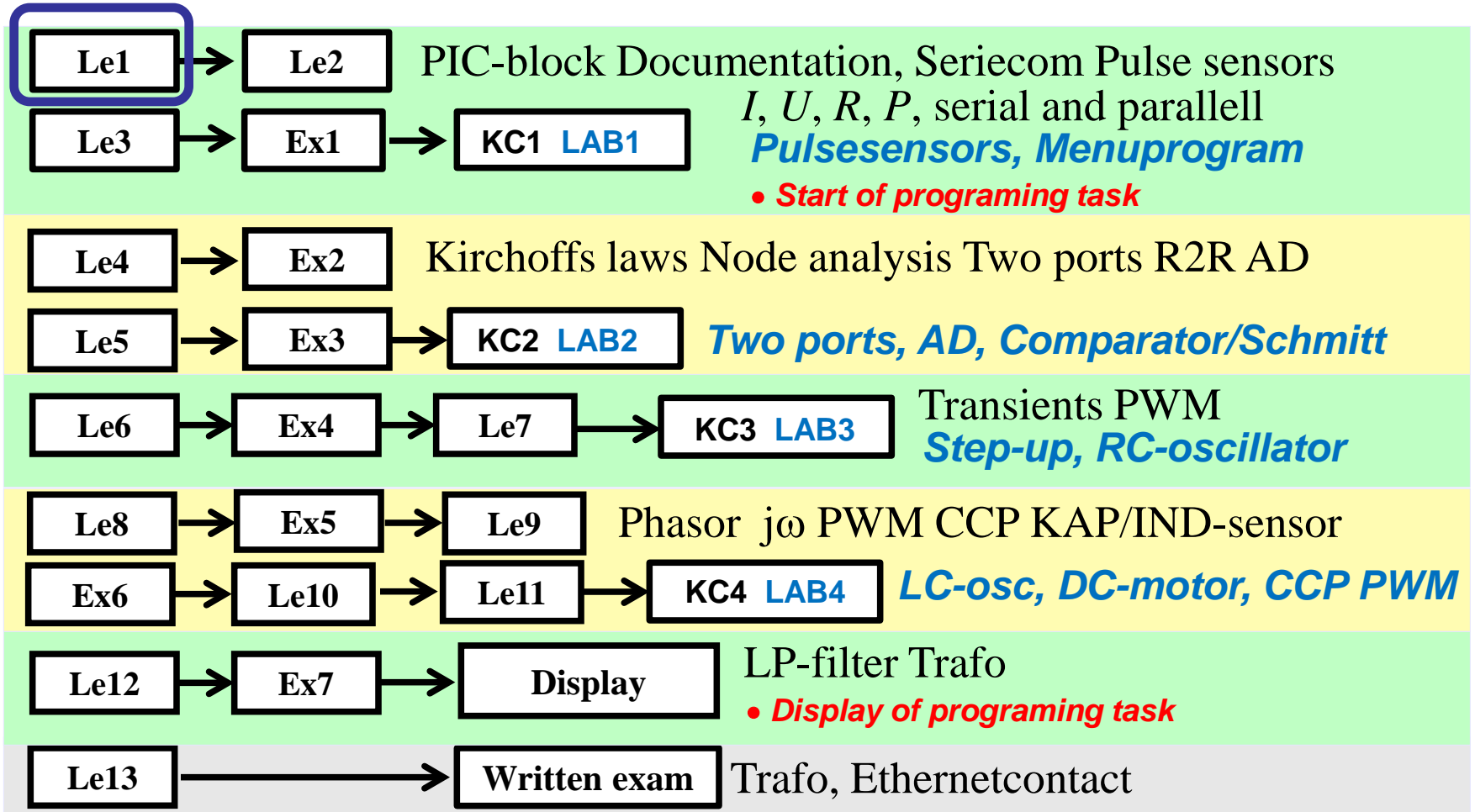


# IE1206 Embedded Electronics



Need a computer engineer to know anything about the **reality** that the data reflects?



# *Not only general databases...*

More than **98%** of the world's processors today sits in embedded systems.

Embedded systems are used everywhere, in satellites, robots, cars, planes and phones, down to the smallest radio, elevator, laundry or dishwashing machine.

# Inbyggda system – computer programs in products

In order to write a good program, you simply has to understand the product.



*Husqvarna robot vacuum cleaner -  
An example of an embedded system*

Often a development team of employees with different skills are needed.

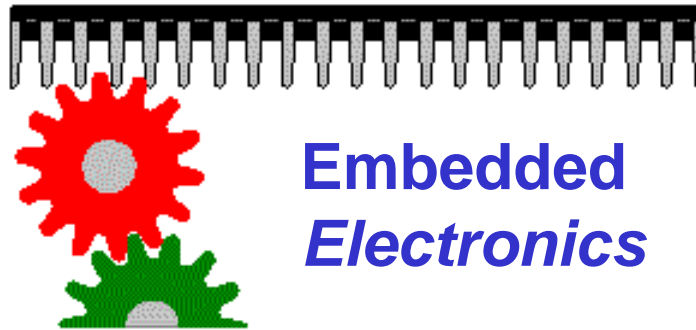
# You as a programmer ...

*In the development team the programmer often interacts with mechanical hardware and electronics technicians.*

*The programmer of the embedded system is usually the one who gives the product the most of the function.*

William Sandqvist [william@kth.se](mailto:william@kth.se)

# Course **Embedded Electronics**



*The course **Embedded Electronics** provides a practical foundation for future project courses and your thesis.*

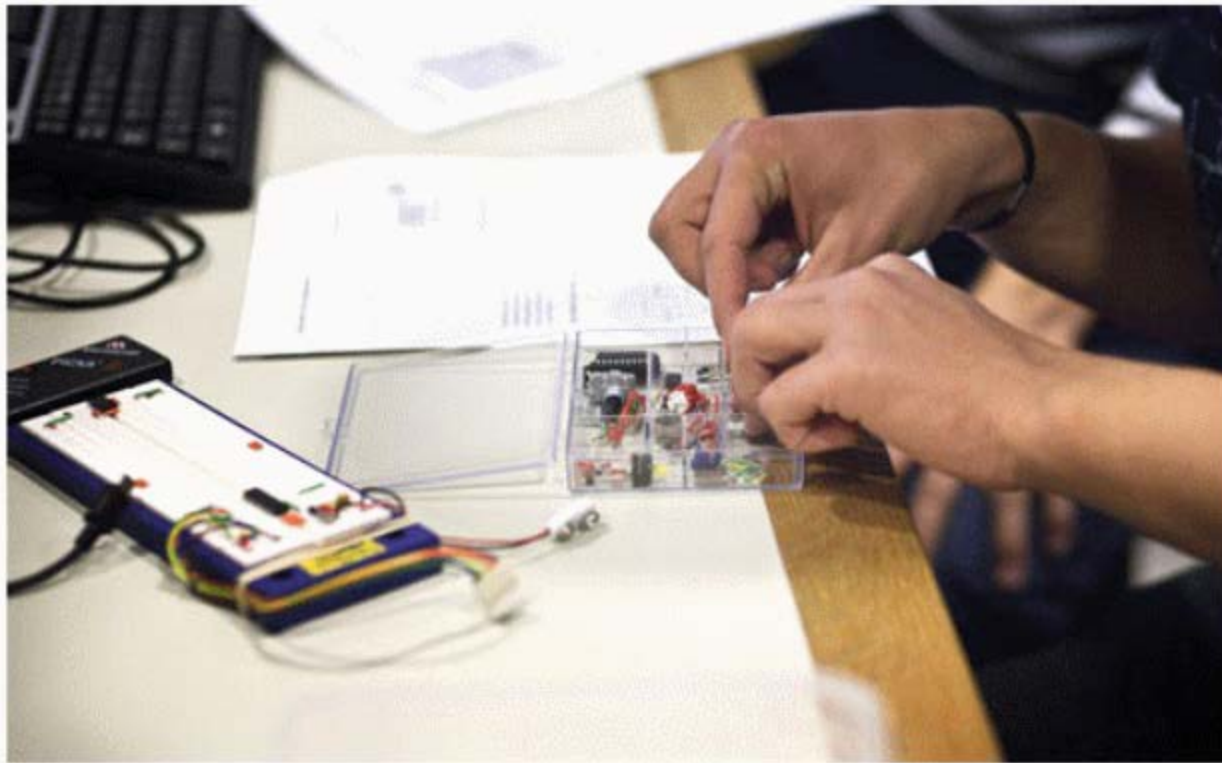
*The course will familiarize you with electro technician's terminology and reasoning, so that you can actively interact, as a valuable member of a development team for products with embedded processors.*

# Laborations – hands on

The laboratory course teaches both the programming of a PIC processor and electricity-electronics circuits!



# We learn **electronics** with the school's smallest processor!

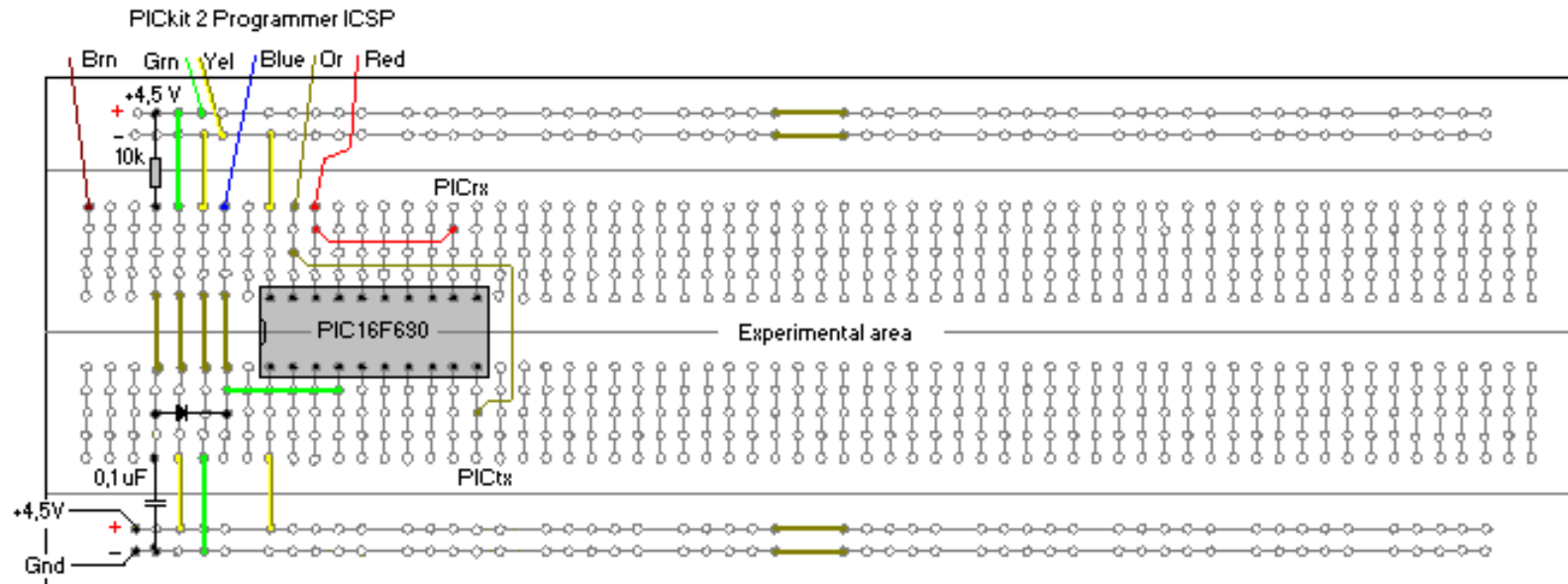


William Sandqvist [william@kth.se](mailto:william@kth.se)

# Simple development system!



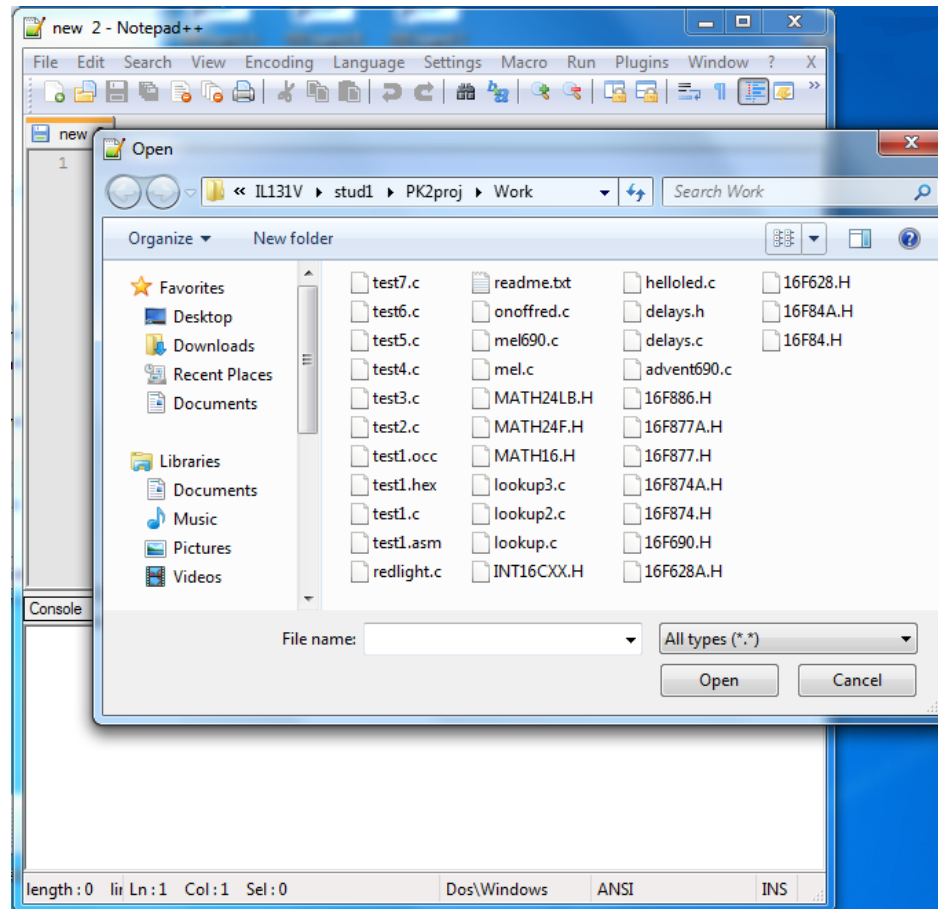
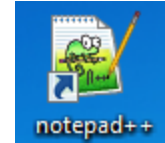
PIC 16F690 8-bit  
microcontroller



4.5V Battery

Orange and Red jumpers are for PICkit2 UART Tool

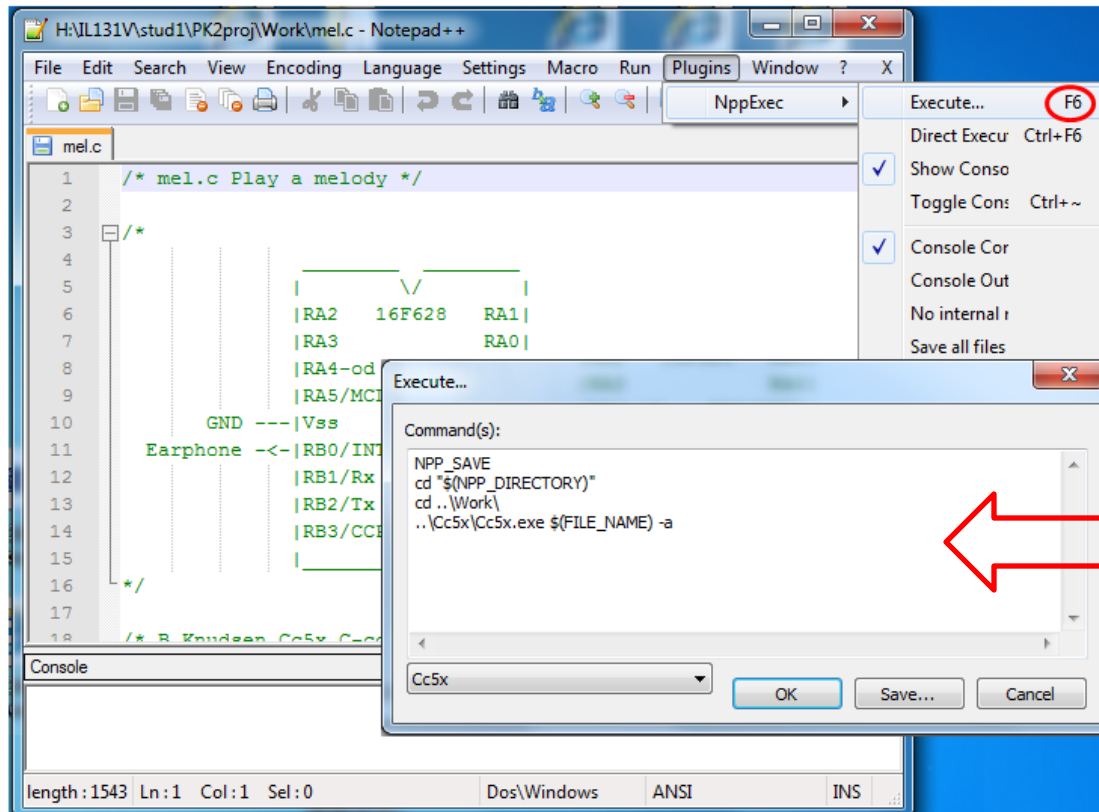
# Texteditor Notepad++



# Compiler Cc5x



- the compiler is invoked from inside the texteditor **Notepad++**.



A script starts the compiler

# Chip programmer PICKit 2



## PICKit2

Stand alone program for the chip programmer

To download code to chip



*Fast download of code to chip!*

The screenshot shows the PICKit 2 Programmer software window. The title bar reads "PICKit 2 Programmer". The menu bar includes File, Device Family, Programmer, Tools, View, and Help. The main window is titled "Midrange/Standard Configuration" and displays the following information:

- Device: PIC16F628A
- Configuration: 21FF
- User IDs: FF FF FF FF
- Checksum: 19FF
- OSCCAL: (blank)
- BandGap: (blank)

A yellow status bar indicates "PICKit 2 found and connected. PIC Device Found." To the right of this bar is the MICROCHIP logo. Below the status bar are several buttons: Read, Write, Verify, Erase, and Blank Check. To the right of these buttons is a section for "VDD PICKit 2" with checkboxes for "On" and "/MCLR", and a numeric input field set to "2,5".

The "Program Memory" section is checked "Enabled" and set to "Hex Only". The "Source" is set to "None (Empty/Erased)". Below this is a table of memory addresses and their corresponding hex values:

Address	Hex 1	Hex 2	Hex 3	Hex 4	Hex 5	Hex 6	Hex 7	Hex 8	Hex 9
000	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF
008	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF
058	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF

The "EEPROM Data" section is also checked "Enabled" and set to "Hex Only". Below this is a table of EEPROM addresses and their corresponding hex values:

Address	Hex 1	Hex 2	Hex 3	Hex 4	Hex 5	Hex 6	Hex 7	Hex 8	Hex 9	Hex 10	Hex 11	Hex 12	Hex 13	Hex 14	Hex 15
00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
10	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
20	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
30	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

On the right side of the EEPROM Data section, there are three buttons: "Auto Import Hex + Write Device", "Read Device + Export Hex File", and the PICKit 2 logo.

William Sandqvist [william@kth.se](mailto:william@kth.se)

# Programming task in groups of two-three ...

Programming assignment should give you a practical experience of basic electronic circuit work. This is useful knowledge for project courses and thesis.

Programming assignment is presented with a demonstration/presentation and a written "mini report".

# Technical report?



# Report on software for an embedded system

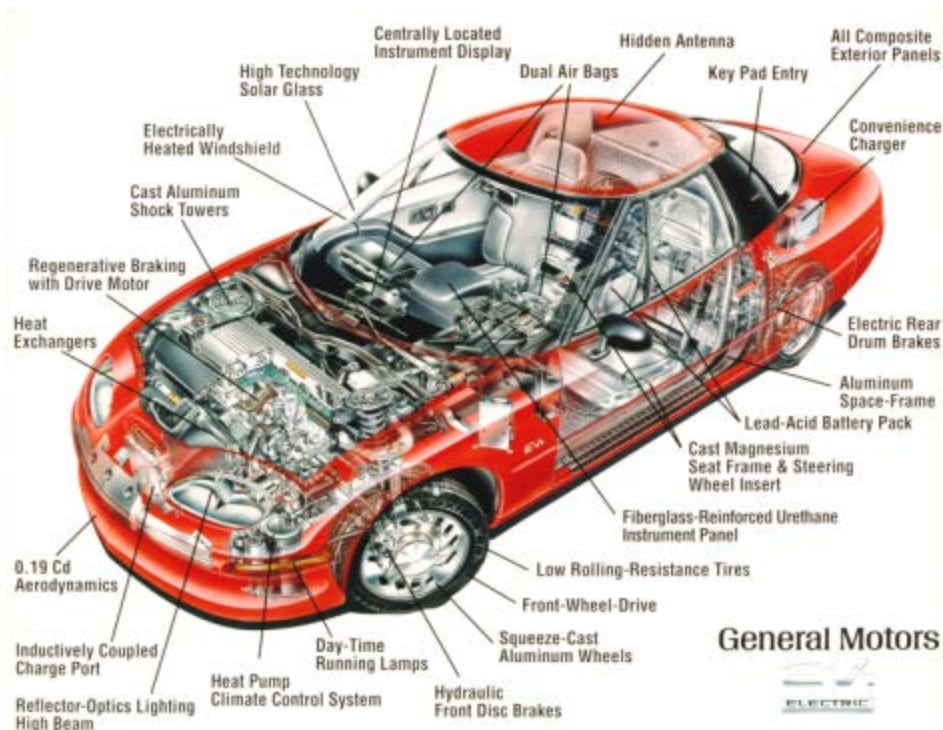
## You document:

- System structure
- The scope of the program
- Program structure
- How application/function is to be tested

Technical reports contains "figures" and text interaction. The figures are often different types of charts.

# ● Assembly Image

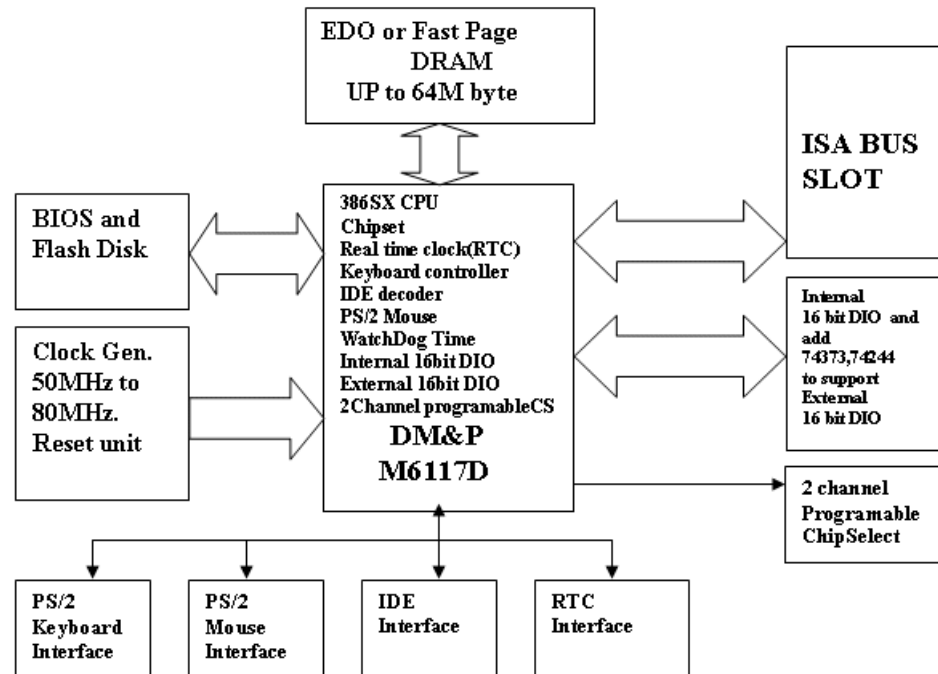
An embedded processor is built into a product. It is hardly possible to describe the processor computer program without first at least briefly **explain the product!**



An assembly image defines what the different parts are called - you then use these exact **names** consistently throughout the report when you need to refer to the product.

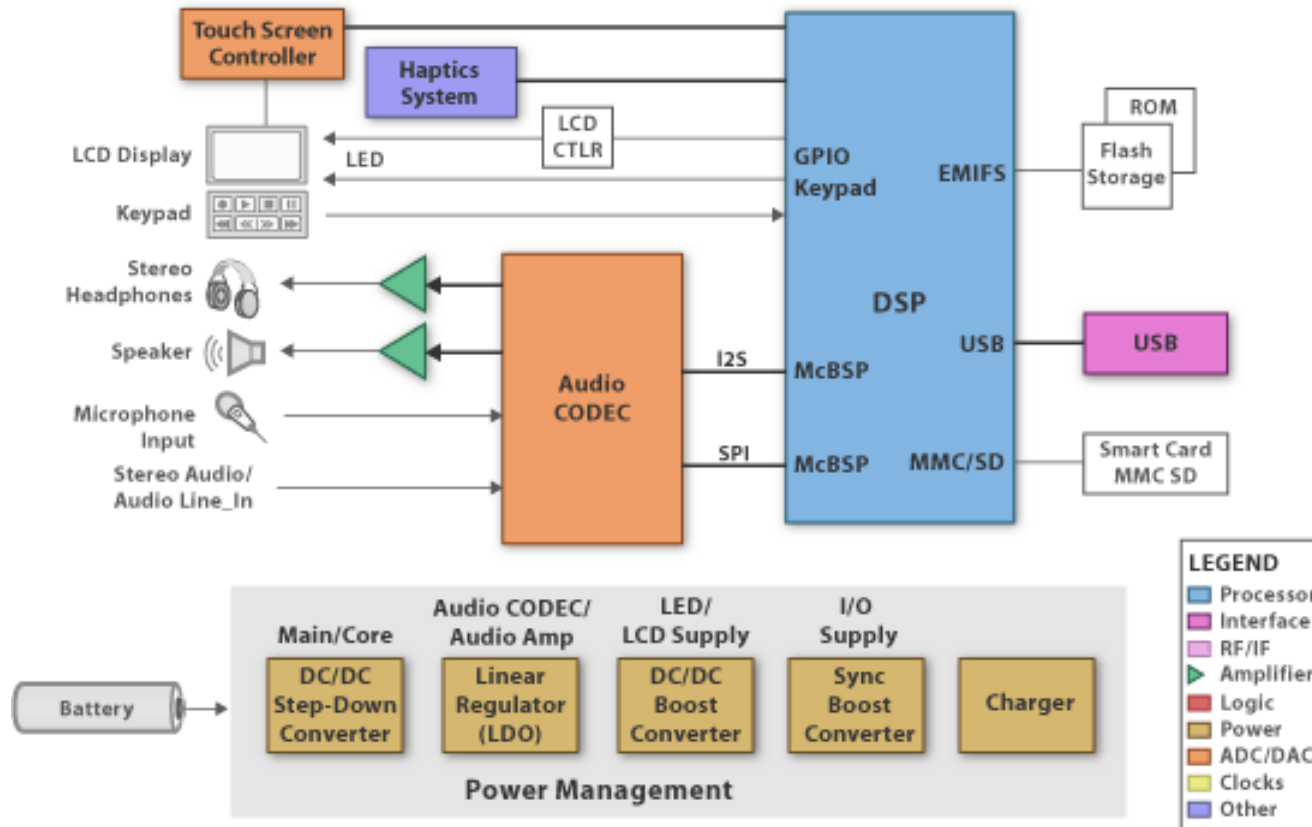
# ● Block diagram

A block diagram is a diagram of a system in which the principal components, or functions, are represented by block interconnected with arrows showing the relationships of the blocks.



It requires no advanced skills or tools for drawing a block diagram. This is a block diagram of the product components. Other chart types are used to describe the program structure.

# Illustrated block diagram



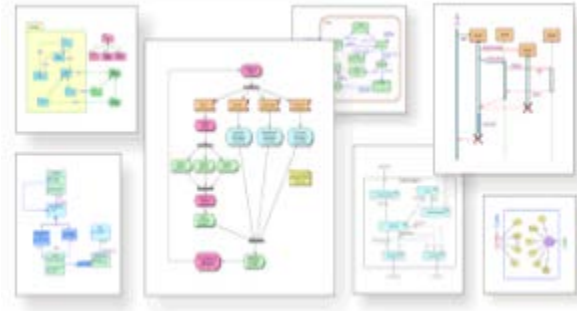
If you turn to readers outside the group of technicians, it often happens that the blocks are colored and illustrated ...

# Chart types for program code?

How to describe your program  
as a picture?

# (There is a standard, **UML** )

**Unified Modeling Language (UML)** is an object-oriented general language for modeling all types of systems.

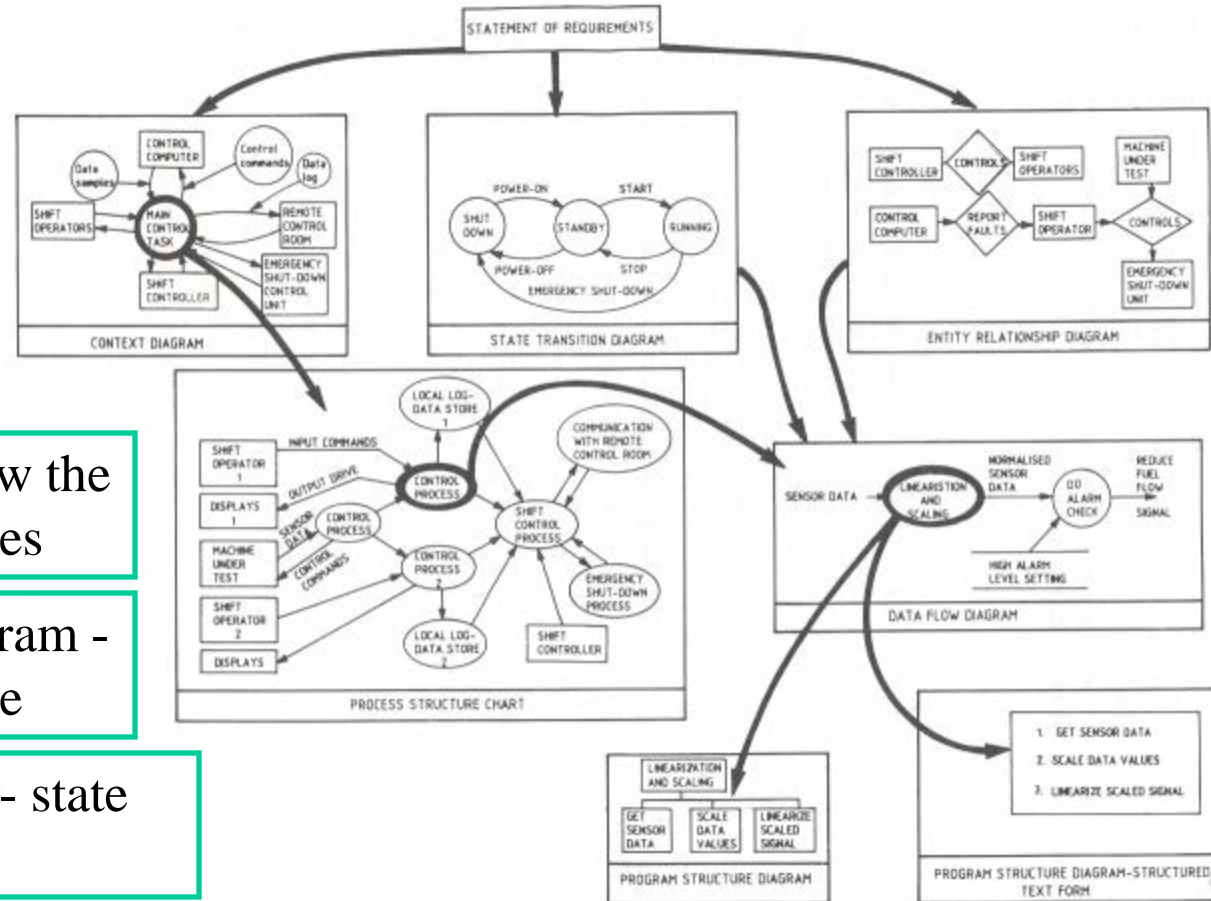


*UML - contains a variety of chart types.*

The language used primarily in software design, but is also useful in other fields such as business process modeling and more. By creating a model of the system to be constructed it will be easier to understand and build it.

***But ... UML is too extensive for this course...***

# Chart types for software

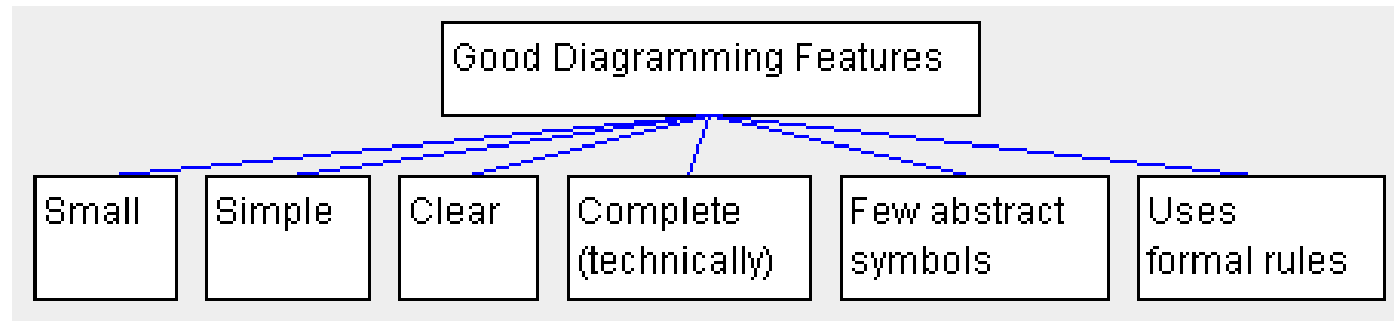


Three common chart types:

- Flowchart - how the processor executes
- Structural diagram - program structure
- State Diagram - state machines

The arrows show how several different chart types interact and may be required to provide a holistic picture.

# Good chart properties



A good chart should be small, simple, clear, contain few abstract symbols, and follow formal rules.

**A good chart is helpful when to explain the system to others.**

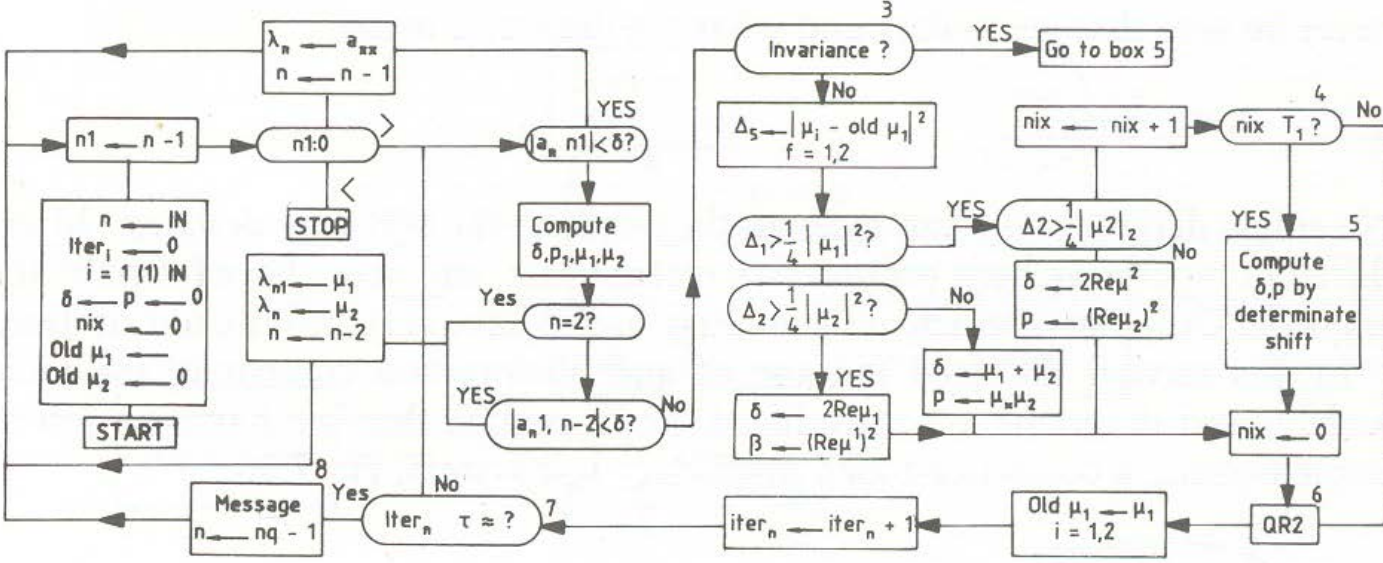


# Example of a chart that *breaks* all the rules!

Good chart properties

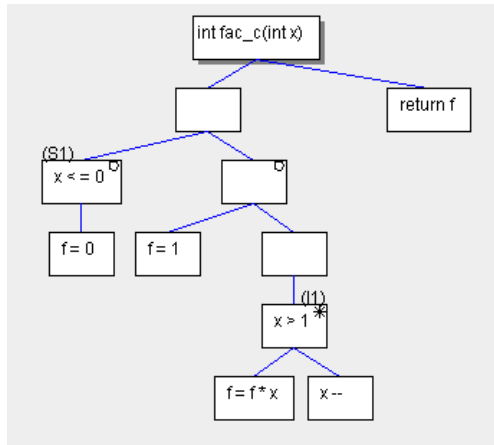
Good Diagramming Features

Small	Simple	Clear	Complete (technically)	Few abstract symbols	Uses formal rules
-------	--------	-------	------------------------	----------------------	-------------------



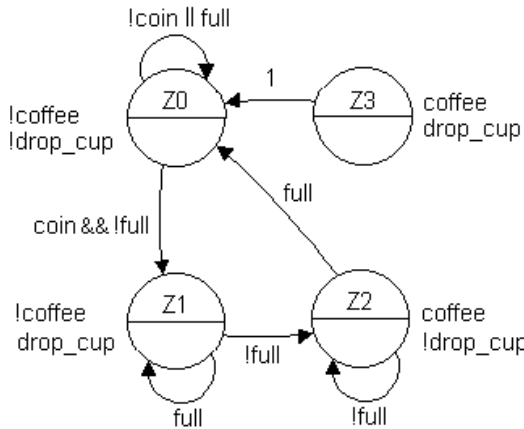
So, the diagrams are only helpful  
if you make an effort when  
drawing them ...

# We will use ...



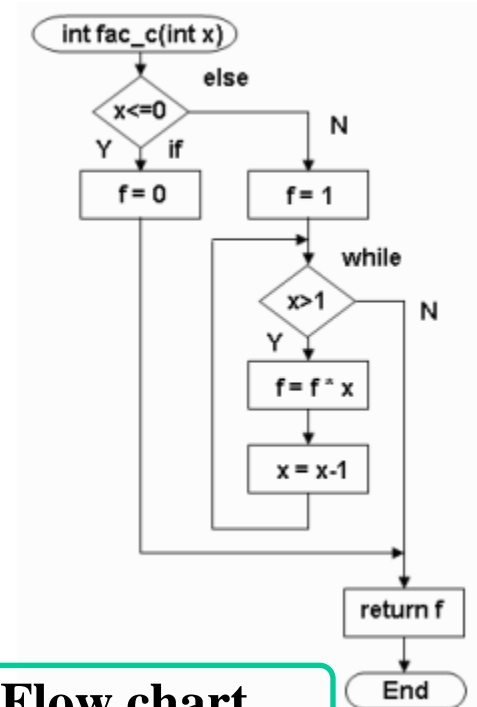
## Structure chart

– program structure



## State chart

– for statemachines



## Flow chart

– how the processor executes

• *See exercise 1!*

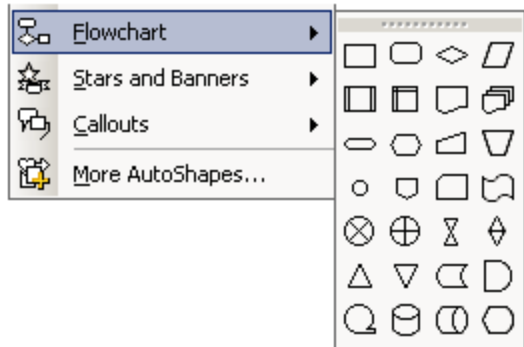
• *See Digital Design!*

• *See Datorteknik!*

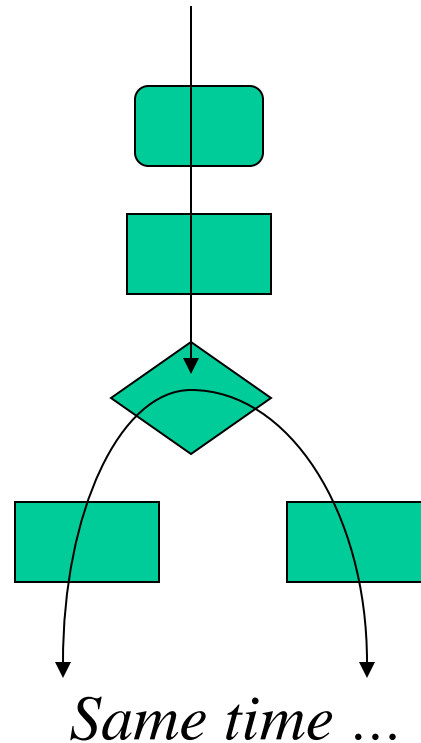
# ● Flow chart

Flow chart is used when it is important to show exactly how the processor executes code. For example, when analyzing assembly code to ensure that different paths through the program takes the exact same time.

## *Programflow*



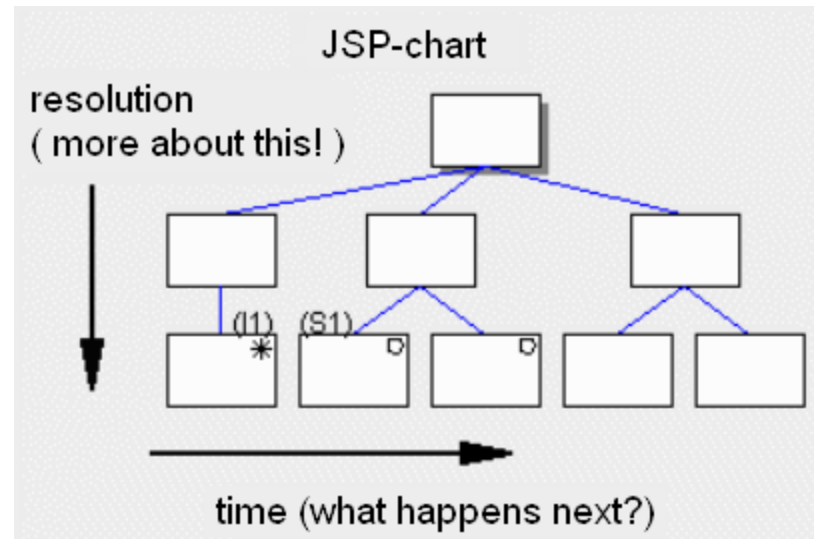
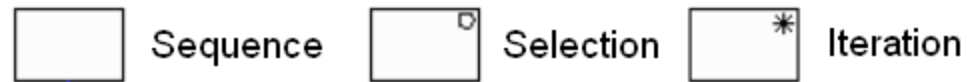
Tool to draw flow charts could be found in Word and PowerPoint.



! →

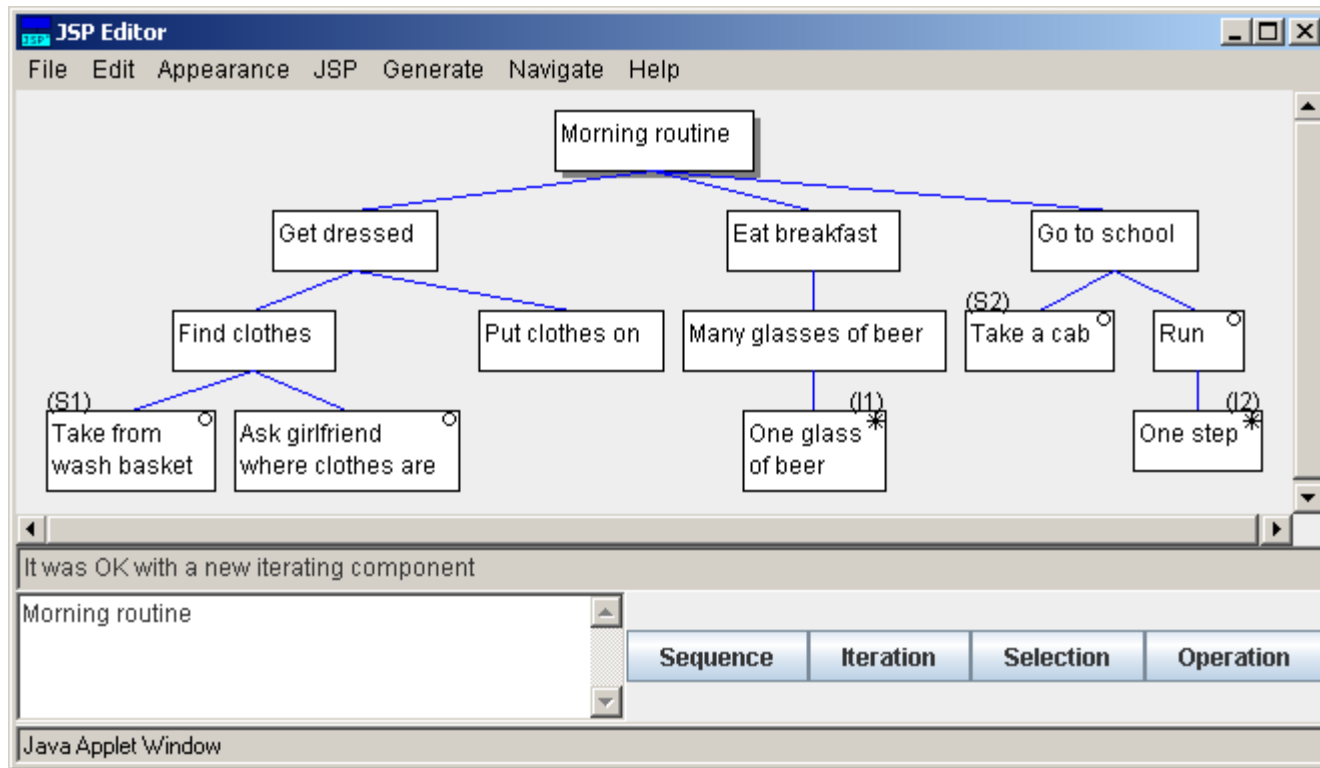
*But the program structure is better described with a Structure chart.*

# • Structure chart



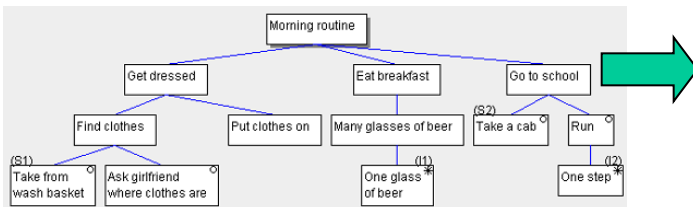
Structure diagram is an easy to draw (just three symbols) two dimensional chart (dimensions: time, and resolution).

# Structure chart editor



*Take the help of the JSP editor when you plot structure charts!*

# • C-code shell from the chart



From the structure chart, one can directly obtain a C code shell - a good beginner's tool!

```
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 */
}
```

# Knowledge control before 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>b) JSP-Strukturdiagram, svars kod</b>	
Calculate	Svars kod
	<input type="text"/>

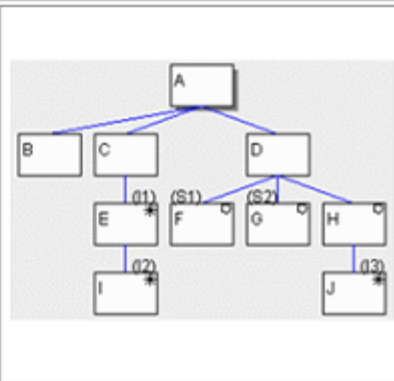


# Knowledge control before Lab1

## 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"/>



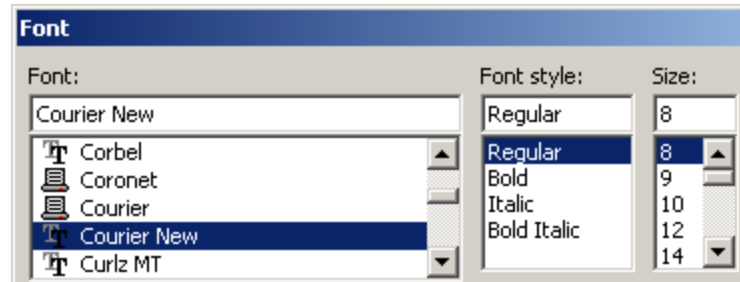
Tryck på knappen för att beräkna din svarskod. *Skriv ner dina svarsnummer inför slutredovisningen*

b) JSP-Strukturdiagram, svarskod	
Calculate	Svarskod
	<input type="text"/>

*Fortsätt själva ...*

# • Publishing software code

Program code are written with a font that has a fixed width ("typewriter style"). Normally fonts have variable width: "i" is more narrow than "W".



**Courier New**, is one such "fixed" font. Have you used 80 characters per line in the text editor, when publishing size should be set to 8 points if one line is to fit on A4. If the lines are broken it makes your code difficult to read. Please note! Tab characters may need to be replaced with a number of spaces.

# Comments

If you write all comments in English you can take advantage of that there are millions of talented PIC programmers around the world.

You can expect to easily find someone who understands your program and who is willing to further develop your code.

If you publish code with English comments it is available to the whole world.

*comments // or /\* \*/ in english.*

# ● ASCII graphics

/\*

```
      |-----|
      | 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
```

\*/

***Important information can be embedded in the application code as ASCII graphics. It will then be available to all regardless of the system.***

***All the course program examples*** has **ASCII-graphics** to show how to "connect" the chip.

# Exemple of ASCII-graphics

```

/*
  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)
  */
  
```



```

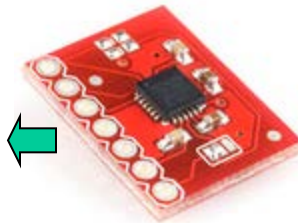
/*
  Board contact
  1 2 3 4 5
  ( . . . . . )
  \ . . . . /
  -----
  6 7 8 9
  1 +7.7V
  2 +7.7V
  3 GoSignal
  4 +5V Vcc
  5 +5V Vcc
  6 Release
  7 Gnd
  8 Gnd
  9 Gnd
  */
  
```

```

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

```

  |-----|
  | ADXRS610 |
  | +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](mailto:william@kth.se)

# Prototyping

# Functional prototype - Product

- Functional prototype (working prototype)

The meaning of a function prototype is to convince others that you have solved the technical/software problems that may exist with the design.

One has deliberately refrained from a variety of properties that a finished product must have. This has been done to save costs and development time.

***What you do in a project course or in thesis work is probably a function prototype of this type.***



# Functional prototype - Product

One must be clear about what is included and what is excluded. Things that are excluded must be perceived as uncontroversial.

For example, one can show a functional prototype with a "big" breadboard, while it is "estimated" that the product would fit on a 5×5 mm PCB. Everyone knows that electronic products can be miniaturized.

***What you do in a project course or in thesis work is probably a function prototype of this type.***

# Functional prototype

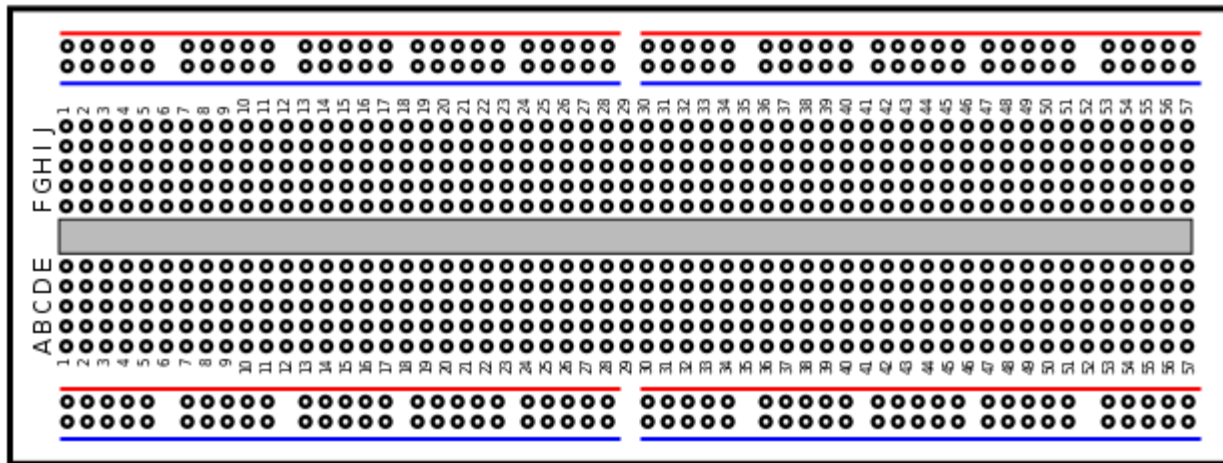
**Prototyping** means building an actual circuit for your design. To verify the function, or to create a platform for troubleshooting, if it turns out that it does not work.

One should always build and test a prototype step by step, there is nothing worse than to troubleshoot and correct the symptoms that have more than one cause.

*It is always easier to find errors one at a time!*

# Everything starts with trying out the the function with a breadboard

A breadboard is used to make temporary connections to test or examine an idea. No soldering is required so it is easy to change the connections and positioning of components.

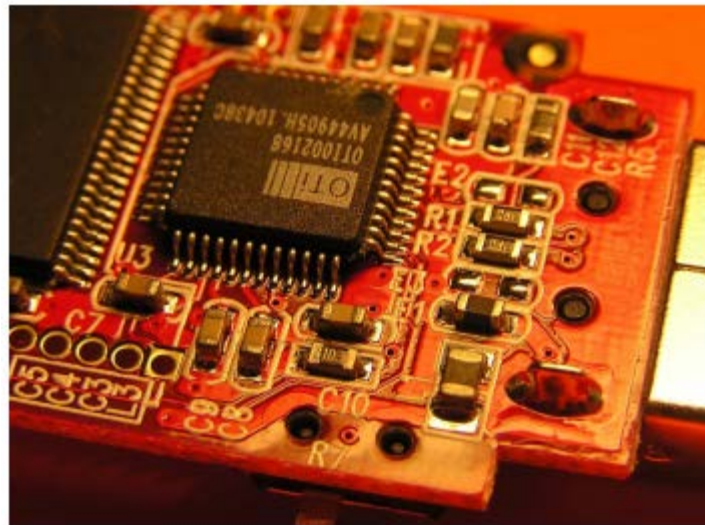


You can practice with: **Virtual Breadboard** program!

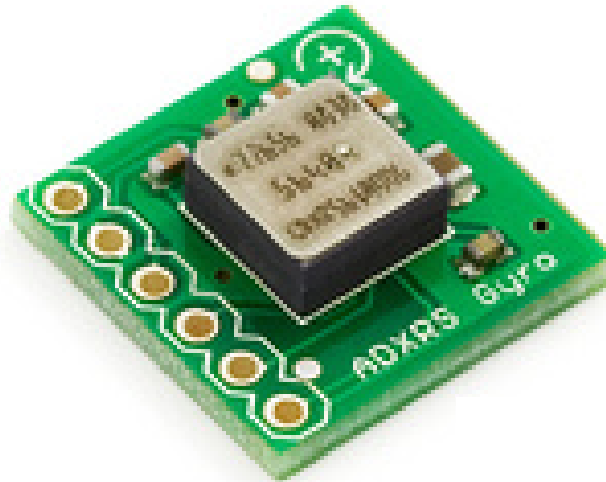
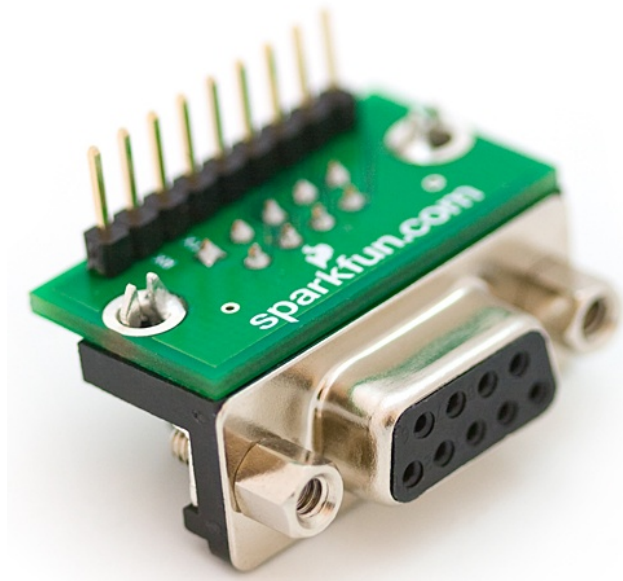
# Components that do not use standard electronics module 0.1“ ?

*Today's electronics consist of complex PCB with surface mount components that do not use 0.1 "leg distance ".*

How can a non-expert then Do anything at all?



# Breakoutboards!



For prototype work there are "breakoutboards" to facilitate the connection of all types of special components to 0.1" pin standard.



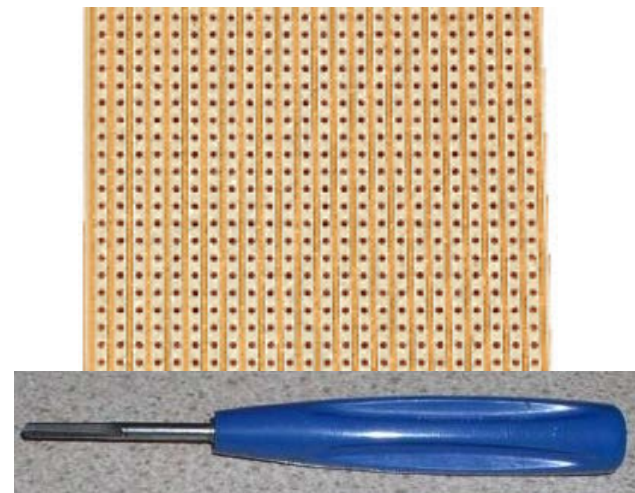
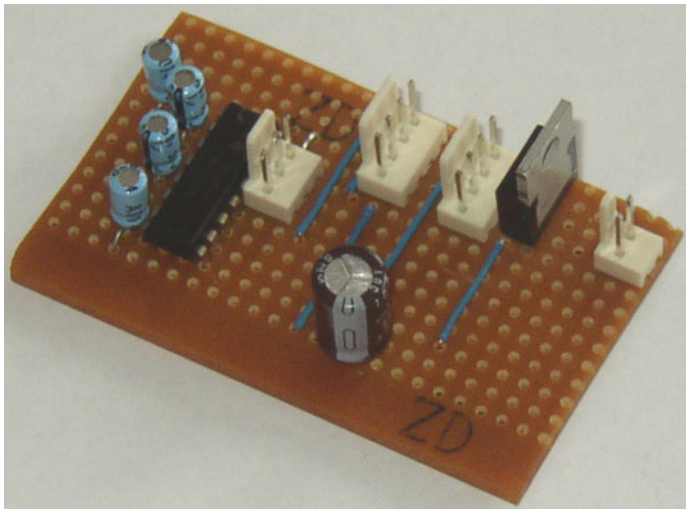
# When solution is complete?

To project courses and thesis projects, you often need a more stable circuit design than the breadboard can provide.

- You can move the function to a **stripboard**

One can assemble and solder components on a stripboard. It is a PCB with 0.1” (2.54mm) pitch pattern of holes and parallel line strips.

Transverse connections are done with wires (jumper wires), and disruption of conduction strips are done, where needed, with a milling tool.

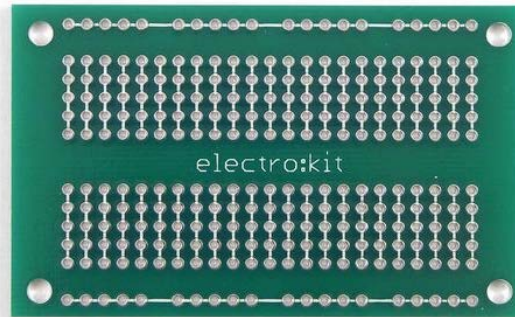
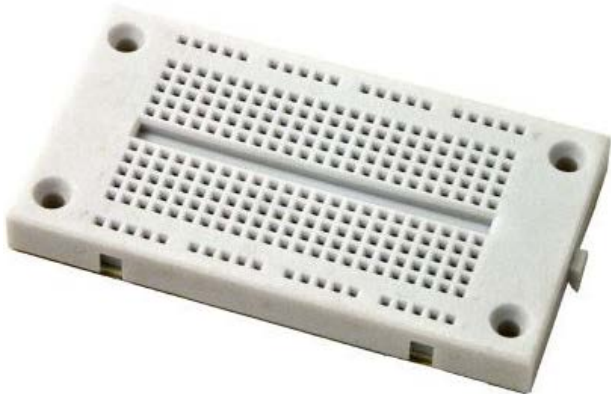


[http://www.zen22142.zen.co.uk/Prac/vero\\_circ/vero.htm](http://www.zen22142.zen.co.uk/Prac/vero_circ/vero.htm)

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

- You can move to a "Breadboard PCB"

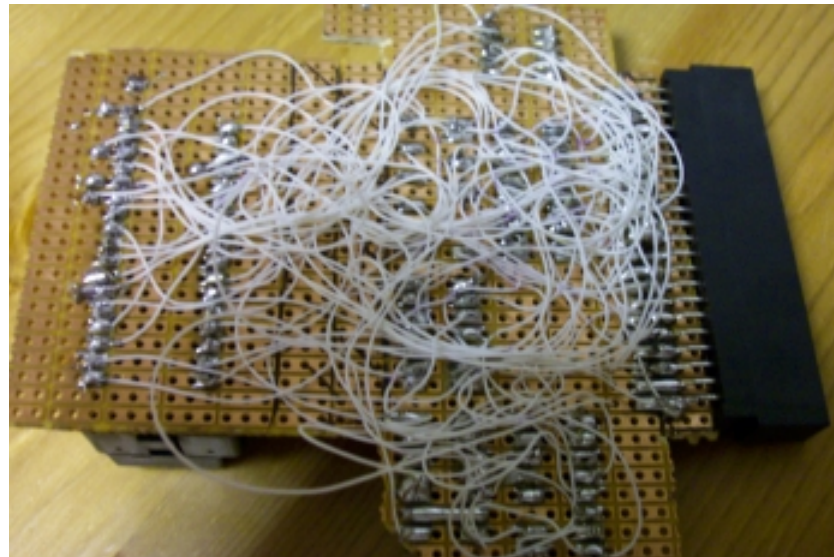
PCB = Printed Circuit Board



There are ready-made circuit boards which have exactly the same layout as a breadboard! First try it out the circuit on a breadboard, and then when you have a final layout, solder to the components on the same way on a PCB!



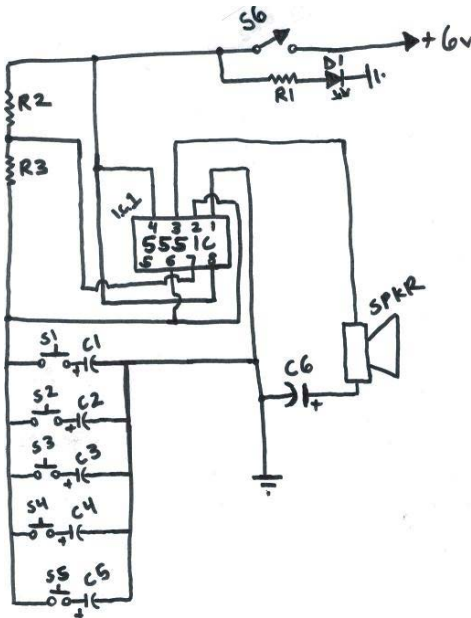
# How can an electronics engineer keep track of all wires?



# Simple and useful tool?

- PAD Pencil Aided Design

- ASCII-graphics



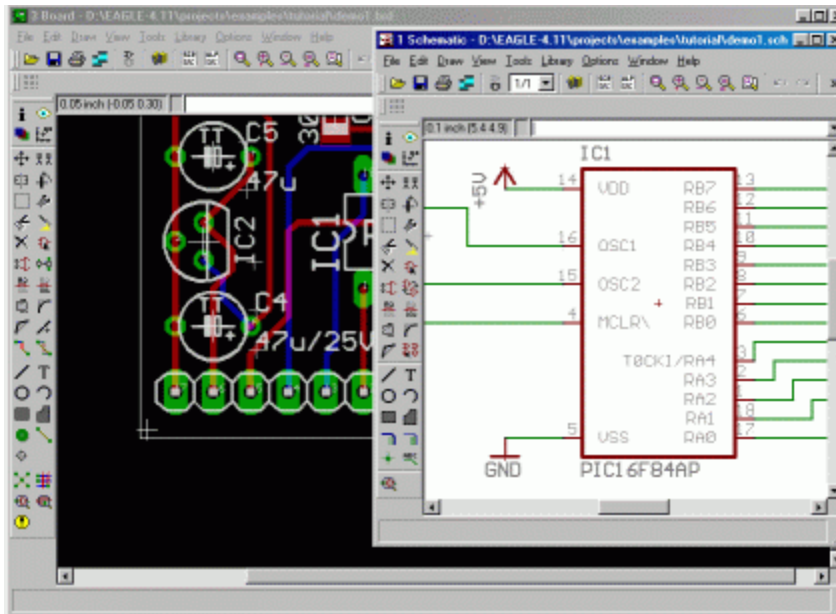
```

| 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
*/

```

- *But electronics engineers often require more*

# The ultimate tool is an electronic CAD software

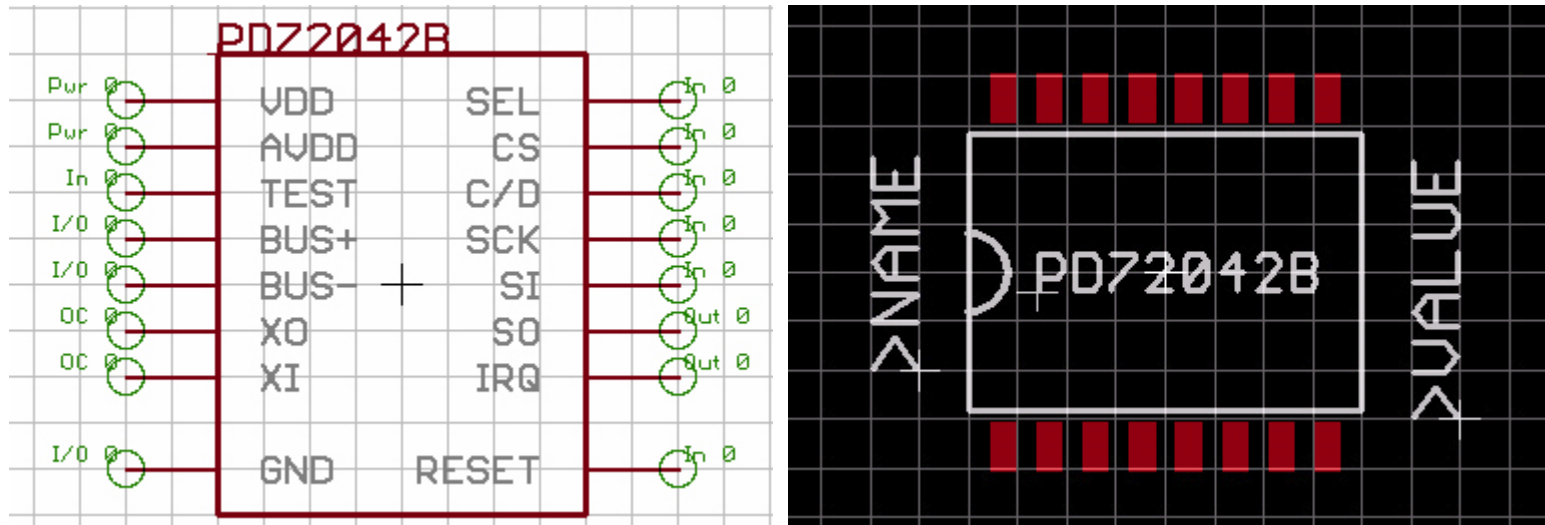


- Board view
- Schematic view

Popular elektronik-cad:  
EAGLE-CAD

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

# Schematic and Board view



Every electronic component has a schematic representation and a PCB representation (the component's foot footprint). The two views are linked.

*If you select a signal line in the diagram, then the corresponding "pin" in the printed circuit representation is also marked! Easy to orient the components.*

# Board view

**Electronics CAD is the main tool for an electro engineer when it comes to organizing the wiring of a circuit!**

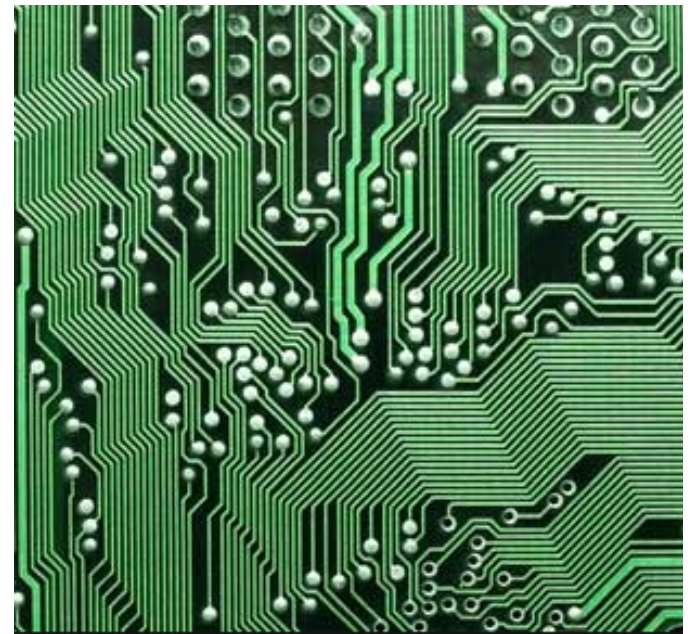
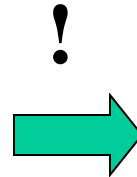
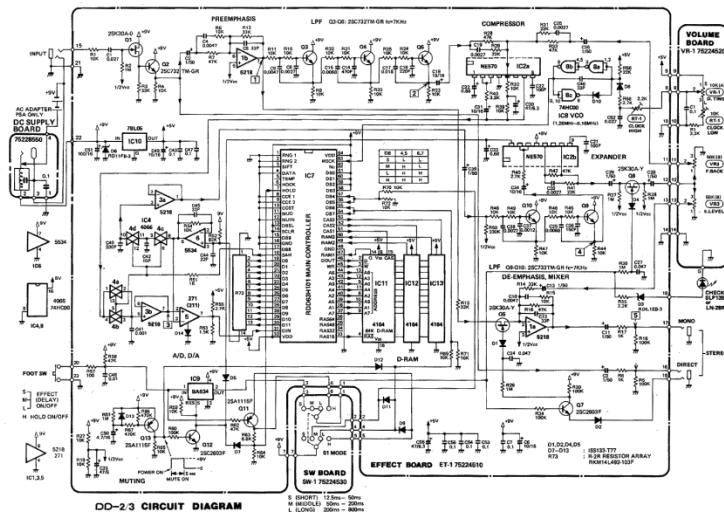
**But one need not produce a circuit board in order to take advantage of all the information on the component pin numbers, footprints, position and orientation, that one can get from a "board view"!**

CAD software is the natural place to gather information about your circuit and your components!

*It will often be confusion about component orientation: up, down, left, right, pin number, and so on - but everything can be described in the CAD file in a precise and unambiguous way!*

# A tried and tested circuit board is **evidence**

If you consistently used an Electronics CAD, then a tested circuit board can be seen as a proof that the circuit diagram presented in a report and the real circuit you have made your measurements on are the same!



- Who else dares to promise that all is connected right?

# You can buy PCP on internet

## PCB-service Mönsterkort

### Utförande

- Storlek upp till 24 dm<sup>2</sup> (425 × 575mm)
- Enkel-, dubbelsidigt eller 4-lager
- FR4 glasfiber 1.6 mm Cu 18/18 (IL 35/35)
- Min hålstorlek 0.30 mm
- Min ledarbredd 0.15 mm
- Min isolation 0.15 mm
- Min annular ring 0.15 mm
- Lödmask grön
- Texttryck vitt på en sida
- Varmförtening (blyfri)
- UL-märkning
- Fräs/scoring
- Panelisering
- Eltest

Leveranstid 5-7 arbetsdagar (ES,DS)

Underlagen skall vara i Gerber-format, helst 274X. Borrfil kan vara i annat format.

Efter genomförd betalning packa dina filer till en zip-fil och sänd den till [pcb@electrokit.se](mailto:pcb@electrokit.se). Ange ordernummer i mailet.

If you buy three cards will cost the same as when buying one (!)

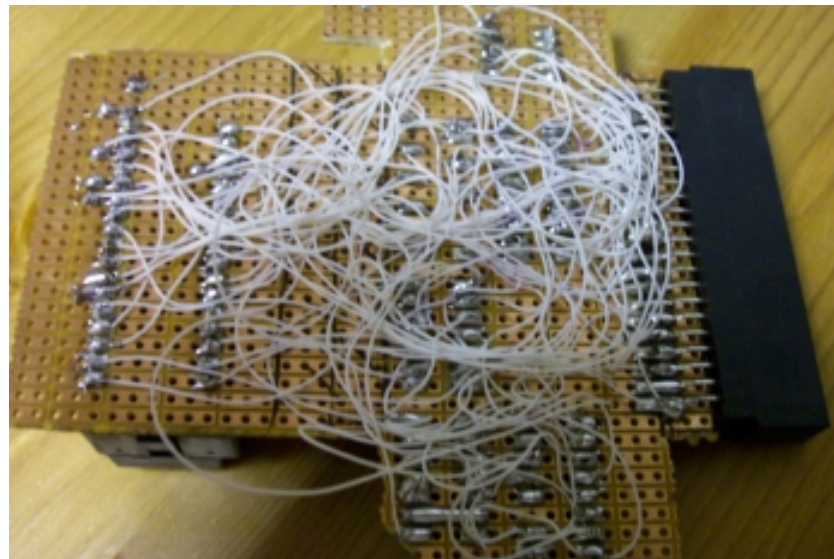
*Electronic-CAD  
can generate all  
files needed ...*



*Buy on  
internet!*

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

So now you know the secret of how an electronics engineer can keep track of all the wires?

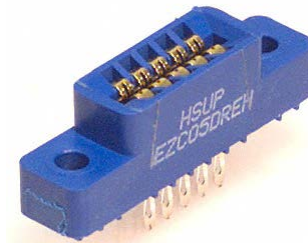
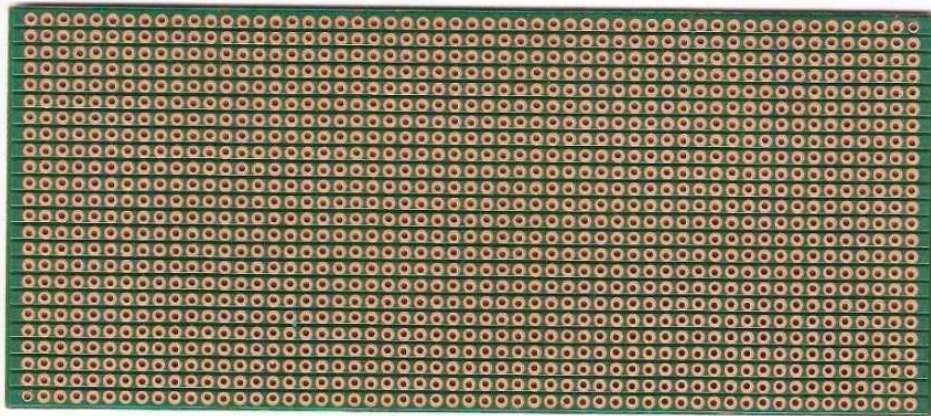




William Sandqvist [william@kth.se](mailto:william@kth.se)

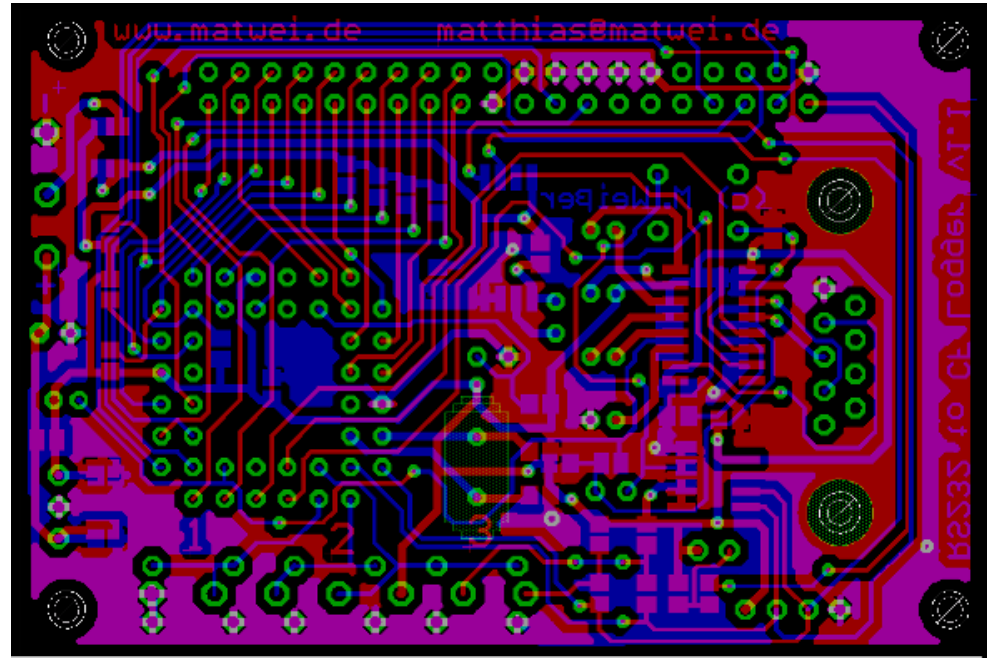
# What is a Mockup?

Mockups are non-functional PCB's that are used to show the physical format of a circuit board which is not yet available. A mockup is just some few components to try out where the contacts should be placed, and to show how large the most bulky components are. The components are soldered to a PCB with 0.1" pitch pattern of holes that only has solder pads.



# Virtual Mockup

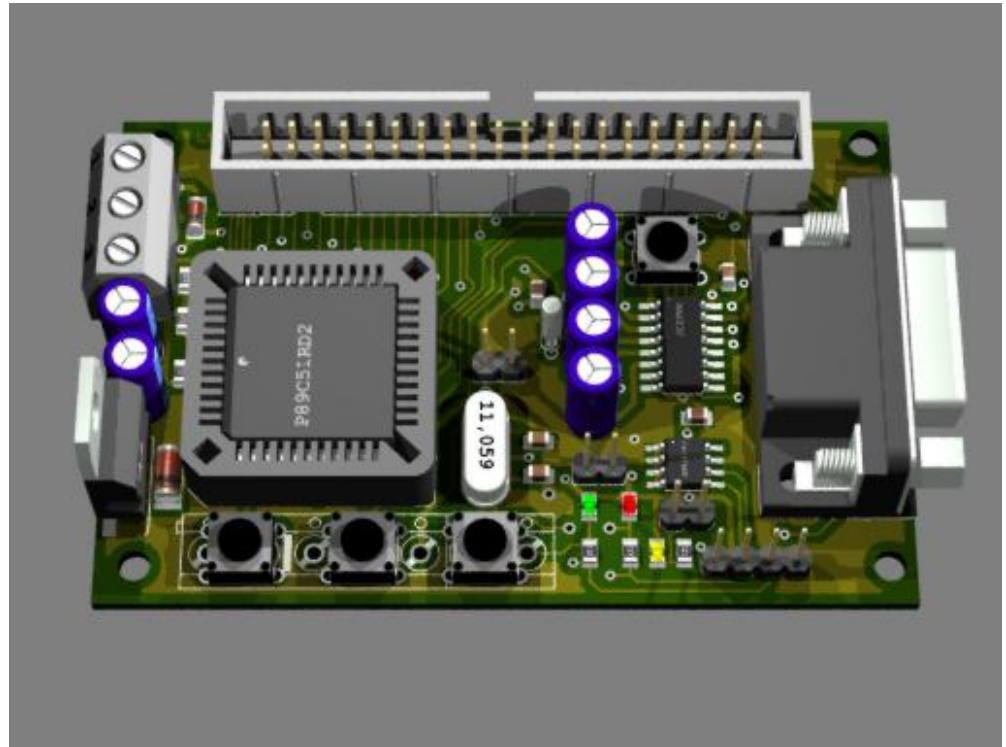
There are programs that can convert a CAD file into a 3D image! This can be used to show the size, appearance, component placement long before the actual product is exists!



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

# Virtual Mockup

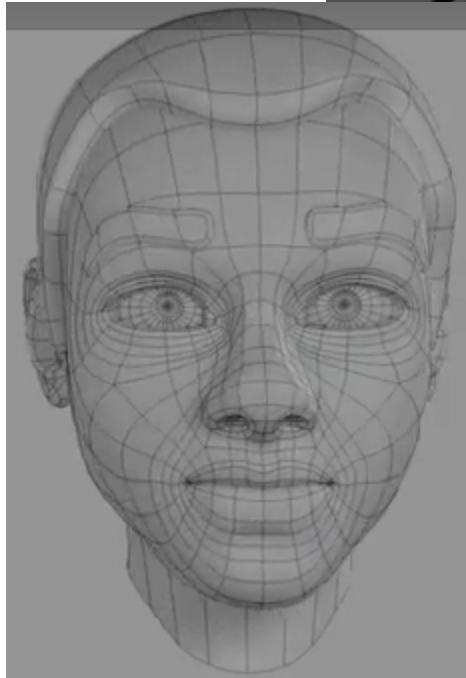
There are programs that can convert a CAD file into a 3D image! This can be used to show the size, appearance, component placement long before the actual product is exists!



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

# Mockup of a product

3D-CAD



A common Mockup material is OASIS foam



LEGO-mill

*A mockup showing how it will look like - but in a material that lacks strength.*

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

# Function Prototype + Mockup

A function prototype together with a mockup eliminates the need to directly produce a finished product.

This is often enough to convince others (technicians/funders) that you have sustainable idea.

William Sandqvist [william@kth.se](mailto:william@kth.se)