort virtrådsrester, kladdpapper, cola-burkar med mera.

## 1.4 Information under kursens gång

Inledande information om kursen delges deltagarna via kursinformationen och under föreläsningarna. Därefter meddelas allt av intresse via utskick via epost eller lisam (för TEMA-inslaget). För att få denna epost måste man vara anmäld på kursen!

## 1.5 Handledarens roll

Handledaren skall hjälpa till att styra in projektet på rätt nivå, och kan ge tips och goda råd angående konstruktionen. Komponenter som finns inlåsta lämnas ut av handledaren. Labplats och nycklar delas ut av och lämnas in till handledaren. Kursens syfte är att lära ut konstruktion och felsökning genom egna experiment. Handledaren skall inte behöva rycka ut varje gång något inte fungerar. Handledaren är inte någon gruppchef, utan en räddningsplanka att söka upp när idéerna tagit slut. I vissa fall kan man boka tid med handledaren i förväg. Det mesta labarbetet skall i allmänhet ske självständigt inom gruppen.

Om handledaren är svårartat oanträffbar, försök få tag i någon annan grupps handledare snarare än examinator. Handledarnas roll är just att avlasta examinator.

Den skriftliga rapporten lämnas direkt till examinator.

**Projektval** De flesta deltagare i dessa kurser har ingen eller liten vana av konstruktion av elektroniska apparater. Brist på erfarenhet är därför absolut ingen anledning till oro, grupperna snabbt lär sig snabbt. Fråga gärna andra labgrupper om något är oklart, och handledaren finns naturligtvis för att hjälpa till om man kör fast.

De som redan har en viss kunskap om konstruktion och programmering kan gärna åta sig något av de mer omfattande projekten. Det är meningen att man skall lära sig något, inte bara visa upp vad man redan kan.

Projekt där mikrokontrollers ingår tenderar att urarta till mer eller mindre renodlad programmering, vilket inte stämmer överens med kursens målsättning. Projektet måste hålla en håller en viss *lägsta nivå* för att få köras. Kursen betonar mikrokontrollernas roll som ”spindeln i nätet”, som hanterar flera in- och utsignaler. Ett minimikrav är att projektet baserar sig på två mikrokontrollers som löser uppgiften tillsammans, det brukar innebära en lagom nivå.

Missa inte möjligheten att använda analoga komponenter. Kopplingen till verkligheten måste alltid göras analog, vilket är lätt att glömma när man hackar kod som flitigast.

Ett urval projekt finns samlade i annat kapitel i detta kompendium. Svårighetsgraden varierar något mellan olika projekt, och det finns möjlighet att specificera funktionen på egen hand för att göra projekten lättare eller svårare.

Ett projektunderlag/kravspecifikation skall lämnas in till examinator och godkännas innan handledare tilldelas. Använd projektkraven som underlag för men formulera projektunderlaget med egna ord och tänk igenom vilka problem som kan (kommer!) att inträffa och hur dessa skall kunna lösas. Projektunderlaget skall lämnas in till och godkännas av examinator. Examinator har rätt att underkänna projektunderlag eller utöka alltför triviala projekt, och kan också ge en liten vink om projektet verkar för svårt. Det är viktigt att projektet kan slutföras på tilldelad tid och samtidigt hinna ge en insikt i både konstruktion *och* felsökning. Eftersläpning med projektet till nästföljande period eller utanför läsperiod är inte möjlig.

### 1.5.1 Ordningsregler för Muxen

För att kunna hantera det mycket stora antal teknologer som passerar Muxen varje år krävs att ordningen upprätthålls. Följande regler och sunt förnuft gäller:

- Eftersom komponenter, virsocklar och virkort återanvänds är det viktigt att *handskas försiktigt* med dessa. Det är till exempel absolut förbjudet att löda på annat än lödsocklar och dylikt.
- *Ställ tillbaka* utrustning och materiel som inte används på sin rätta plats. Skåpen i Muxen är märkta, ställ tillbaka på rätt plats.
- Gruppen ansvarar för utkvitterad materiel. Lämna aldrig verktyg liggande framme. Ställ in verktygslådan och övrig utrustning i ert skåp när det inte används.
- Ta reda på komponentbeteckningen *innan* ni ber handledaren om fler komponenter. Undvik onödigt spring, hämta ut samtliga nödvändiga komponenter vid samma tillfälle om möjligt.
- Återställ och städa av labplatsen efter avslutat projekt. Använd sopborsten och skyffel. Se till att virtrådsklipp inte hamnar på golvet.

### 1.5.2 Skåp

I Muxen finns ett antal låsbara skåp där projekten skall förvaras när de inte används. Vid utkvittering av nyckel till skåpet kvitteras också en mindre verktygslåda belägen i skåpet. Denna verktygslåda innehåller en standarduppsättning av verktyg, bestående av bl.a. virpistol, avbitartång, flacktång, avvirningsverktyg och kabelskalare. Dessutom ingår en sortimentask där man kan förvara sina komponenter.

***Tänk på att inte låsa in gemensam utrustning som högtalare, nummersändare, tangentbord och sladdar i skåpen.*** Gemensam utrustning skall återställas i respektive förvaringsskåp.

### 1.5.3 Tillgängliga komponenter

Databladsservern visar vilka komponenter som finns tillgängliga. Vissa komponenter saknar dock datablad i elektronisk form och kan alltså inte presenteras där. Komponenterna förvaras i komponentförrådet och kan utdelas av handledaren. En hel del vanliga ”strömkomponenter” som motstånd, kondensatorer, lödsocklar och potentiometrar samt en del monteringsmateriel finns i en hurts, åtkomliga för alla.

I komponentförrådet finns ett urval av kretsar ur TTL-familjen, men också klockoscillatorer, minnen, vissa linjära kretsar som operationsförstärkare, A/D- och D/A-omvandlare, speciella CMOS-kretsar som analoga switchar med mera.

Om ni absolut vill använda komponenter som inte finns på databladsservern, tag kontakt med handledaren. Om komponenterna är av allmänt intresse och inte är alltför dyrbara kan institutionen kanske köpa in dem. Av pedagogiska och administrativa skäl rekommenderas dock att i första hand använda de kretsar som finns tillgängliga. Dessutom kan projektet dra ut på tiden om beställda komponenter inte skulle komma i tid. Leveranstider på sex veckor är inte ovanliga!

### 1.5.4 Virning

All konstruktion i kursen sker med virning, s.k. *wire-wrap*. Virning är en snabb, säker och — rätt utförd — mycket hållbar metod för prototypkonstruktion. Alla kretsar monteras i virsocklar, som placeras i ett virkort. Andra komponenter som motstånd, kondensatorer och lysdioder löds fast på en komponentadapter, som sedan sätts i en virsockel.

Virkort med avstörda matningsskenor för spänning lämnas ut av handledaren.

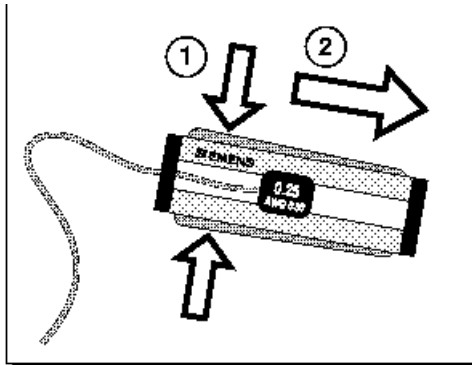
För att identifiera kretsarnas anslutningar finns märklappar i plats eller papper med bennummer att sätta på socklarna från baksidan av kortet innan man börjar vira. På databladsservern finns original för utskrift av ytterligare pappersmärklappar. Samtliga märklappar förvaras i hurtsen i Muxens samlingsrum.

Bildserien i figuren nedan förklarar hur man utför virning och avvirning med de verktyg som finns i labsatsen. Om du aldrig virat förut, studera bildserien och öva handgreppen några gånger innan du tar itu med de riktiga kopplingarna. Lägg särskilt märke till att det är viktigt att vara lätt på handen. Man får absolut inte trycka emot när virpistolen börjar röra sig uppåt, det garanterar en katastrofal virning. Var också noga med att inte låta någon avisolerad del av tråden sticka utanför virstiftet, detta orsakar gärna kortslutning. Virning enligt metoden *modified wrap* innebär att man låter isoleringen sträcka sig nästan hela första varvet runt virstiftet. Detta utesluter möjligheten till kortslutning.

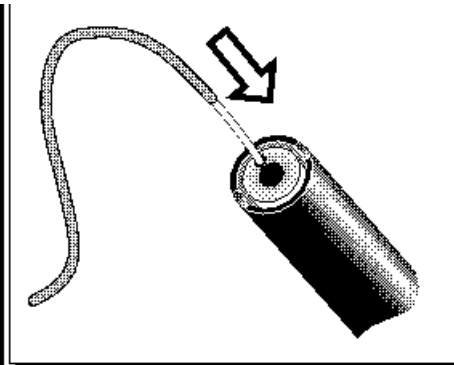
OBSERVERA! Virpistolen ser kanske robust ut, men den är ett känsligt och dyrbart precisionsinstrument. Om den utsätts för våld, tappas i golvet eller läggs ner oförsiktigt böjs eller bryts spetsen, vilket gör pistolen obrukbar. En ny spets kostar mer än femtonhundra kronor. Alltså:

**HANDSKAS VARSAMT MED VIRPISTOLEN!!**

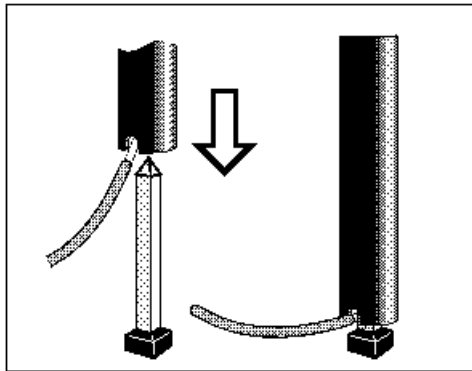
## 1 Inledning



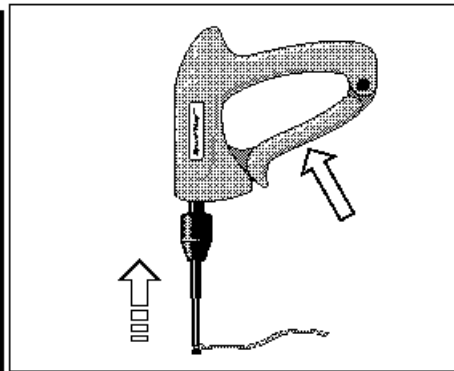
Skala av ca 3 cm av isoleringen på virtråden med skalverktyget - *inte* med avbitartången.



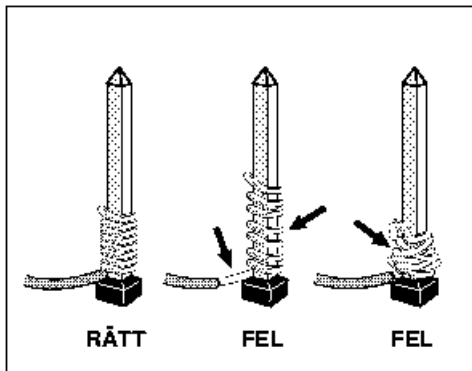
Virpistolens spets har två hål. För in hela de skalade delen av virtråden i det lilla hålet.



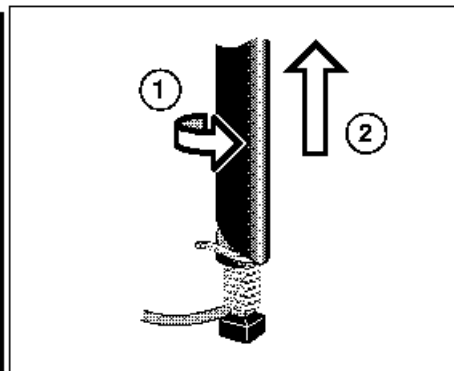
För det stora hålet i virpistolens spets över virrstiftet. Se till att ingen oskalad virtråd sticker ut.



Tryck in pistolhantaget. Var lätt på handen. Tryck inte för hårt med spetsen. Följ med upp



Drag upp spetsen från stiftet. Släpp hantaget. Om du gjort rätt har du en hållbar förbindning.



För att lossa en virning, linda upp några varv med avvinningsverktyget och drag rakt upp.

## 2 Konstruktionsmetodik

### 2.1 Allmänt

Konsten att designa digitala apparater består i stort sett av att bryta ned ett större problem i mindre delproblem och därefter realisera dessa mindre delproblem med separata, och helst oberoende, block. Denna konst är omöjlig att formalisera, och därför nästan omöjlig att lära ut (vi har försökt. . .). Det bästa sättet att lära sig denna konstruktionsmetodik är genom egna experiment och erfarenheter.

Konstruktionskurser kräver goda förkunskaper. Det är inte fel att repetera assembler-programmering nu. Kursen handlar om konstruktion på högre nivå, grunderna antas du redan behärska.

Man måste vänja sig vid att tänka i funktioner snarare än i signaler, börja med ett abstrakt blockschema och dela upp i delproblem på allt lägre nivåer tills man har realiserat hela nätet med tillgängliga standardkomponenter, mikrokontrollers och vad man har till hands. Detta förfarande går under benämningen *top-down-design*. I praktiken kan man oftast inte konstruera strikt enligt denna metod. Man måste klättra upp och ner på abstraktionsstegen i flera omgångar och omväxlande konstruera med både *top-down*- och *bottom-up*-metodik. I vissa fall kan man realisera delproblemen med standardmetoder som tillståndsgrafer och Karnaughdiagram, men i många fall kan man komma undan med mer informella metoder och mer avancerade kapslar än elementära grindar och vippor.

I och med att en mikrocontroller finns tillgänglig kan, och skall, mycket funktion läggas i denna. Tänk redan vid början av konstruktionen vilka funktioner som kan läggas i mjukvara. Detta är både en för- och en nackdel att ha funktionerna i mjukvara: Å ena sidan är det smidigt att göra korrekationer i kretsens beteende, å andra sidan luras man gärna till att inte lägga lika mycket tid på hårdvarukonstruktionen innan programmeringen — för att det är så lätt att ändra sedan. . .

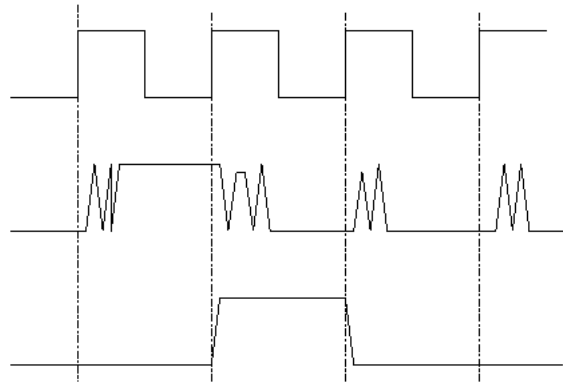
Med ett modulärt byggsätt *även* vad gäller programmet kan delsystem för delsystem testas oberoende av varann. Detta kortar ned felsökningstiden avsevärt.

### 2.2 Konstruktionstips

Vid all digital konstruktion kan de vanligaste problemen undvikas genom iakttagande av följande:

- Använd synkron klockning
- Använd avkopplingskondensatorer för spänningsmatningen





Figur 2.1: Hasardfenomen

- Undvik långa slingor i tråddragningen, speciellt för matningsspänningen
- Håll nere alla sladdlängder till rimliga längder
- Skilj på olika matningsspänningar och -jordar
- Läs — och förstå! — databladerna
- Fallgropar

Dessa problem behandlas nedan var för sig.

### 2.2.1 Synkron klockning

*Synkronism är egentligen bara ett problem vid rent digitaltekniska konstruktioner. Det kan dock fortfarande gälla i vårt fall för digitala funktioner utanför processorn. Men det finns ingen anledning att låta yttre signaler vara synkrona med processorklockan. Om det behövs kan programkod användas för att "avstudsas" tryckknappar m.m.*

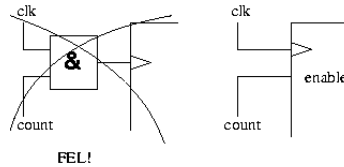
*Avsnittet tas ändå med här då det är en vanlig felorsak i digitala system.*

Synkrona digitala system förutsätter en klocksignal som är gemensam för alla delar i systemet. Brott mot denna regel är den ojämförligt vanligaste felorsaken i studentprojekt. Om klocksignalen till någon del av systemet av någon anledning inte är synkron med övriga delar blir resultatet ett asynkront sekvensnät. Asynkrona apparater kan fås att fungera, men de måste konstrueras med andra metoder än synkrona system. Bland annat är bivillkoren på signaler och tillståndsvariabler mycket strängare.

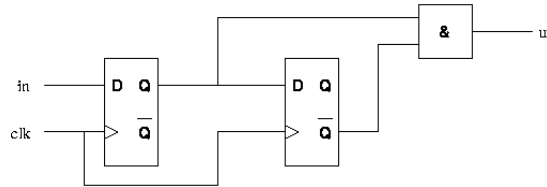
Alla klockade komponenter skall klockas från en och samma stabila klocksignal. Ingen krets får klockas med en utsignal från andra komponenter i systemet. Anledningen till detta är att nästan inga kretsar lämnar rena signaler på utgångarna. På grund av olika signalfördröjningar i nätet, både fram till kretsens ingångar och internt i kretsen, uppstår så kallade *hasardfenomen* på utgångarna. Se Figur 2.1.

Av samma anledning får under inga omständigheter systemklockan grindas utanför kretsen. En koppling som den vänstra i Figur 2.2 kommer att bete sig på ett oförutsägbart sätt. Även om signalen `count` mot förmodan skulle vara hasardfri kommer klockningen

## 2 Konstruktionsmetodik



Figur 2.2: Synkron lösning



Figur 2.3: Enpulsare

av räknaren att ske med en liten fördröjning relativt andra, direkt klockade kretsar. Fördröjningen är inte stor, endast omkring 10–15 ns i TTL-kapslar, mycket mindre i PLD, men spelar en avsevärd roll.

En synkron lösning visas i Figur 2.2. Denna koppling är säker.

Även om signalen Enable uppvisar hasardfenomen kommer den att hinna stabilisera sig tills nästa klockpuls kommer. Detta är hela idén med synkrona sekvensnät. Alla händelser i systemet sker idealt sett momentant vid, ofta uppgående, klockflank, och signalerna i nätet får sedan en klockperiod på sig att stabiliseras innan nästa klockflank anländer. Det är därför viktigt att klockflanken anländer samtidigt till alla delar av systemet. Om den kommer några nanosekunder för sent till någon komponent är det inte säkert att signalerna i systemet är stabila längre, och vad som helst kan gå snett. Observera att detta inte har med den använda klockfrekvensen att göra. Synkronicitetsproblem kan visa sig även vid låga klockfrekvenser.

Om man vill ha samma beteende hos nätet som om man kopplade räknevillkoret direkt till klockingången, det vill säga att räknaren räknar upp ett steg endast vid övergång från 0 till 1 i räknevillkoret, får man använda en *enpulsare* enligt Figur 2.2.1.

Det som hittills sagts om synkron klockning gäller speciellt vid realisering av digitala sekvensnät. Det finns i allmänhet ingen anledning att insignalerna till *mikroprocessorstyrda* system är synkrona, eller avstudsade för den delen. Mikroprocessorn är synkron internt och ofta emot sina minnen, men andra insignalerna kan tillåtas vara asynkrona.

### 2.2.2 Avkoppling

Avkoppling innebär i det här fallet att man här och där i nätet kopplar en kondensator<sup>1</sup> mellan spänningsmatning och jord. Induktansen i matningsledningen kan annars orsaka störningar när en krets plötsligt behöver mer (eller mindre) ström. Eftersom utgångarna i ett digitalt sekvensnät hela tiden växlar läge mellan hög och låg nivå kommer strömförbrukningen i varje krets att bli mycket ojämn. Speciellt vid klockflankerna uppstår stort strömbehov och därmed spikar på matningsspänningen. En kondensator, placerad *nära* kretsen, tar hand om tillfälliga strömbehov eller strömöverskott och minskar därigenom fluktuationerna i matningsspänningen. Fluktuationer som i ogynnsamma fall fortplantar sig över hela systemet och ger egendomliga, svårfunna felfunktioner.

Avkopplingskondensatorer finns redan inlödda mellan matningsspänning och jord på de virkort som används i kursen. Tänk på att inte dra för långa trådar för matningen, utan använd för varje krets närmaste virstift i kanten av kortet.

I detta sammanhang måste också nämnas jordpotentialen, 0V. På samma sätt som matningsspänningen kan denna vara ojämn av ovan nämnda orsaker. Om systemet innehåller komponenter som momentant kan dra mycket ström (motorer exempelvis) kan detta orsaka att hela jordpotentialen ”lyfts upp” flera tiondels volt, med felfunktion hos elektroniken som följd. Skall man styra stora strömmar används optokopplare.

**Separata matningsspänningar** För att minska effekten av av strömkrävande komponenter kan man i sådana fall använda separata matningar, både plus och jord, till de olika komponenterna. Det är då viktigt att det finns en (1) central referenspunkt för matning och jord och att ledningar dras direkt från denna centralpunkt ut till de olika komponenterna.

Detta gäller speciellt om analoga komponenter (operationsförstärkare m.m.) ingår i konstruktionen. För att förhindra digitala störningar i den analoga elektroniken använder man ofta separata matningar (åtminstone separata filtreringar) i detta fall.

### 2.2.3 Slingor

Långa virtrådar kan ge osäker eller felaktig funktion, framför allt vid höga klockfrekvenser. En stor trådslinga kan ha en icke försumbar induktans, och kan orsaka störningar om andra strömförande ledningar går genom eller i närheten av densamma. I långa ledningar kan även strökapacitansen till jord vara tillräckligt stor för att orsaka störningar. Långa ledningar orsakar dessutom signalfördröjningar, inte så mycket genom propageringstiden genom tråden, utan genom att de har en induktans som vid omslag mellan hög och låg nivå kan belasta en utgång kraftigt och därmed orsaka en lägre spänningsderivata än vad som normalt är fallet.

Vira med lagom korta trådar. Ett skatbo av långa trådar är inte bara störningskänsligt, det är också fullständigt hopplöst att felsöka i. Naturligtvis bör man inte heller gå så

---

<sup>1</sup>Olika tillverkningsmetoder ger olika snabba kondensatorer. För avkoppling av digitala system är det viktigt att kondensatorerna är snabba och man använder sig då av antingen keramiska kondensatorer eller sådana med tantalelektrolyt med en kapacitans av cirka  $0.1\mu F$ . Två till tre TTL-kapslar kan samsas om en sådan kondensator medan en mikrokontroller kan behöva en kondensator på varje matningsingång.

långt i sin nit att virtrådarna ligger spända som fiolsträngar mellan stiften. Detta medför lätt att en virtråd går av, vilket är ett svårfunnet fel. Det är också svårt att följa enstaka trådar om dessa är ihopbuntade till kabelstammar.

### 2.2.4 Läs databladet

Detta kan inte upprepas för ofta. All nödvändig information står i databladet. *Läs dom.* En del datablad kan vara kärva i sin framställning, låt inte det hindra genomläsning. Använd aldrig en komponent utan att först ha läst — och förstått — databladet.

### 2.2.5 Fallgropar

Projektarbete kräver mycket disciplin av deltagarna. Det finns flera fallgropar att ramla ner i. För att undvika de vanligaste, följ dessa råd:

- Välj inte ett för omfattningsrikt projekt med orealistiskt hög ambitionsnivå vid projektets början. Anpassa nivån till projektdeltagarnas antal, erfarenheter och färdigheter. Bolla idéer mot examinator.
- Låt inte förstudien ta för lång tid. Det är viktigt att komma igång med utvecklingsverktygen och få erfarenhet av de befintliga rutinerna och hårdvaran. Å andra sidan är det meningslöst att forcera förstudien och slarva igenom den.
- Kontrollera emellanåt att det ni konstruerar faktiskt är det efterfrågade.
- Ytterligare planering krävs i större grupper där deltagarna blivit specialiserade på en del av projektet. Om inte kommunikationen inom gruppen fungerar kan hela projektet plötsligt avstanna för att någon gruppmedlem inte är närvarande. Var uppmärksam på det.
- Upprätta en tidplan inom gruppen med klara delmål. Och se till att den följs. Upptäck eftersläpningar tidigt. Tänk på att det i många fall är kontraproduktivt att påbörja ett nytt projektmoment innan de tidigare momenten är testade och klara. För gärna dagbok över projektet, det underlättar tidsuppskattningar med mera.

### 2.2.6 Felsökning

Räkna med felsökning. Det går **aldrig** att vira upp hela konstruktionen och omedelbart erhålla önskad funktion.

Mycket av felsökningstiden kan elimineras om man har ett tydligt **schema** över konstruktionen. Det är också av vikt att hela konstruktionen är modulärt uppbyggd så att test av varje delfunktion kan ske så tidigt så möjligt under konstruktionstiden. En felfunktion som visar sig i en sådan (liten) modul är mycket lättare att hitta och korrigera än en felfunktion som visar sig när hela systemet är uppvirat. Underskatta inte detta. Simulera programkoden i utvecklingsverktyget.

Felsökning kan ske på många sätt, och varje konstruktion uppvisar sina speciella felsymptom. Den första åtgärden vid felaktig funktion hos projektet kan vara att kontrollera att

ledningsdragningen stämmer överens med schemat. Om så är fallet bör man tänka efter om schemat verkligen föreställer den önskade funktionen. Om felet kan spåras till själva konstruktionen är det bara att konstruera om. Programfel upptäcks förhoppningsvis redan i simulatorn. I annat fall måste man skära i koden tills man hittar det felaktiga kodavsnittet vilket kan vara mycket tidskrävande, speciellt om subrutinerna är stora.

I vissa fall kan det vara idé att lägga skvällersignaler på oanvända pinnar på kretsens utsida för att kunna mäta på dem med en logikanalysator. Många fel kan bara upptäckas med logikanalysatorn, den brukar vara ovärderlig för att studera seriell kommunikation.

För kontroll av funktioner under programmets gång kan det underlätta att koppla vissa viktiga signaler till lysdioder eller sifferindikatorer.

Slutligen ett självklart men alltför ofta försummat råd: **var noggrann** från början. Det är mycket större risk att det blir fel om man gör slarviga kopplingar från dåliga scheman, och felen blir dessutom mycket svårare att hitta.

### 2.2.7 Dokumentation och rapport

Precis som vid allt annat laborationsarbete gäller att en noggrann och systematisk löpande dokumentation är värdefull. Koppla inte utan ett noggrant schema, och dokumentera alla steg, dels för att kunna gå tillbaka och kontrollera, och dels för att beskriva designöverväganden i slutrapporten.

Se till att *alltid* ha ett uppdaterat kretsschema som stämmer överens med hur konstruktionen på virkortet ser ut. Det tar tid att hålla schemat à jour men är en tidsbesparing i det långa loppet, eftersom det underlättar felsökning. Utan kretsschema kan inte handledaren sätta sig in i konstruktionen om det skulle behövas. Det blir dessutom lättare att skriva slutrapporten med ett korrekt schema som underlag.

Använd konsekventa kända symboler för alla komponenter i kretsscheman. Hitta inte på egna schemasymboler. Det blir svårt att läsa schemat då. Som alltid bör viss omsorg läggas vid utformningen av språket i rapporten. Se vidare avsnittet om rapportens utseende på annat ställe i detta kompendium.

## 3 Examination

I kursen ges ingen tentamen. Examinationen består av fyra moment: presentation, demonstration, skriftlig rapport och återlämnande av komponenter. Slutbetyg i kursen erhålles först då samtliga moment är godkända. På kursen i sin helhet ges endast betyget "Godkänt" eller "Underkänt".

### 3.1 Presentation

Förbered presentation med videokanon. Vit tavla finns. Tidsomfattning 10–15 minuter. Utnyttja tiden rätt! Börja med att definiera kraven. Använd gärna projektorn för blockschema, flödesschema och andra översiktliga figurer, kretsscheman blir oläsliga på lite håll. Detaljerade beskrivningar på grindnivå i hårdvaran eller på instruktionsnivå i mjukvaran är i detta sammanhang oftast ointressanta. Hela gruppen ska aktivt delta i föredraget. Planera föredraget noga och glöm inte bort att vara underhållande!

### 3.2 Demonstration

I anslutning till presentationen skall den fysiska prylen också uppvisas. Detta sker i labbkalen och inte i presentationsrummet. Ungefärlig tidsåtgång för den demonstrationen är 5 minuter per grupp. Vanligen sker presentationen med sex grupper i taget.

### 3.3 Rapport

En skriftlig rapport över projektet skall lämnas till examinator och TEMA. Rapporten skall beskriva projektidén, kravskissen samt belysa de överväganden som gjorts vid konstruktionen.

Rapporten skall vända sig till personer med ungefär samma grundkunskaper som ni själva har. Den skall kunna tjäna som konstruktionsunderlag för någon annan som vill bygga en kopia av er konstruktion. Rapporten skall dessutom kunna användas för underhåll och felsökning av såväl hård- som mjukvara.

Vidare måste rapporten innehålla åtminstone:

- ett tydligt blockschema över konstruktionen,
- ett fullständigt kretsschema, samt
- en kortfattad bruksanvisning för den färdiga apparaten.

På ett försättsblad skall tydligt framgå kursens och projektets namn, datum för inlämnandet, samtliga gruppmedlemmars namn, personnummer, linje och årskurs, samt epost-adress till minst en gruppmedlem (gruppens kontaktperson). Det finns inget krav på att det bifogade kretsschemat skall vara datorritat. En underkänd rapport lämnas i retur för omarbetning.

#### Rapportförslag

Rapporten kan formuleras enligt nedanstående indelning:

- Försättsblad
- Innehållsförteckning
- Uppgiften (kort beskrivning med kravspecifikation, mål etc)
- Hårdvara (beskrivning, blockschema, kopplingschema och komponentförteckning som bilaga)
- Mjukvara (beskrivning, flödesschema, struktur, programlistning som bilaga)
- Användarhandledning
- Sammanfattning (nåddes krav och mål?, vad är bra?, vad kan göras bättre?, motivering av valda konstruktionslösningar)
- Kopplingschema

För att rapporten inte skall behöva gå en eller flera gånger mellan examinator och grupp är det viktigt att tänka igenom innehåll och upplägg.

För att få jämn detaljeringsnivå genom rapporten kan man skriva rapporten till en läsare med en viss kunskapsnivå. Tänk också på att det handlar om en beskrivning av en teknisk produkt och formen skall vara tämligen opersonligt strikt. Uttryck som ”först provade vi med XXX men då sa YYY att: ’bla bla bla’ och efter pizzan gjorde vi ZZZ. . .” hör inte hemma i rapporten. Det är inte en levnadsbeskrivning.

De flesta rapporter får påpekanden och returer som skulle undvikits om följande punkter används som rättesnöre vid skrivandet:

- Rapporten skall inledas med en framsida som talar om projektets namn, vilken typ av dokument det är och vilka medarbetarna till dokumentet är. Dokumentversion eller datum skall också framgå. Den av gruppen som skall fungera som kontaktperson måste ha sin epostadress angiven.
- Tänk på att läsaren inte vet någonting om projektet innan han får rapporten i sin hand. Rapporten måste alltså inledas med en (översiktlig) beskrivning av apparaten så att läsaren får en snabb möjlighet att komma ”i synk”. I vårt fall, med dussintals rapporter att läsa, är denna inledning särskilt viktig. Inledningen måste fungera som en lättare introduktion till hela systemets syfte utan att för den skull gå in i detaljer, kanske med undantag för något revolutionerande särdrag hos konstruktionen som förtjänar att nämnas redan här.
- För att den tekniska beskrivningen skall vara läsvärd och snabbt begriplig krävs bilder! Redan i inledningen är ett översiktligt blockschema på sin plats. I början på den tekniska beskrivningen är det i praktiken ett krav.

### 3 Examination

- Ett blockschema med lämplig detaljeringsgrad är ett utmärkt sätt att få läsaren införstådd med konstruktionens struktur. Till blockschemat måste alltid fogas en textbeskrivning som gör just det: Beskriver blockschemat. Tänk på att alltid presentera förutsättningarna innan en mer detaljerad beskrivning påbörjas.
- Även delsystembeskrivningarna måste i praktiken vara försedda med figurer.
- Bilder och figurer underlättar läsningen av dokumentet och kan lätta upp framställningen. En av de viktigaste detaljerna vid bilder och figurer är att de måste numreras och förses med läsvärd bildtext. En bildtext är inte bara ”**Fig 3.2.** Schema över styrenheten.” utan innehåller också en mindre beskrivning av det figuren förmedlar. En regel är att varje figur skall beskrivas med tre rader text.
- Samtliga bilder och figurer skall refereras i brödtexten samtidigt som de beskrivs.
- I bilagor skall man lägga sådant material som inte enkelt kan infogas i brödtexten. Det kan gälla detaljerade och kompletta kretsscheman, större figurer och programkod. Tänk på att läsaren inte skall behöva hoppa mellan brödtext och bilagor för att förstå innehållet. Bilagorna är till för den läsare som är speciellt intresserad av alla detaljer men skall inte behöva läsas för att förstå rapporten i stort.
- Även programkoden skall dokumenteras. Dels skall detta ske genom beskrivning i ord i lämpligt kapitel, dels som bifogad faktiskt programkod. Programkoden måste, på någon detaljeringsnivå, beskrivas i form av JSP-strukturdiagram. Programkoden har däremot inget i brödtexten att göra utan skall placeras i bilaga. Det finns inget som hindrar att den övergripande programbeskrivningen innehåller brottstycken av koden men koden i sin helhet skall läggas i bilaga. Programkoden skall dokumenteras så att den är så lättläst som möjligt, dvs kommentarer skall användas där det behövs och samtliga rutiner måste förses med ett inledande huvud som beskriver rutinens namn, syfte, in- och ut-parametrar osv.
- Bilagorna skall numreras och en innehållsförteckning skall upprättas för att underlätta för användaren att få översikt och snabbt hitta rätt kapitel och bilaga.
- I de fall dokumentet innehåller en användarhandledning förläggas denna till ett särskilt kapitel efter att den tekniska genomgången är gjord.

Flera frågor om hur man skriver fotnoter, hänvisningar, samman- och särskrivning osv besvaras i *Svenska skrivregler* utgiven av Svenska språknämnden.

I och med att dessa punkter uppmärksammas ovan kommer de rapporter som inte följer dem att garanterat returneras.

#### Kopplingschema

- Alla pinnar skall numreras/namnges
- Samtliga signaler skall markeras med namn
- Komponenter skall numreras så att de lätt återfinns i komponentförteckningen

#### Programlistning

- Programmet skall vara rikligt och förnuftigt kommenterat



### *3 Examination*

- Programlistningen skall innehålla namn, datum och revisionsnummer
- Använd programhuvud på de olika rutinerna
- Använd subrutinnamn som beskriver vad de representerar

## 4 Projektkrav

Här följer de projekt du kan välja att utföra under perioden. De är ordnade i ungefärlig svårighetsgrad men beroende på hur mycket extra du och din grupp vill lägga till kan de bli godtyckligt komplicerade.

Nu är inte syftet med projekten att de ska vara komplicerade. Ett projekt bör vara en trevlig balans mellan hårdvara och mjukvara. Vad som är ”trevligt” i detta sammanhang avgörs vid en granskning av din grupps kompletta projektförslag och examinator.

Många komponenter återfinns på Datortekniks databladsserver <http://vanheden.isy.liu.se>

Efter att du valt grupptillhörighet (4 pers per grupp) ska gruppen:

- Välja ett projekt ur dessa projektförslag som verkar kul.
- Fundera på hur det kan anpassas för att passa gruppen.
- Skriva en kort beskrivning av det resulterande projektet och en skiss över hur gruppen tänkt lösa det hela. Detaljerade schemata osv behövs inte men det ska klart framgå att allt är genomtänkt och att det finns en lösningsväg. Ett hyggligt detaljerat blockschema **måste** vara med.
- Häfta ihop beskrivningen med anmälningsslappen som sitter sist i Projektförslagen. Diskutera med examinator.

### Projekttyper

Förutom kravspecifikationsexemplen nedan kan projekt innehålla och/eller vara varianter av till exempel

- Kommunicera via laser
- Inspelningsmaskin/fickminne
- Ekoenhet
- Spel på grafisk LCD 64x128 pixlar
- Spel på RGB-display, DAMatrix, 8x8 pixlar
- RGB-strip

### Kravspecifikation TSIU51 — Projekt 1. Nummerpresentatör

*Samtliga grundkrav och minst ett utökat krav skall väljas.*

## 4 Projektkrav

### *Uppgift:*

I detta projekt bygger du en apparat som läser av de DTMF-toner (*Dual Tone Multi Frequency*) som skickas innan telefonens ringsignal kommer. Tonerna som kan skickas via en lokal telefonväxel eller separat nummersändare kan även innehålla information om bl.a. vem som är uppringande eller om numret är hemligt.

### *Grundkrav:*

- Det mottagna telefonnumret skall visas på en display.
- Apparaten skall kunna ha 10 nummer i minnet.
- Det skall gå att bläddra mellan numren med upp-/ner-knappar.
- Inkommet nummer skall kunna raderas.
- Varje nummer skall tidsmärkas.

### *Utökade krav:*

- Blinkande ljussignal skall meddela att nytt nummer inkommit och slockna när någon knapp trycks.
- Ljudsignal skall meddela att nytt nummer inkommit.

## **Kravspecifikation TSIU51 — Projekt 2. MIDI-styrning**

*Samtliga grundkrav och minst ett utökat krav skall väljas.*

### *Uppgift:*

I detta projekt bygger du en MIDI-baserad styrning som via en synt spelar upp MIDI-information. Det finns en föreläsning om MIDI-standarden som behövs för att gruppen ska komma igång.

### *Grundkrav:*

- Styrningen skall kunna överföra MIDI-meddelanden till en synt.
- Ett tangentbord eller annan yttre inmatningsenhet skall användas som ”klaviatur”.
- Det skall finnas möjlighet att ändra volym på den utsända informationen.
- Mikrokontrollerns interna seriekrets skall användas.

### *Utökade krav:*

- Med yttre joystick skall ljudets tonhöjd kunna ändras.

## **Kravspecifikation TSIU51 — Projekt 4. DCF77-styrenhet**

*Samtliga grundkrav och minst ett utökat krav skall väljas.*

## 4 Projektkrav

### *Uppgift:*

I detta projekt bygger du en apparat som styr en digital radiostyrd klocka enligt DCF77-standard.

### *Grundkrav:*

- Apparaten skall på ett säkert sätt kunna meddela tid till en radiokontrollerad klocka.
- Apparaten skall kunna sätta tid och datum helt godtyckligt.
- Apparatus räckvidd skall vara mindre än 20 cm.
- Klockans noggrannhet skall vara bättre än 60 sekunder per dygn.

### *Utökade krav:*

- Enheten skall kunna gå snabbare och långsammare än ”normaltid”.
- Enheten skall kunna agera ”föreläsningklocka” dvs skynda på minuterna 15–60 och sakta ner minuterna 00–15.

## **Kravspecifikation TSIU51 — Projekt 5. Morsemottagare**

*Samtliga grundkrav och minst ett utökat krav skall väljas.*

### *Uppgift:*

I detta projekt bygger du en telegrafmottagare som visar mottagen text på en display. Se laboration 2 i kursen TSIU25 för specifikation av morsetelegrafi.

### *Grundkrav:*

- De mottagna tecknen skall visas på en display.
- Apparaten skall kunna hela alfabetet, siffror och skiljetecken enligt internationella morse-standard.
- Den mottagna texten ska rullas fram automatiskt.
- Apparaten ska kunna ta emot morsetelegrafi i hastigheter mellan 30 och 150 tecken per minut.
- Apparaten ska kunna ta emot manuellt sänd telegrafi såväl som maskingenererad.

### *Utökade krav:*

- Apparaten skall automatiskt anpassa sig om den sända texten varierar i hastighet (inom rimliga gränser).
- Apparaten skall kunna ta emot ljussignalerad telegrafi.

## Kravspecifikation TSIU51 — Projekt 6. Plotter

*Samtliga grundkrav och minst ett utökat krav skall väljas.*

### *Uppgift:*

Konstruera hårdvara som styr en plotter som ritar på ett papper. Plottern är styrbar med stegmotorer i X- och Y-led. I X-led genom att styra pennan och i Y-led genom att rotera den trumma papperet ligger på. Genom en solenoid kan pennan tvingas ligga an mot papperet. Indata kan vara förinställda texter men även figurer och bilder. De senare måste antagligen gå igenom någon kant-extraherande bildbehandling först.

### *Grundkrav:*

- Linjer och cirklar ska kunna ritas.
- Apparaten skall kunna erhålla ritinformation (punkter, linjer, radier) från yttre hextangentbord.

### *Utökade krav:*

- Plottern ska kunna styras från bildinformation i en PC.
- Plottern ska kunna styras från en yttre joystick.

## Kravspecifikation TSIU51 — Projekt 7. Skywriter

*Samtliga grundkrav och minst ett utökat krav skall väljas.*

### *Uppgift:*

Konstruera en Skywriter, dvs lysdiodsrad som när den a) fäst längst ut på en pinne och viftas fram och tillbaka i luften, eller b) fäst på roterande cykelhjul, kakburk, e.dyl, visar en vald text. En sensor måste användas för att känna av a) pinnens ändlägen eller b) hjulets rotation. Texten skall kunna ändras på något sätt under drift.

### *Grundkrav:*

- Texten skall inte vara konstant oföränderlig.
- Texten skall visas stadigt och utan onödiga hopp.
- Apparaten skall fungera autonomt, utan yttre spänningsanslutning.
- Apparaten skall automatiskt anpassa sig efter varierande rotationshastighet eller ”viftningstakt”.
- En teckenhöjd om minst 8 pixlar skall användas.
- Om hjul används skall rotationshastigheten i m/s anges i texten, annars slagtakt i Hz.

### *Utökade krav:*

- Texten skall kunna ändras under gång från yttre tangentbord.
- Texten skall kunna styras från en IR-fjärrkontroll. (Exvis en text per siffra på kontrollen.)