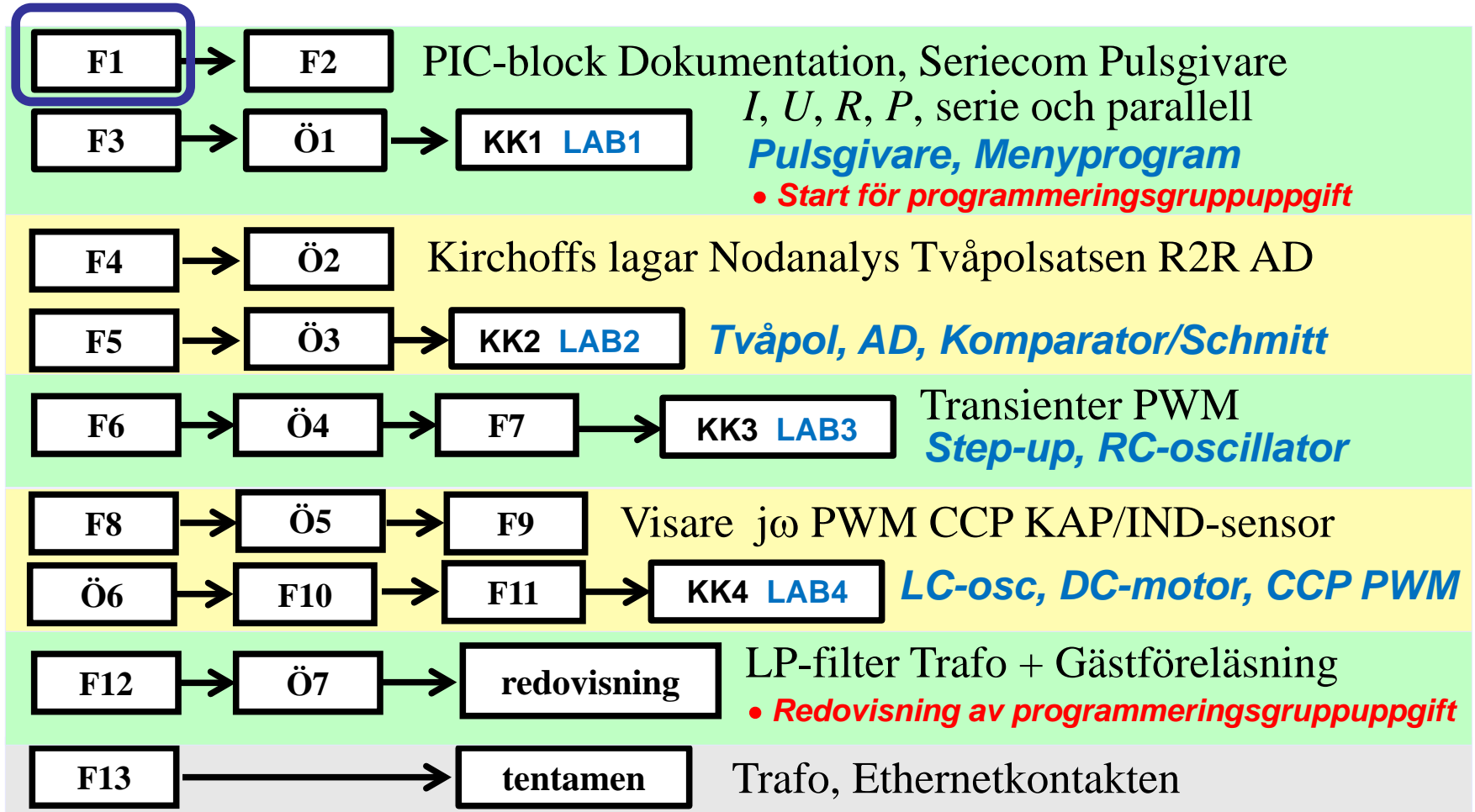


IE1206 Inbyggd Elektronik



Behöver en dataingenjör veta något om den **verklighet** datat

återspeglar?



Inte bara generella databaser ...

Mer än **98%** av världens processorer sitter idag i inbyggda system.

Inbyggda system används **överallt**, i satelliter, robotar, bilar, flygplan, och mobiltelefoner, ner till minsta radioapparat, hiss, tvätt- eller diskmaskin.

Inbyggda system – dataprogram i produkter

För att kunna skriva ett bra program måste man *helt enkelt förstå* produkten.



*Husqvarna robotdammsugare
– ett exempel på ett inbyggt system*

Ofta behövs det då ett **utvecklingsteam** av medarbetare med *olika* kompetenser.

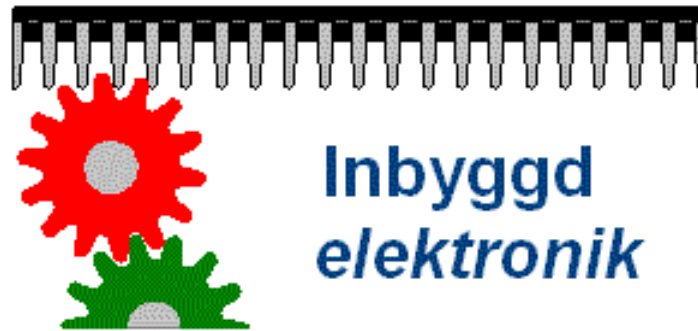
Du som är programmerare ...

I utvecklingsteamet samverkar ofta data- maskin- och elektro- tekniker.

Programmeraren av det inbyggda systemet är oftast den som ger produkten det mesta av sin funktion.

William Sandqvist william@kth.se

Kursen Inbyggd elektronik



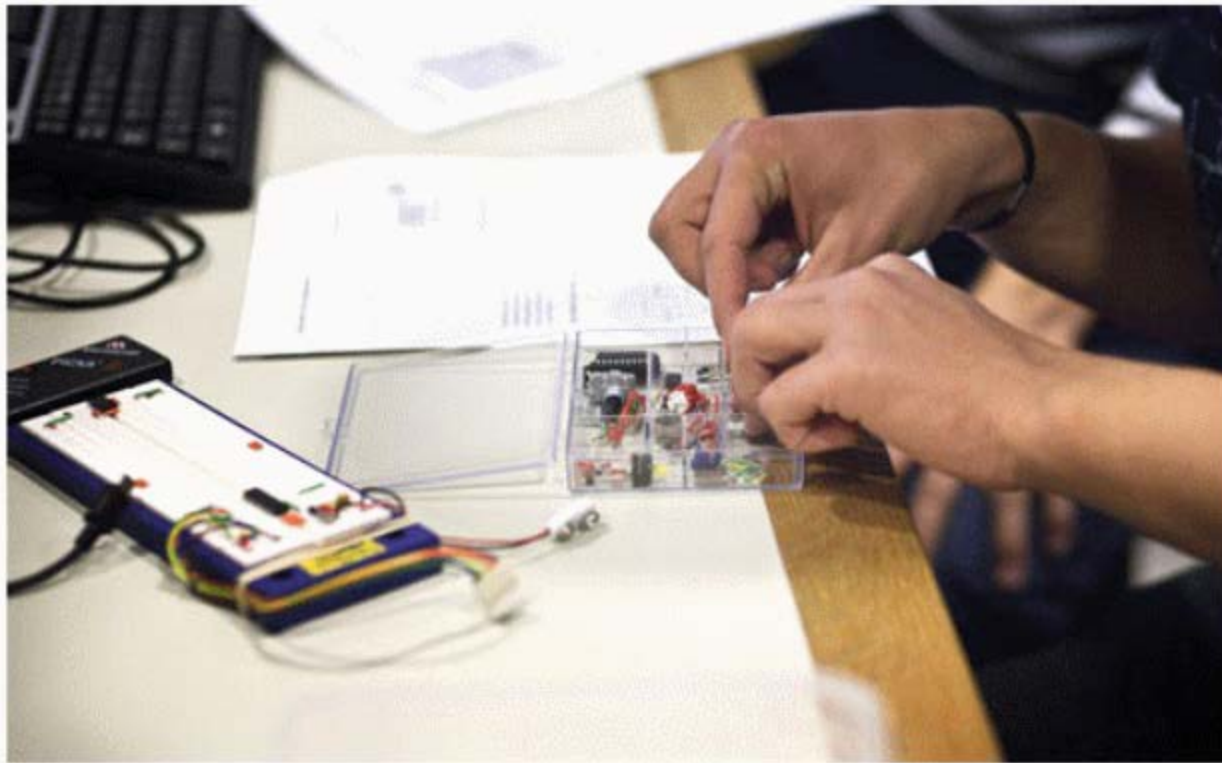
*Kursen **Inbyggd Elektronik** utgör en praktisk grund inför kommande projektkurser och ditt examensarbete.*

*Kursen ska göra dig förtrogen med elektroteknikers terminologi och sätt att resonera, så att du på ett **aktivt sätt** kan samverka, som en **värdefull medlem** av ett utvecklingsteam för produkter med inbyggnadsprocessorer.*

Laborationer – hands on

Laborationerna lär ut både programmering av en **PIC-processor** och **ellära-elektronik-kretsar!**

Vi lär oss **elektronik** med skolans minsta processor!

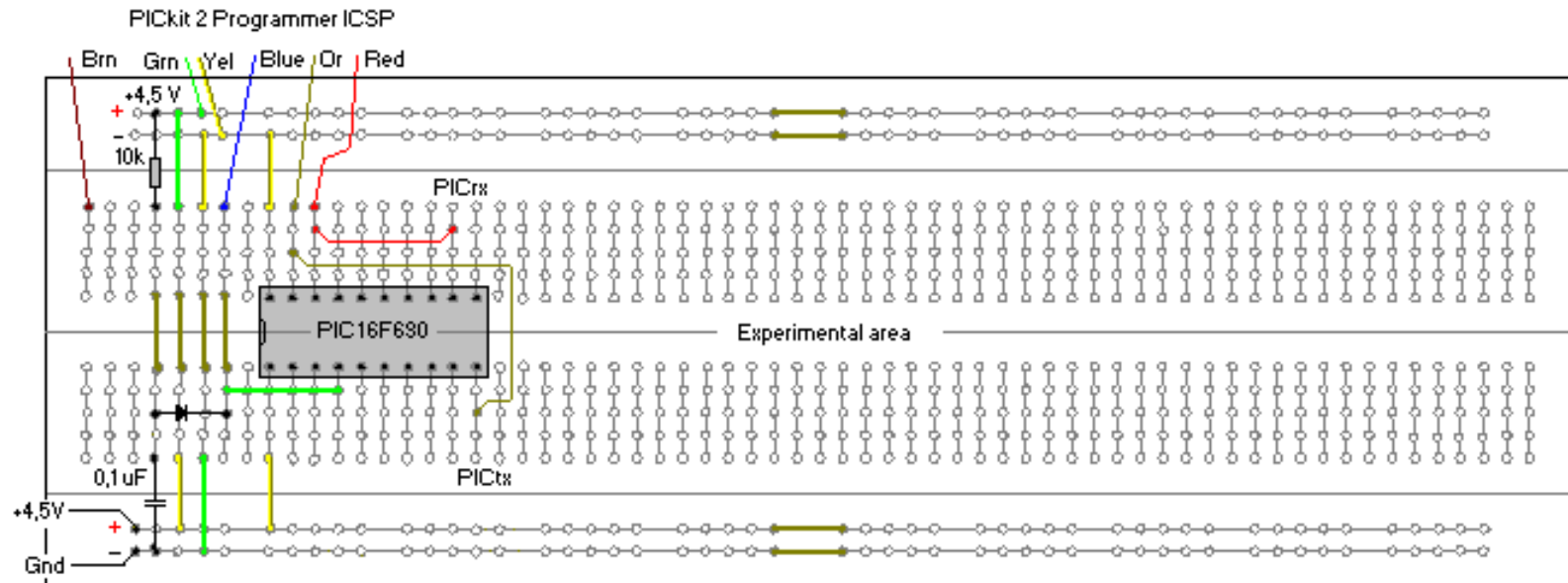


William Sandqvist william@kth.se

Busenkelt utvecklingsystem!



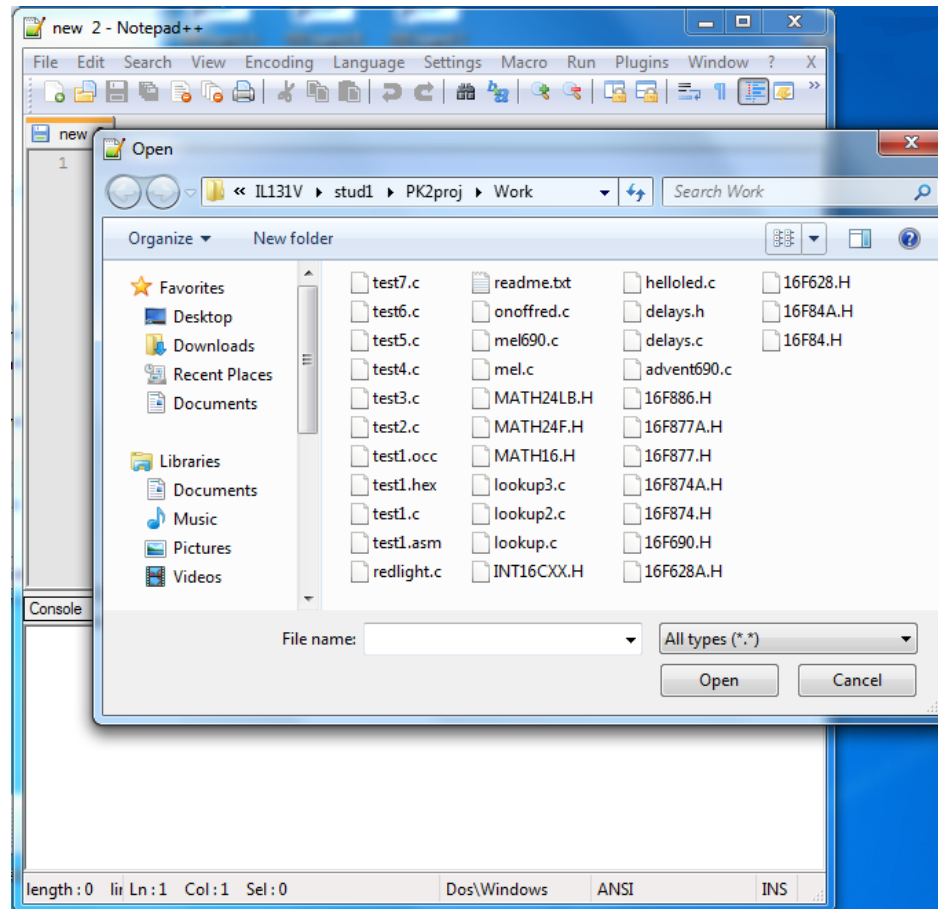
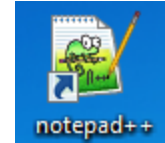
PIC 16F690 8-bitars
microcontroller



4.5V Battery

Orange and Red jumpers are for PICkit2 UART Tool

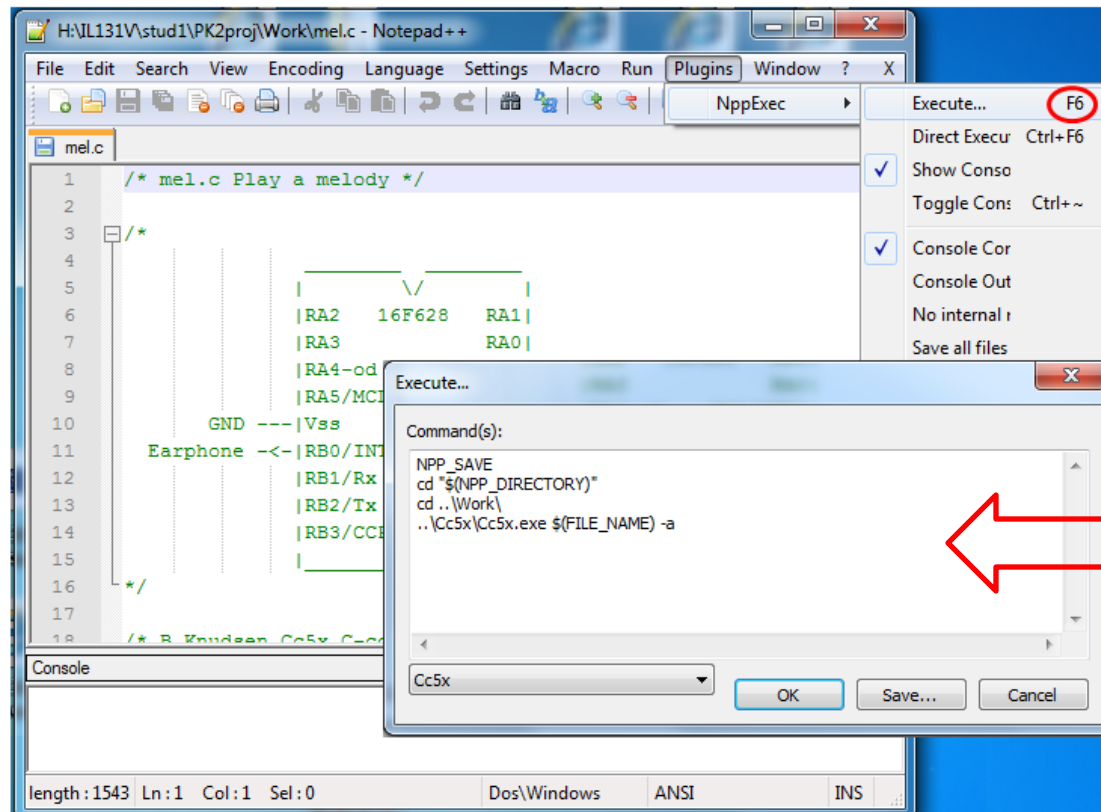
Texteditor Notepad++



Kompilator Cc5x



- Kompilatorn startas inifrån texteditorn **Notepad++**.



Ett script som startar kompilatorn

Kretsprogrammeraren PICKit 2



PICKit2

Fristående program
för Kretsprogrammeraren

För nedladdning av kod
till chip



*Blixtnabb
nedladdning av
koden!*

The screenshot shows the PICKit 2 Programmer software interface. The title bar reads "PICKit 2 Programmer". The menu bar includes File, Device Family, Programmer, Tools, View, and Help. The main window displays configuration details for a PIC16F628A device, including User IDs (FF FF FF FF), Checksum (19FF), and Configuration (21FF). A yellow status bar indicates "PICKit 2 found and connected. PIC Device Found." Below this, there are buttons for Read, Write, Verify, Erase, and Blank Check. The VDD PICKit 2 section has checkboxes for On and /MCLR, with a voltage set to 2.5V. The Program Memory section is enabled and set to Hex Only, showing a table of memory addresses (000, 008, 058) and their corresponding hex values (3FFF). The EEPROM Data section is also enabled and set to Hex Only, showing a table of addresses (00, 10, 20, 30) and their corresponding hex values (FF FF FF FF). On the right side, there are buttons for "Auto Import Hex + Write Device" and "Read Device + Export Hex File". The PICKit 2 logo is visible in the bottom right corner of the interface.

William Sandqvist william@kth.se

Programmeringsuppgift i grupp om två ...

Programmeringsuppgiften ska ge dig en praktisk erfarenhet av enklare elektronik kopplingsarbete. Detta är användbar kunskap inför **projektkurser** och **examensarbete**.

Programmeringsuppgiften redovisas med en **demonstration/presentation** och en skriftlig ”**Minirapport**”.

Teknisk rapport?

Rapport om ett program för ett inbyggt system

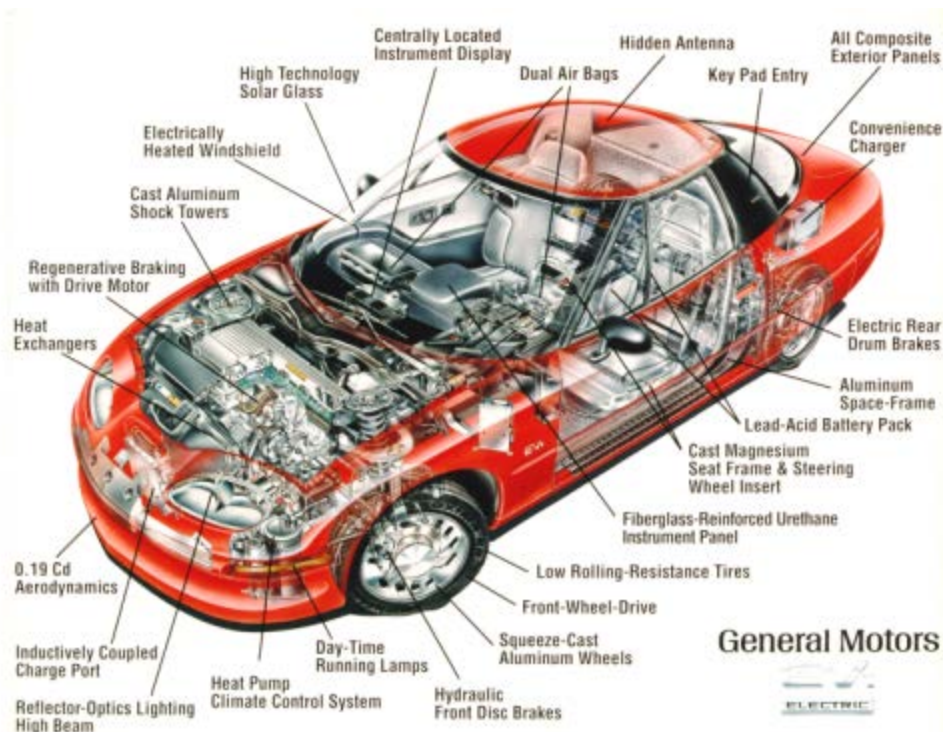
Man dokumenterar:

- Systemets uppbyggnad
- Programmets omfattning
- Programmets uppbyggnad
- Hur programmet/funktionen testats

Tekniska rapporter innehåller ”figurer” och text i samverkan.
Figurerna är ofta olika typer av **diagram**.

● Sammanställningsbild

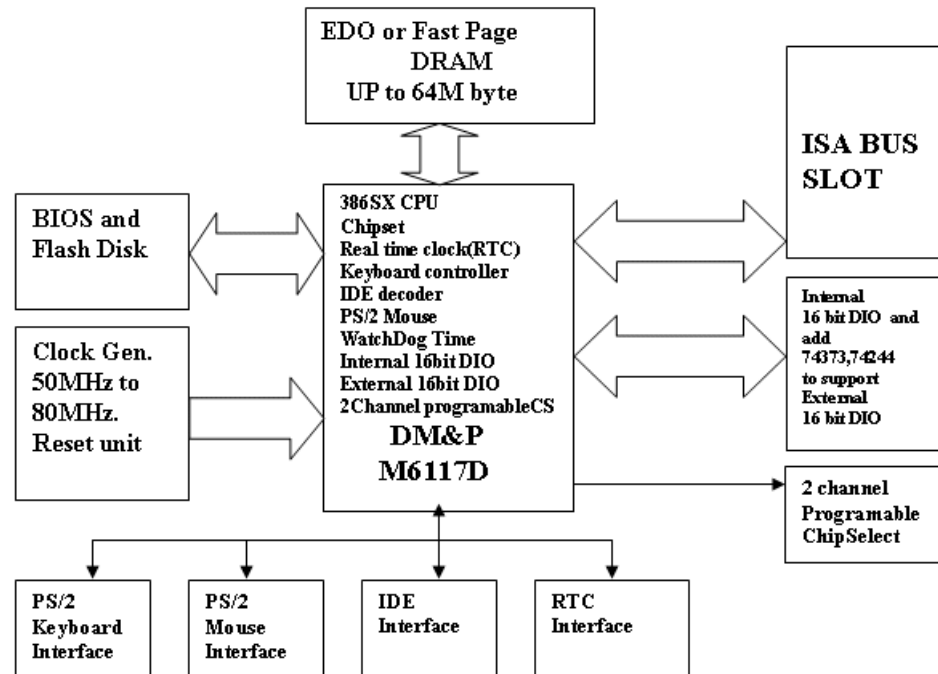
En inbyggd processor används inbyggd i en produkt. Det går nog knappast att beskriva processorns dator-program utan att man först åtminstone översiktligt förklarar **produkten!**



En **sammanställningsbild** definierar vad de olika delarna kallas för – man använder sedan exakt de namnen konsekvent genom hela rapporten när man behöver hänvisa till produkten.

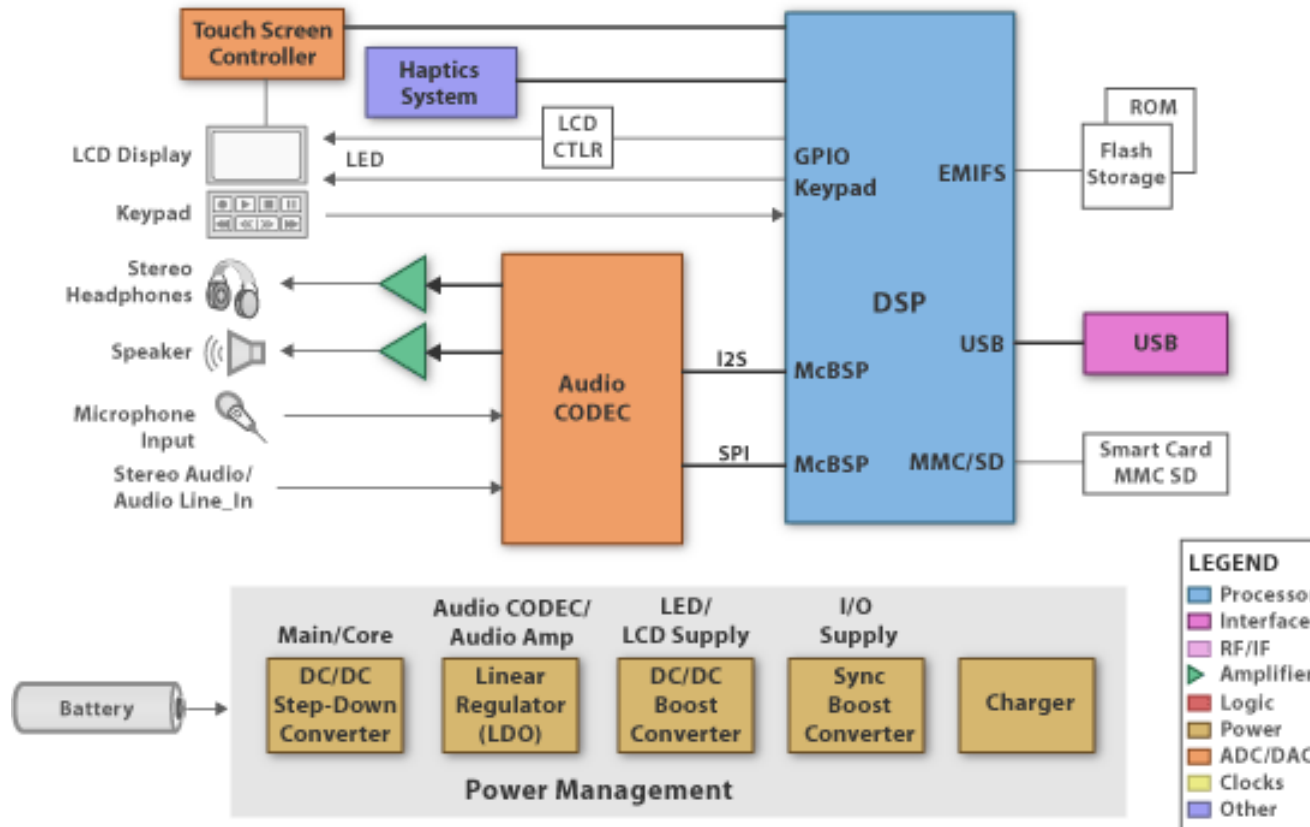
• Blockdiagram

Ett blockdiagram är ett diagram över ett system där de principiella delarna, eller funktionerna, representeras av block sammanbundna med pilar som visar blockens relationer.



Det krävs inga avancerade hjälpmedel för att rita ett **blockdiagram**. Blockdiagrammet gäller produktens delar. Andra diagramtyper används för att beskriva programmets struktur.

Illustrerat blockdiagram



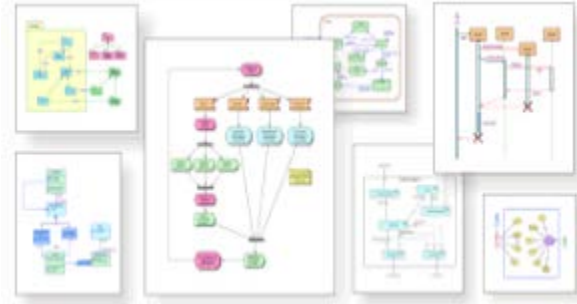
Vänder man sig till läsare utanför gruppen av tekniker händer det ofta att blocken är färglagda och illustrerade ...

Diagramtyper för programkod?

Hur beskriver man sin
programkod i bild?

(Det finns en standard, **UML**)

Unified Modeling Language (UML) är ett objektorienterat generellt språk för modellering av alla typer av system.

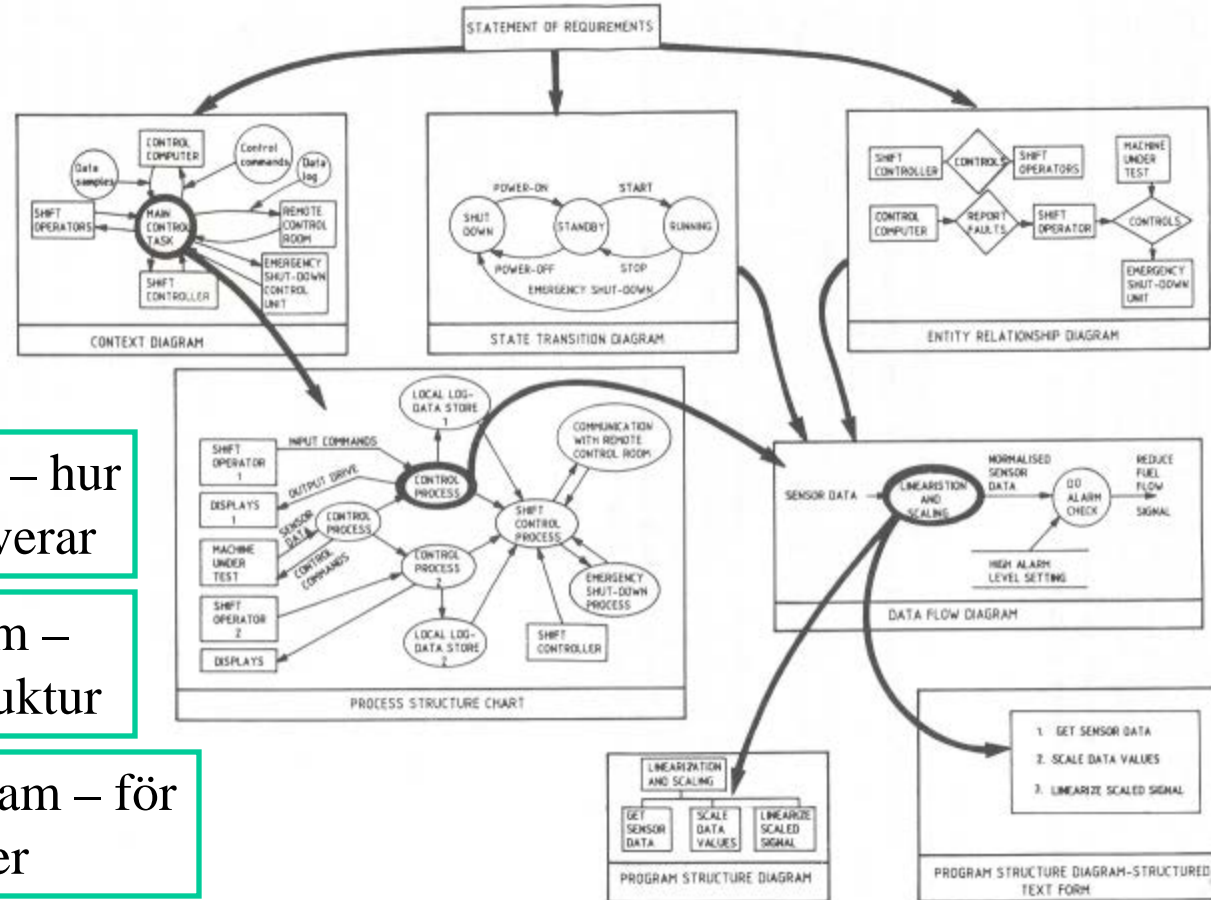


UML – innehåller en mängd olika diagramtyper.

Språket används främst inom programvarukonstruktion men är även användbart inom andra områden som exempelvis modellering av affärsprocesser med mera. Genom att skapa en modell av systemet som skall konstrueras blir det enklare att förstå och bygga det.

UML är dock för omfattande för denna kurs ...

Diagramtyper för programkod

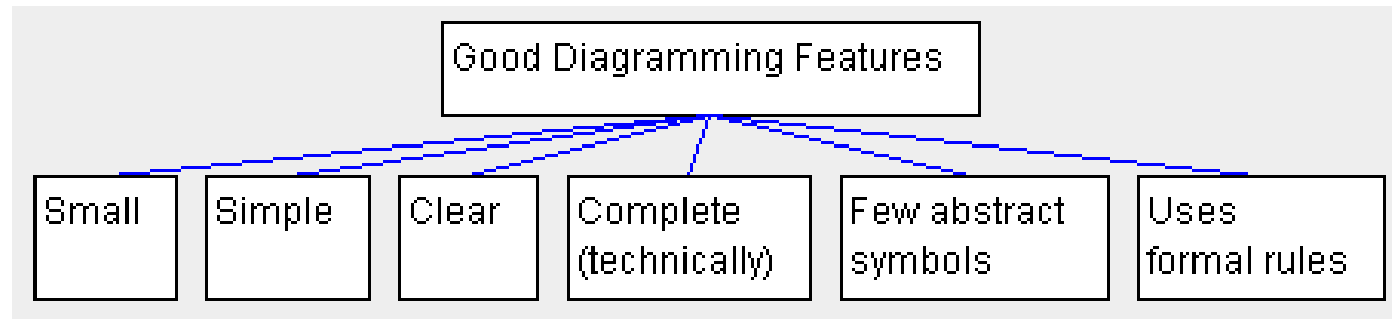


Tre vanliga diagramtyper:

- Flödesdiagram – hur processorn exekverar
- Strukturdiagram – programmets struktur
- Tillståndsdigram – för tillståndsmaskiner

Pilarna visar hur **flera olika** diagramtyper samverkar och kan behövas för att ge en helhetsbild.

Bra diagramegenskaper

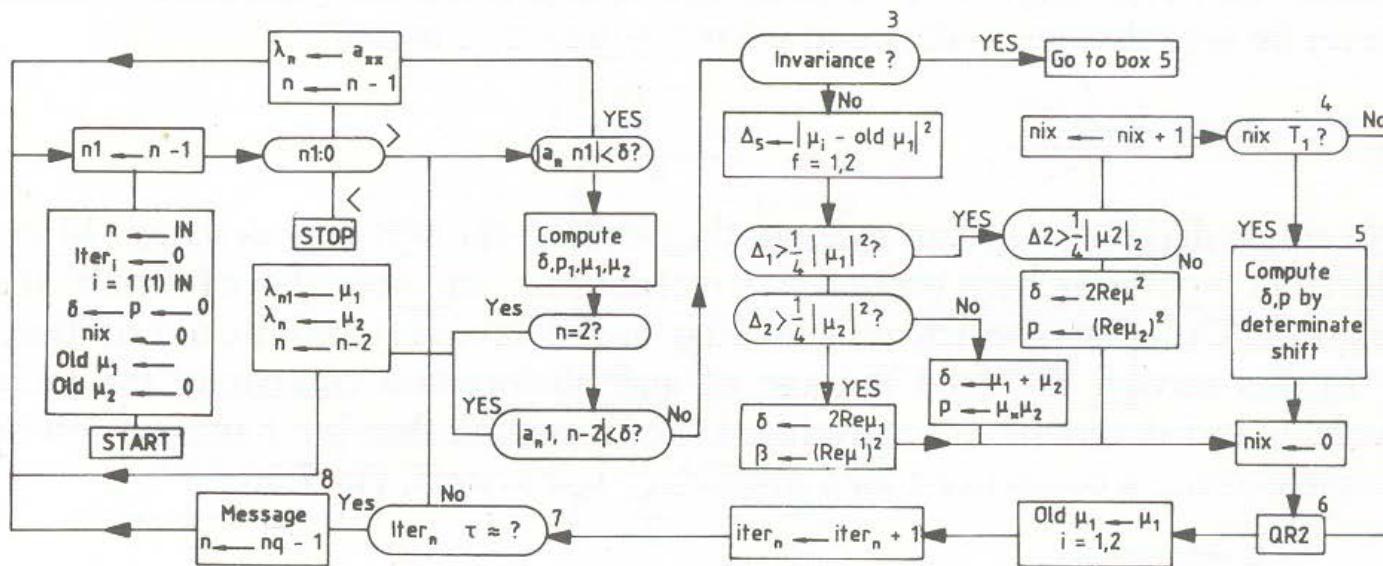


Ett bra diagram ska vara litet, enkelt, klart, innehålla få abstrakta symboler, och följa formella regler.

Ett bra diagram är till god hjälp när man ska förklara systemet för andra.

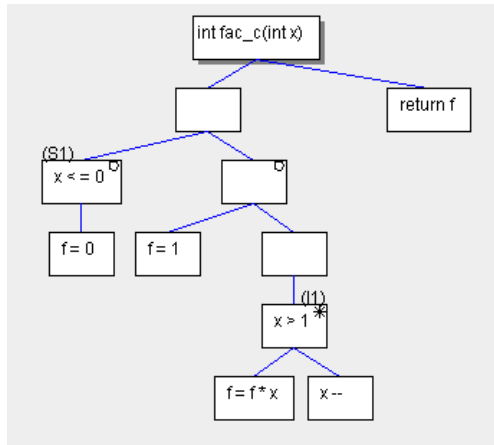
Exempel på ett diagram som bryter mot alla dessa regler!

Bra diagram-egenskaper



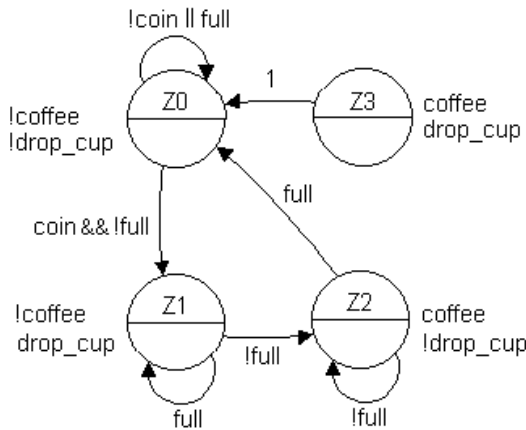
Så, diagrammen blir bara till
hjälp om Du anstränger dig ...

Vi använder ...



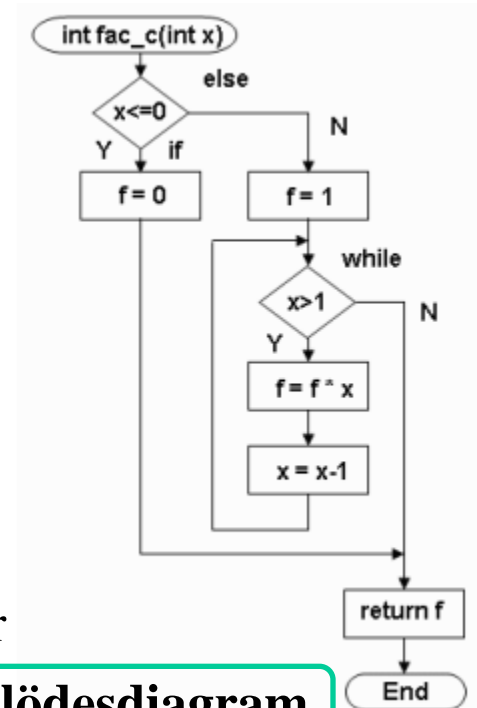
Strukturdiagram

– programmets struktur



Tillståndsdigram

– för tillståndsmaskiner



Flödesdiagram

– hur processorn exekverar

• *Se övning 1!*

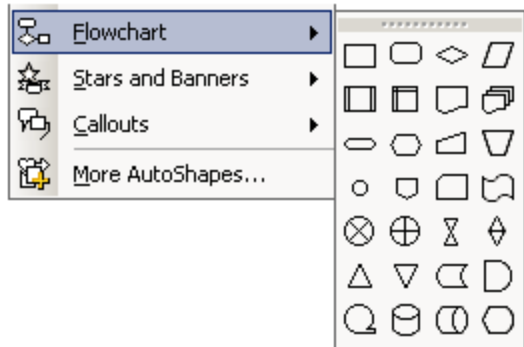
• *Se Digital Design!*

• *Se Datorteknik!*

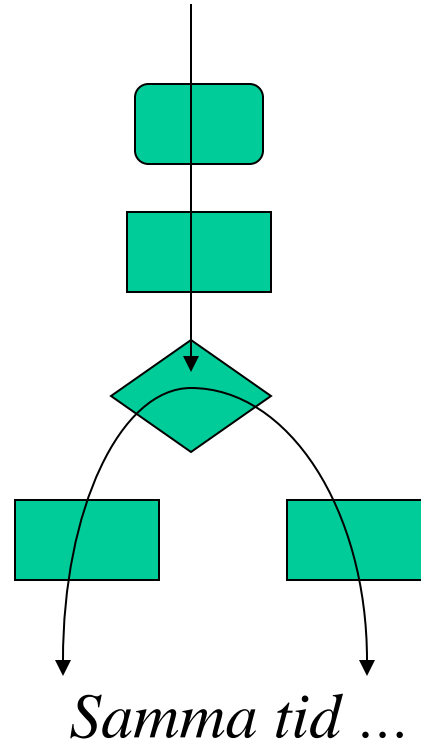
● Flödesdiagram

Flödesdiagram använder man när det är viktigt att **visa precis hur processorn exekverar koden**. Till exempel när man analyserar assemblerkoden för att kunna se till att olika vägar genom programmet tar samma tid.

Programflöde

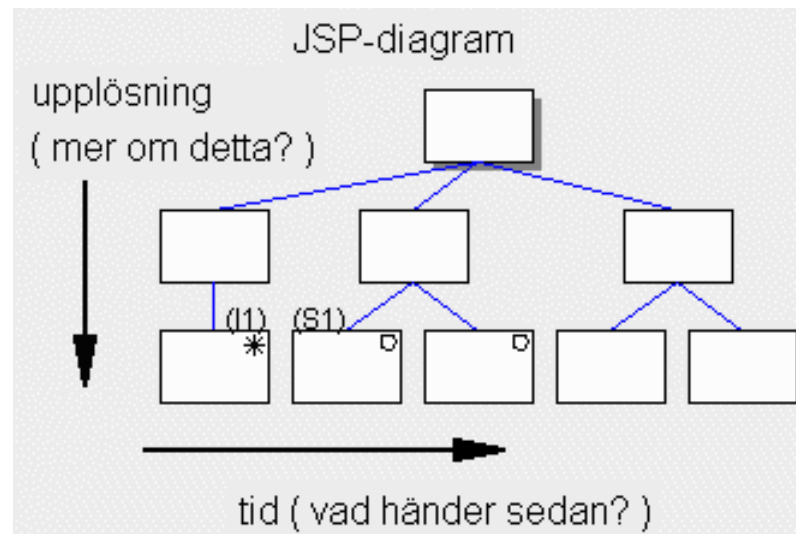
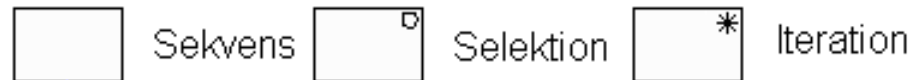


Verktyg för att rita flödes-schemor finns tex. i **Word** och **Powerpoint**.



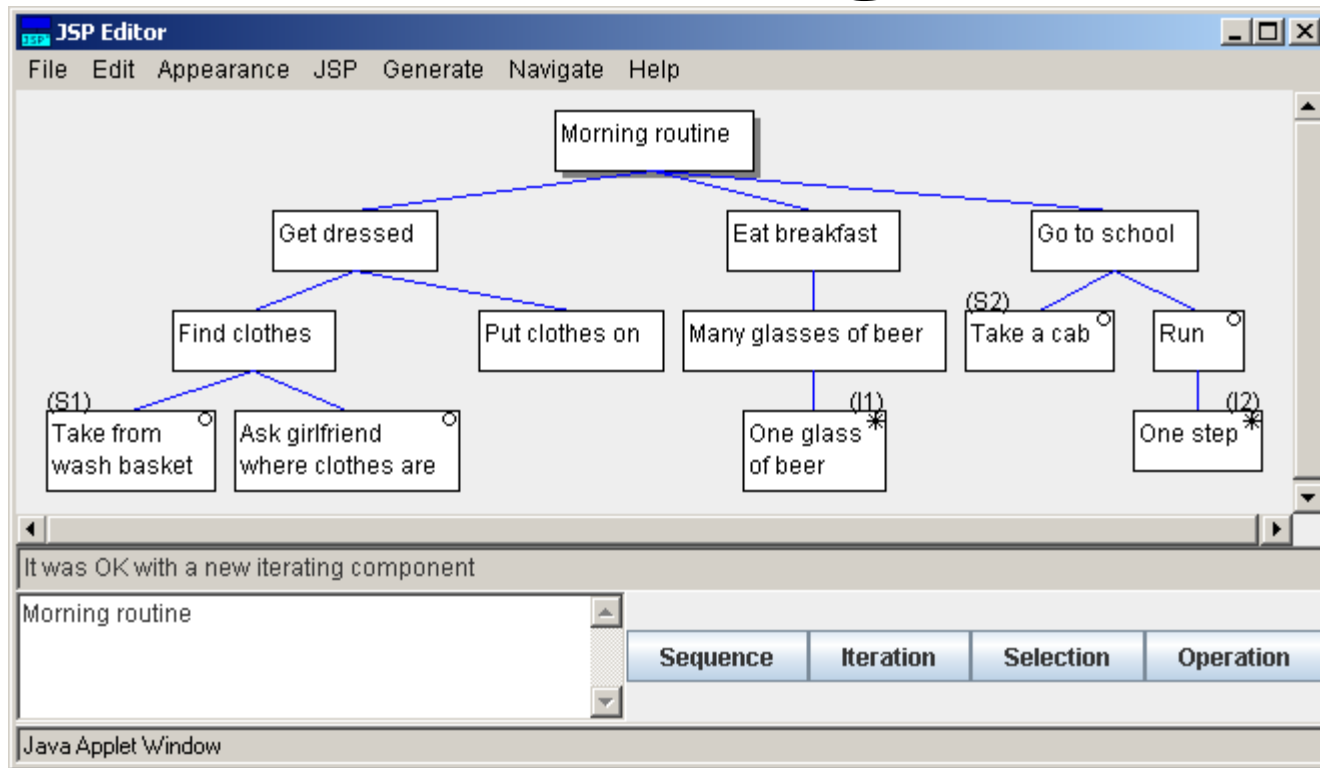
! →
*Men programstrukturer beskriver man bättre med ett **strukturdiagram**.*

• Strukturdiagram



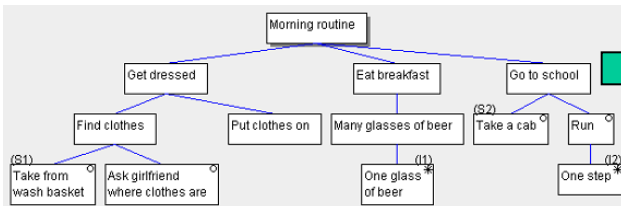
Strukturdiagrammet är ett lätttritat (bara tre symboler) **tvådimensionellt** diagram (dimensioner: tid, upplösning).

Strukturdiagram



Ta gärna hjälp av JSP-editorn när Du ritar strukturdiagram!

• C-skal från diagrammet



Från strukturdiagrammet kan man direkt få ett C-kodskal – ett bra nybörjarhjälpmedel!

```
C Program
Fil
int main()
{
    if (S1)
        /* Take from - wash basket */
    else
        /* Ask girlfriend - where clothes are */
    /* Put clothes on */
    while (I1)
        /* One glass of beer */
    if (S2)
        /* Take a cab */
    else
        while (I2)
            /* One step */
}
```

Java Applet Window

Kunskapskontroll inför Lab1

JSP-Strukturdiagrammets delar Sekvens Selektion Iteration

Bocka för vilka olika "roller" de tio rutorna i strukturdiagrammet har:

| Funktioner | Komponenter | | | | | | | | | |
|---------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| | A | B | C | D | E | F | G | H | I | J |
| Sekvens | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Sekvensdel | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Iteration | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Iterationsdel | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Selektion | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Selektionsdel | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Tryck på knappen för att beräkna din svars kod. *Skriv ner dina svars koder inför slutredovisningen*

| | |
|--|--|
| b) JSP-Strukturdiagram, svars kod | |
| <input type="button" value="Calculate"/> | Svars kod <input type="text"/> |

Kunskapskontroll inför Lab1

JSP-Strukturdiagrammets delar

Bocka för vilka olika "roller" de tio rutorna i strukturdiagrammet har:

| Funktioner | Komponenter | | | | | | | | | |
|---------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | A | B | C | D | E | F | G | H | I | J |
| Sekvens | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Sekvensdel | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Iteration | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Iterationsdel | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Selektion | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Selektionsdel | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

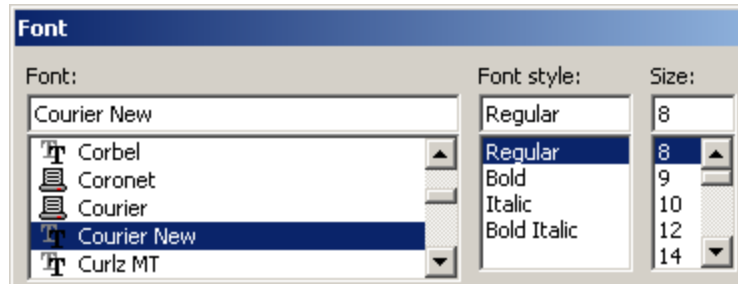
Tryck på knappen för att beräkna din svars kod. *Skriv ner dina svars koder inför slutredovisningen*

| | |
|-----------------------------------|----------------------|
| b) JSP-Strukturdiagram, svars kod | |
| Calculate | Svars kod |
| | <input type="text"/> |

Fortsätt själva ...

• Att publicera programkod

Programkod skrivs med ett typsnitt som har **fast bredd** ("skrivmaskinstil"). Vanliga typsnitt har annars variabel bredd: "i" är ju smalare än "W".



Courier New, är ett sådant "fast" typsnitt. Har Du använt 80 tecken per rad i texteditorn ska Size väljas till 8 punkter för att raderna ska få plats på A4. Om raderna bryts blir koden svårläst. Observera! Tab-tecken kan behöva bytas ut mot ett antal mellanslag.

Kommentarer

Om man skriver alla kommentarer på **engelska** så kan man dra nytta av att det finns miljoner duktiga PIC-programmerare världen över (se Internet).

Du kan räkna med att lätt hitta någon som *förstår* ditt program och som är beredd att vidareutveckla din kod (mot skälig betalning ...).

Om Du Publicerar kod med engelska kommentarer så är den tillgänglig för hela världen.

Kommentarer // eller / */
på engelska.*

● ASCII-grafik

/*

```
      |-----|
      | RA2   16F628   RA1 | ->- PWMGND 1/0
      | RA3                               RA0 |
      | RA4           RA7/OSC1 |
      | RA5/MCLR     RA6/OSC2 |
GND ---| Vss           Vdd | -- +5V
      | RB0/INT   (RB7)/PGD |
      | RB1/Rx    (RB6)/PGC |
2688 Hz | RB2/Tx           RB5 |
      | RB3/CCP   (RB4)/PGM |
      |-----|
```

Use LP-filter to se sinus

*/



Viktig information kan bakas in i programkoden som ASCII-grafik. Den finns då tillgänglig för alla oavsett system.

Alla kursens programexempel har ASCII-grafik som visar hur man ska "koppla" chippet.

Exempel på ASCII-grafik

```

/*
  LM2577 Step Up converter (Top view)
  ELFA 73-277-45
  
```

```


  |-----|
  | LM2577 |
  | Vin    5 |<-<- Vin +7.7 V
  | Switch 4 |>-> D_Schottky >-> Vout +12V
  | GND    3 |--- GND
  | FeedBack 2 |<-<- R divider (R1 18k, R2 3.3//5.6k)
  | Compare 1 |--- RC-net (Rc 2.2k, Cc 0.33uF) - GND
  |-----|
  | Dot at pin 1
  |-----|
  GND
  L 150mH (4-5) ELFA 58-690-37
  Cin 0.1 uF (5-GND)  Cout 680 uF ( Vout-GND)
  */
  
```

```

/*
  Board contact
  1 2 3 4 5
  ( . . . . . )
  \ . . . . /
  -----
  6 7 8 9
  1 +7.7V
  2 +7.7V
  3 GoSignal
  4 +5V Vcc
  5 +5V Vcc
  6 Release
  7 Gnd
  8 Gnd
  9 Gnd
  */
  
```

```

/*
  ADXRS610 one axis Gyro SparkFun breakout board (Top view)
  (15 mm * 15 mm)
  
```



```

  +5V ---| +5V
  Gnd ---| GND
  Rate <-| Rate
  <-| Temp
  TestP >-| ST2
  TestN >-| ST1
  |-----|
  Rate LP-filter 2k2, 0.68 uF
  */
  
```

With ASCII-grafics inside your code, you will have hardware information in reach all the time while you are coding.

William Sandqvist william@kth.se

Prototypbygge

Funktionsprototyp - Produkt

- Functional prototype (working prototype)

Meningen med en **funktionsprototyp** är att övertyga andra om att man har löst de tekniska/programmeringstekniska problem som kan finnas med designen.

Man har *medvetet* avstått från en mängd egenskaper som en färdig produkt måste ha. Detta har man gjort för att spara kostnader och utvecklingstid.

Det Du gör i en projektkurs eller i ett examensarbete är förmodligen en funktionsprototyp.

Funktionsprototyp - Produkt

Man måste vara *tydlig* med vad som *ingår*, och vad man *avstått* ifrån. Det man avstått ifrån måste upplevas som okontroversiellt.

Till exempel kan man visa en funktionsprototyp på ett ”stort” kopplingsdäck, samtidigt som man har ”beräknat” att produkten skulle få plats på ett 5×5 mm kretskort. Alla känner till att elektronikprodukter kan miniaturiseras.

Det Du gör i en projektkurs eller i ett examensarbete är förmodligen en funktionsprototyp.

Funktionsprototyp

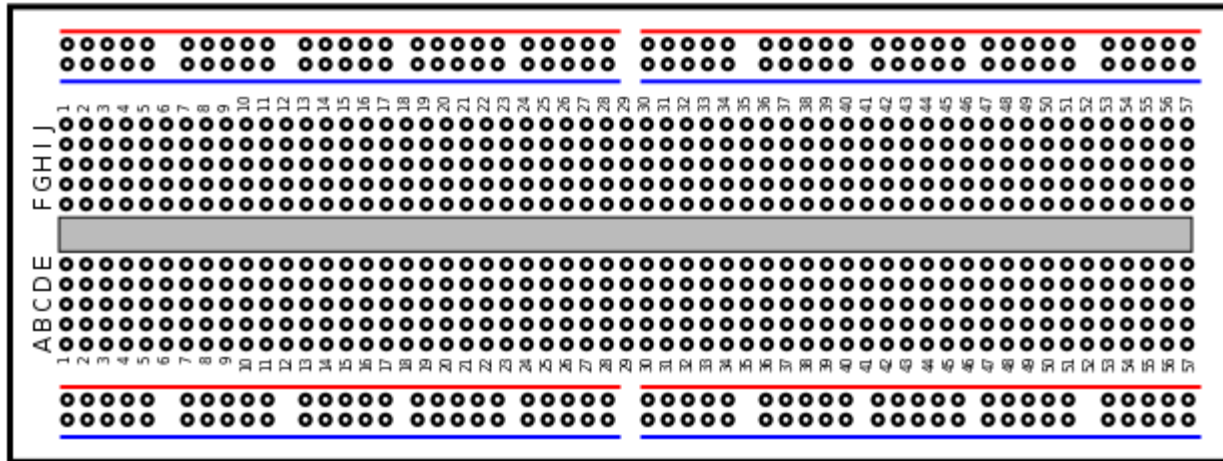
Prototyparbetet innebär att man bygger en verklig krets efter sin design. För att verifiera funktionen, eller för att skapa sig en plattform för felsökning, om det nu skulle visa sig att det inte fungerar.

Man ska alltid bygga och testa en prototyp **steg för steg**, det finns inget värre än att felsöka och rätta symtom som har mer än *en* orsak.

Det är alltid enklast att hitta felen ett i taget!

Allt börjar med att man provar fram funktionen med ett **kopplingsdäck**

Ett kopplingsdäck används för att göra tillfälliga kopplingar för att testa eller pröva en ide. Ingen lödning krävs så det är enkelt att ändra anslutningar och att placera om komponenter.

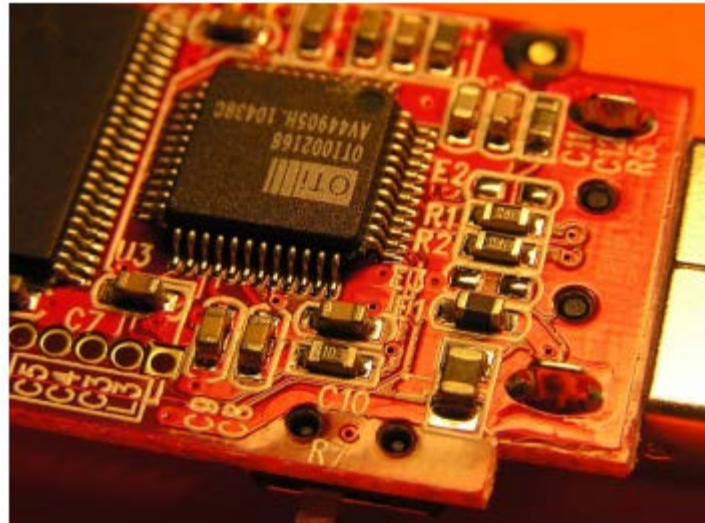


Man kan öva med: **Virtual Breadboard** programvara!

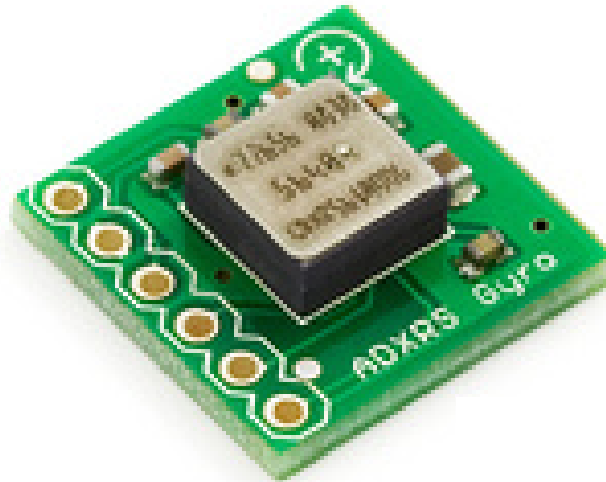
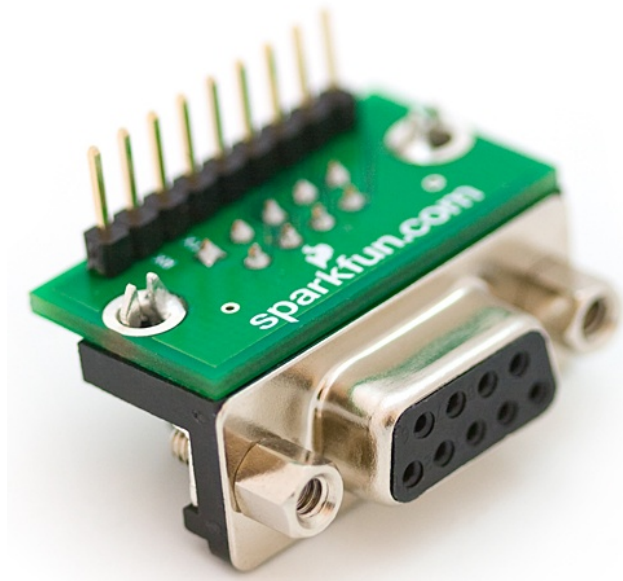
Komponenter som *inte* använder standard elektronikmodulen 0.1” ?

Dagens elektronik består av komplicerade kretskort med ytmonterade komponenter som *inte* använder 0.1” benavstånd.

Hur kan då en icke-expert göra någonting överhuvudtaget?



Breakoutboards!



För prototyparbeten finns det ”breakoutboards” för att underlätta anslutningen av alla slags specialkomponenter till 0.1” pinn avstånds ”standarden”.

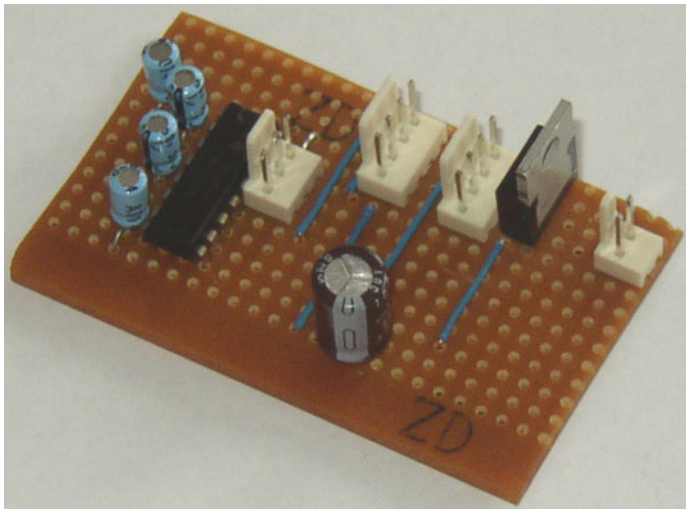
När lösningen är klar?

Till projektkurser och examensarbeten behöver man ofta en **stabilare** kretslösning än vad kopplingsdäcket kan ge.

- Man kan flytta över funktionen till ett **stripboard**

Man monterar och **löder** komponenterna på ett stripboard. Det är ett mönsterkort med 0.1" (2,54 mm) hålraster och parallella ledningsremсор.

Tvåra anslutningar gör man med kopplingstrådar (jumper wires), och avbrott i ledningsremсорna gör man, där det behövs, med en spår-fräs.



OBS!
Detta behöver
Du *inte* göra
till denna kurs.

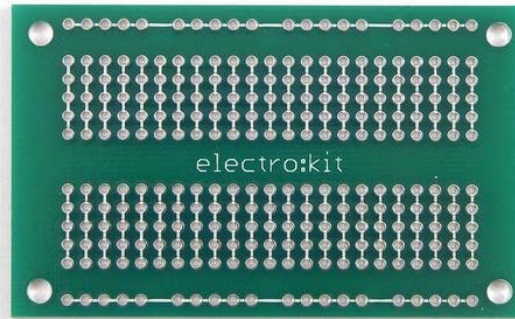
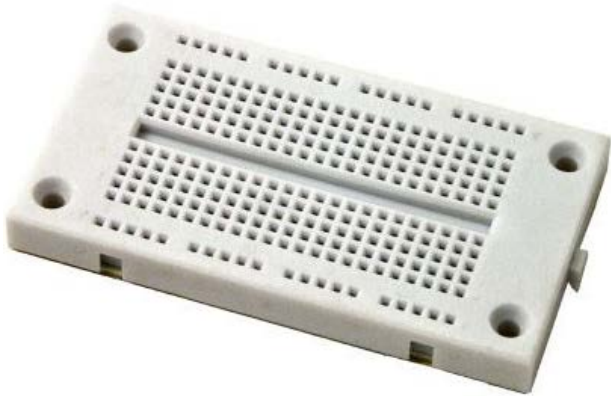


http://www.zen22142.zen.co.uk/Prac/vero_circ/vero.htm

<http://www.printedcircuitsboards.com/software/88/veroboard-stripboard-software.html>

- Man kan flytta till ett ”Kopplingsdäck PCB”

PCB = Printed Circuit Board

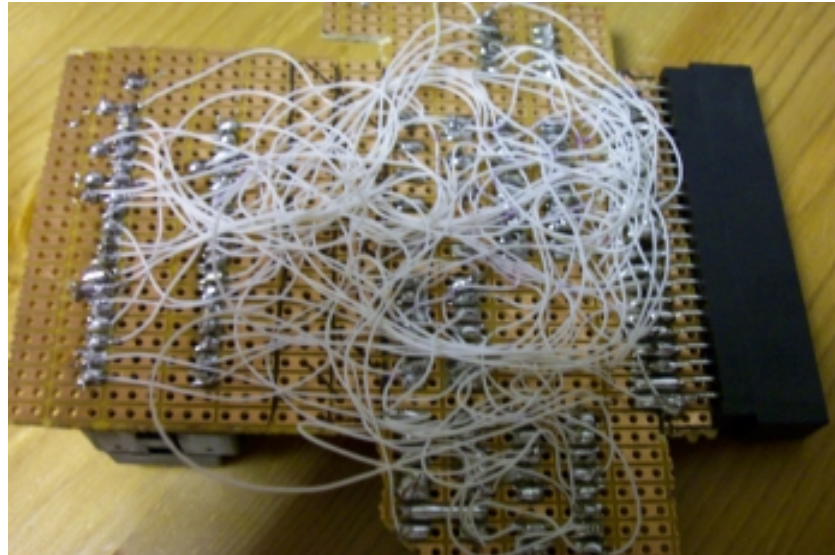


OBS!

Du behöver *inte* göra något kretskort till denna kurs.

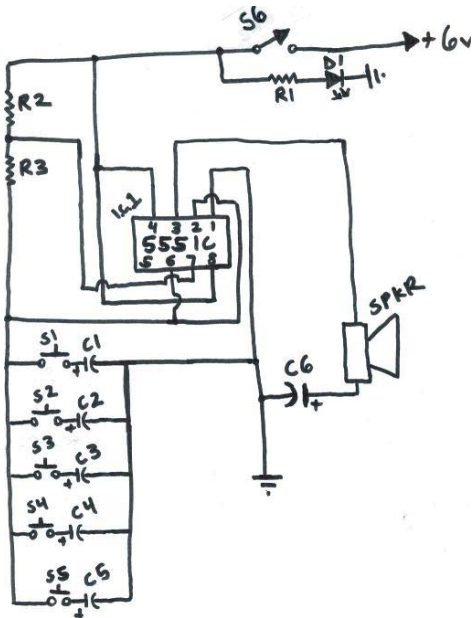
Det finns färdiga mönsterkort som har **exakt samma layout** som ett kopplingsdäck! Först provar man ut sin krets på ett kopplingsdäck, och sedan när man har en slutgiltig layout, löder man komponenterna på samma sätt på mönsterkortet!

Hur kan en elektronikingenjör hålla reda på alla ledningar?



Enkla och användbara hjälpmedel?

- PAD Pencil Aided Design

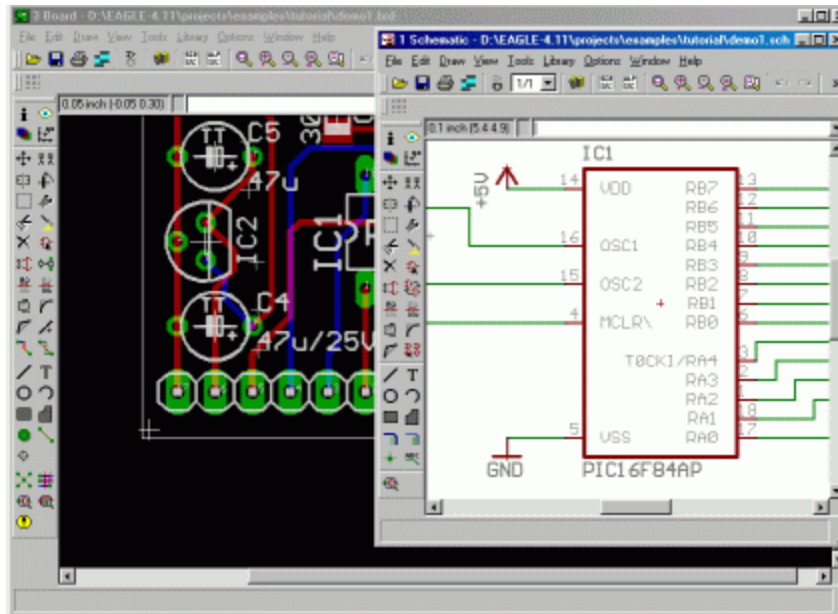


- ASCII-grafik

```
| RA2  16F628  RA1 | ->- PWMGND 1/0
| RA3                RA0 |
| RA4                RA7/OSC1 |
| RA5/MCLR  RA6/OSC2 |
GND ---| Vss                Vdd | -- +5V
| RB0/INT  (RB7)/PGD |
| RB1/Rx   (RB6)/PGC |
2688 Hz | RB2/Tx                RB5 |
PWM  -<-| RB3/CCP  (RB4)/PGM |
|-----|-----|
Use LP-filter to se sinus
*/
```

- *Men elektronikingenjörerna kräver ofta mer*

Det ultimata verktyget är ett elektronik CAD-program



- Board view
- Schematic view

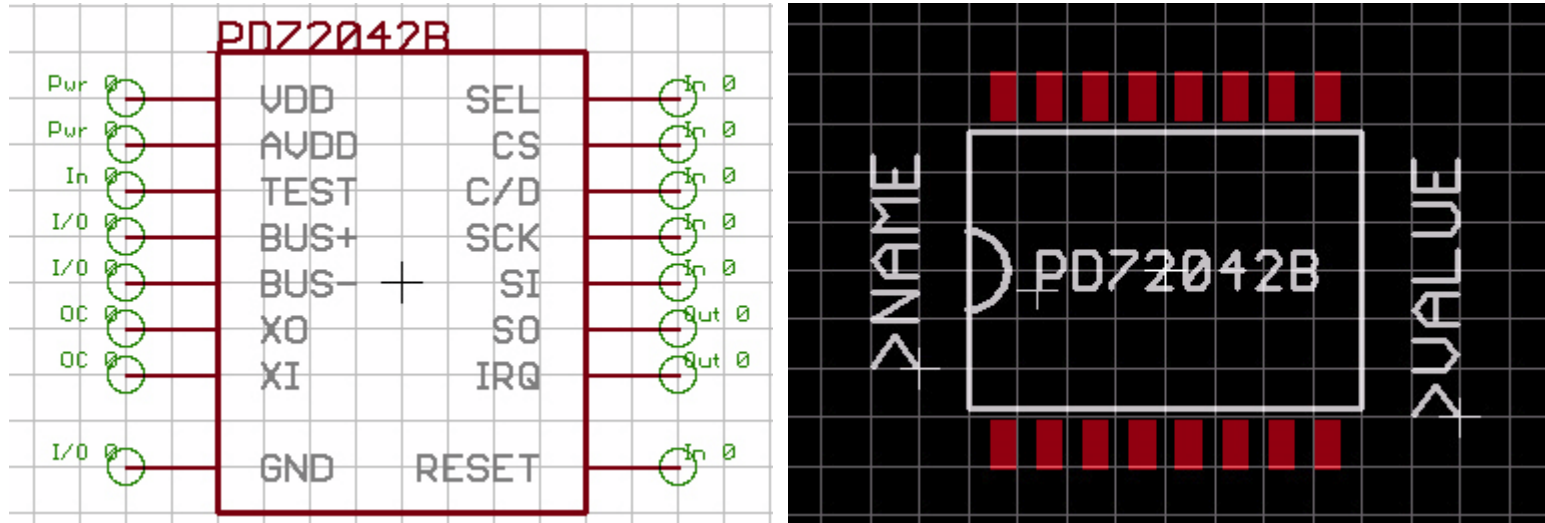
OBS!

Du behöver *inte* använda en elektronik-cad i denna kurs.

Populär elektronik-cad:
EAGLE-CAD

<http://www.cadsoftusa.com/>

Schematic och Board view



Varje elektronik-komponent har en **schema representation** och en **mönsterkorts-representation** (komponentens fotavtryck). De två vyerna är kopplade till varandra.

Om Du markerar en signalledning i schemat, så kommer motsvarande "pinne" i mönsterkortsrepresentationen att också markeras! Lätt att orientera sig.

Board view

Elektronik CADen är det huvudsakliga verktyget för en elektronikingenjör när det gäller att organisera ledningsdragningen av en krets!

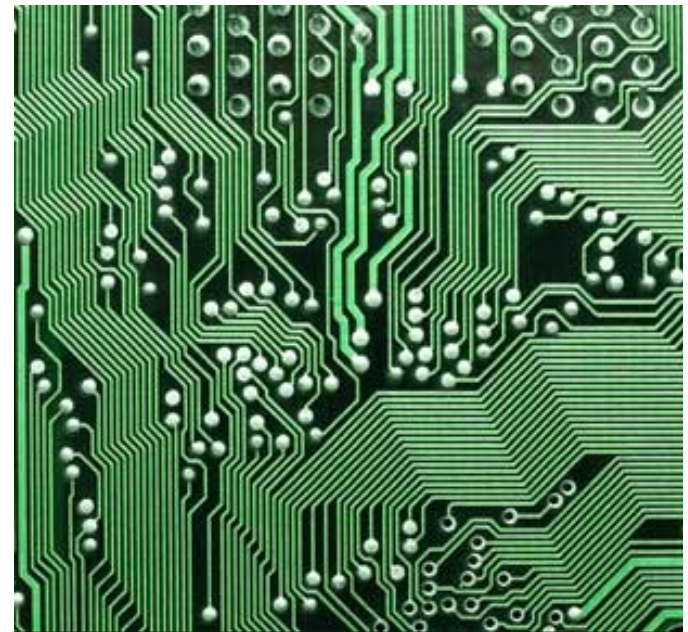
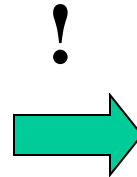
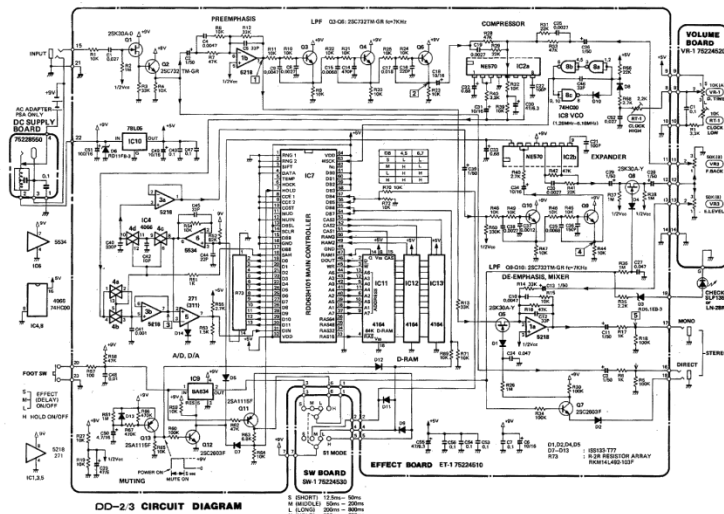
- Men man behöver inte tillverka ett kretskort för att kunna dra nytta av all den information om komponenternas pinn-nummer, footprints, position och inriktning, som man kan få från en ”board view”!

CAD programmet är den naturliga platsen att **samla information** om din krets och dina komponenter!

Det kommer ofta att bli missförstånd om komponentinriktning: upp, ned, vänster, höger, pinn-nummer och så vidare – men allting kan beskrivas i CAD-filen på ett exakt och entydigt sätt!

Ett utprovat kretskort är bevis

För elektronikingenjören så är ett framtaget och utprovat kretskort **beviset** för att det kretsschema man presenterat i en rapport också är den verkliga krets man sedan gjort sina mätningar på. (**Om man nu genomgående använt en Elektronik-CAD**)



- **Vem vågar annars lova att man kopplat allt rätt?**

Kretskort köper man över internet



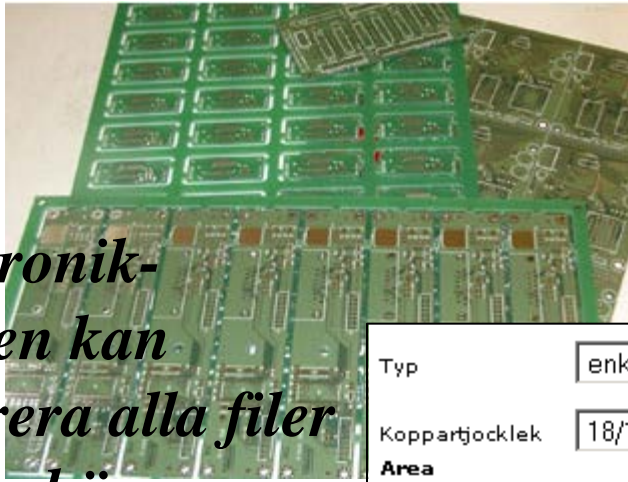
PCB-service Mönsterkort

OBS!

Du behöver *inte* göra något kretskort till denna kurs.

25 x 575mm)
-lager
8/18 (IL 35/35)

*Elektronik-
CADen kan
generera alla filer
som behövs ...*



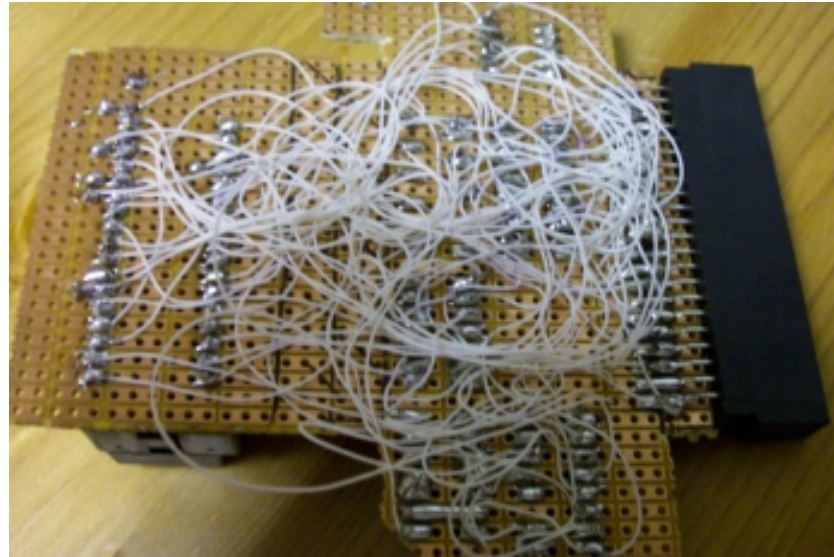
*Köp över
internet!*

Efter genomförd betalning packa dina filer till en zip-fil och sänd den till pcb@electrokit.se. Ange ordernummer i mailet.

Köper man tre kort blir kostnaden samma som när man köper ett(!)

| | |
|----------------|--|
| Typ | <input type="text" value="enkelsidigt"/> |
| Koppartjocklek | <input type="text" value="18/18u"/> |
| Area | |
| Storlek | <input type="text" value="100"/> x <input type="text" value="100"/> mm (max 575 x 425) |
| Antal | <input type="text" value="1"/> st |
| Pris/st | <input type="text" value="1300.00"/> SEK <input type="button" value="Köp"/> |
| | inkl moms |

Så nu vet Du hemligheten bakom
hur en elektronikingenjör kan
hålla reda på alla ledningar?



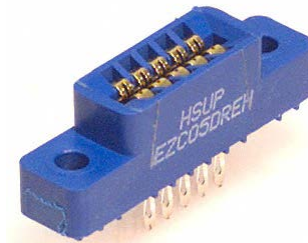
William Sandqvist william@kth.se

Vad är en Mockup?

Mockups är *icke-funktionella* PCB's som används för att visa det fysiska formatet på ett kretskort som ännu inte finns. En mockup har bara något fåtal komponenter för att prova ut var kontakter bör placeras, och för att visa hur stora de mest skrymmande komponenterna är. Komponenterna löds fast på ett mönsterkort med 0.1" hålraster som bara har lödöar.

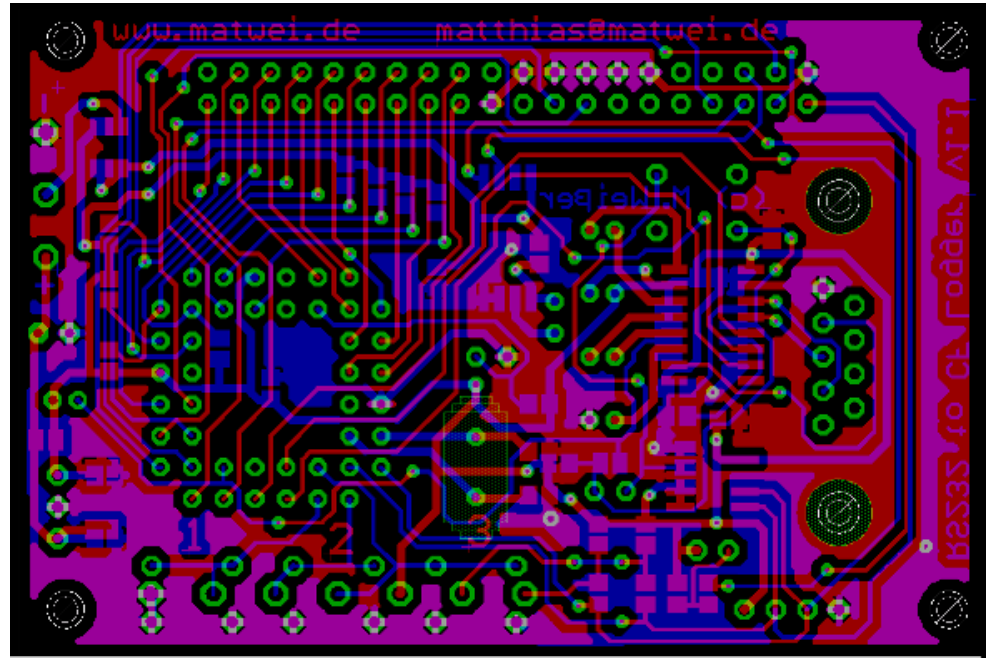
OBS!

Du behöver *inte* göra någon mockup till denna kurs.



Virtual Mockup

Det finns program som kan konvertera en CAD-fil till en 3D bild! Detta kan användas för att visa storlek, utseende, komponent-placering lång tid innan den verkliga produkten finns framme!

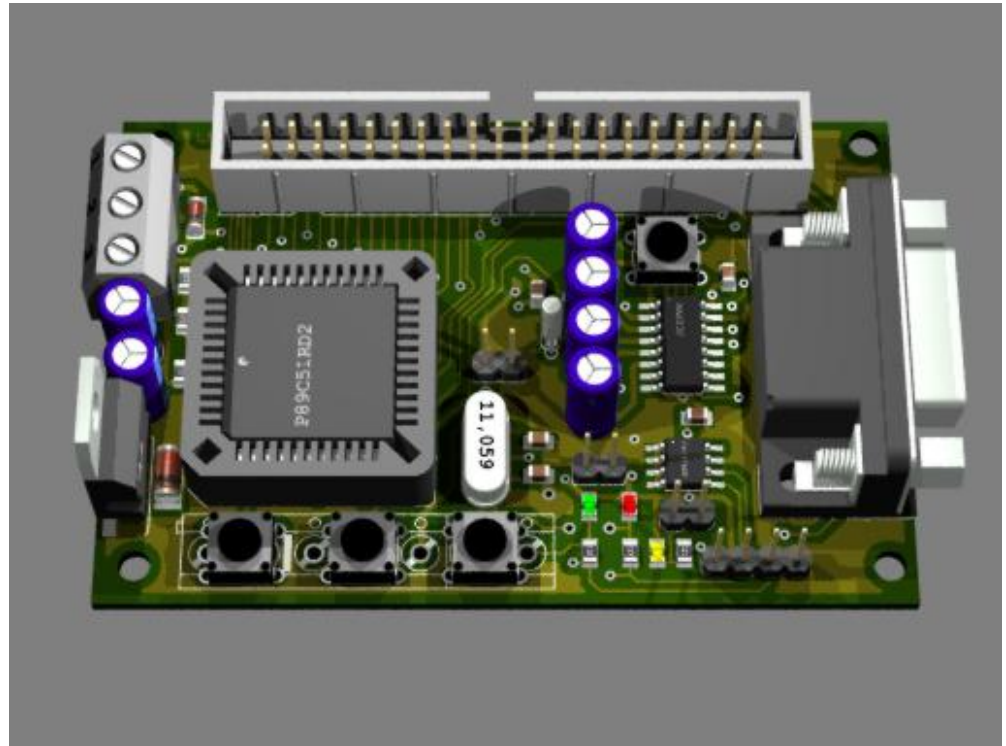


<http://www.matwei.de/doku.php?id=en:eagle3d:eagle3d>

Virtual Mockup

Det finns program som kan konvertera en CAD-fil till en 3D bild!

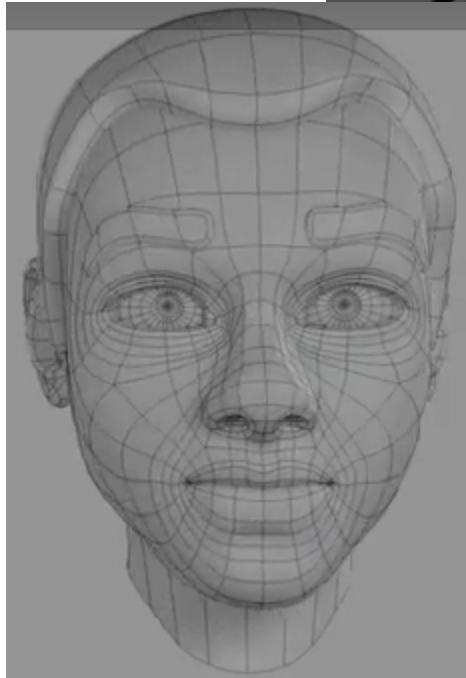
Detta kan användas för att visa storlek, utseende, komponentplacering lång tid innan den verkliga produkten finns framme!



<http://www.matwei.de/doku.php?id=en:eagle3d:eagle3d>

Mockup av en produkt

3D-CAD



Ett vanligt Mockup-material är OASIS-skumplast



LEGO-fräs

En mockup visar hur det kommer att se ut – men i ett material som saknar hållfasthet.

<http://www.youtube.com/watch?v=Cf6mWQ8QreA>

Funktionsprototyp + Mockup

En funktionsprototyp tillsammans med en mockup gör att man slipper att direkt ta fram en färdig produkt.

Detta är ofta tillräckligt för att övertyga andra (tekniker/finansiärer) om att man har en hållbar idé.

William Sandqvist william@kth.se