4. Product Data

4.2 Non-Geometrical Engineering Data

Requirements and Methods to Manage Data for Electronic Engineering, Automation, Production Planning etc.

Overview

- Electronic Engineering Data
- Automation Data
- Production Planning Data
- Simulation Data
- …

4.2.1 Electronic Engineering Data

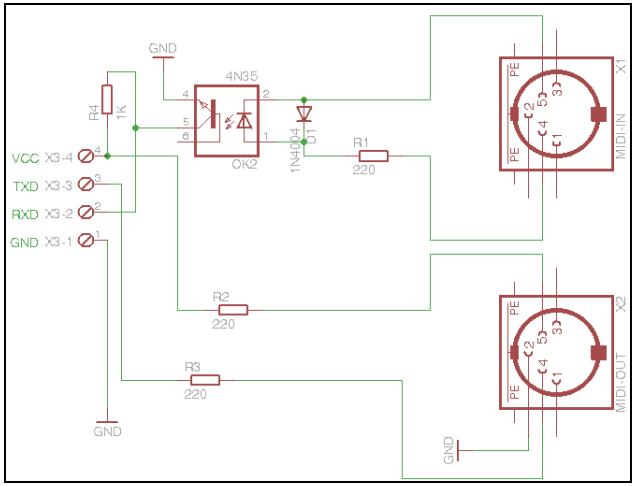
Thanks to Philipp Ludwig for support with slides!!!

Electrical and Electronic Engineering

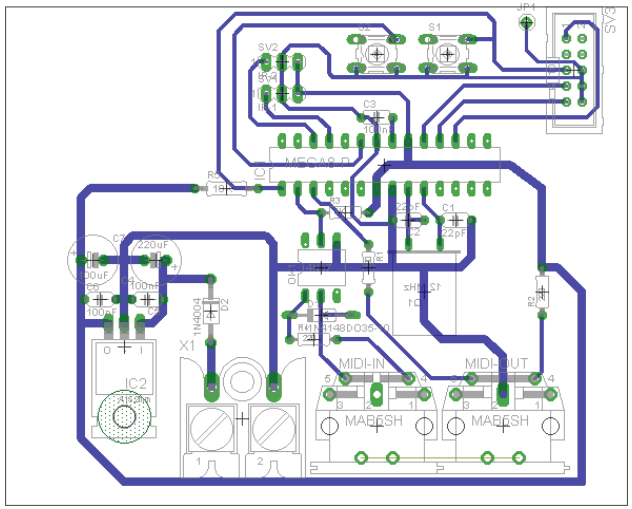
- **Electrical Engineering** is the general application of electricity, including high voltage, power plant technology, electric power distribution, etc.
- **Electronic engineering** is subfield of Electrical Engineering dealing with the development of (relatively small-scale) electronic circuits, i.e. those consisting of electrical components
- Here: focus on Electronic Engineering

Steps of Electronic Engineering

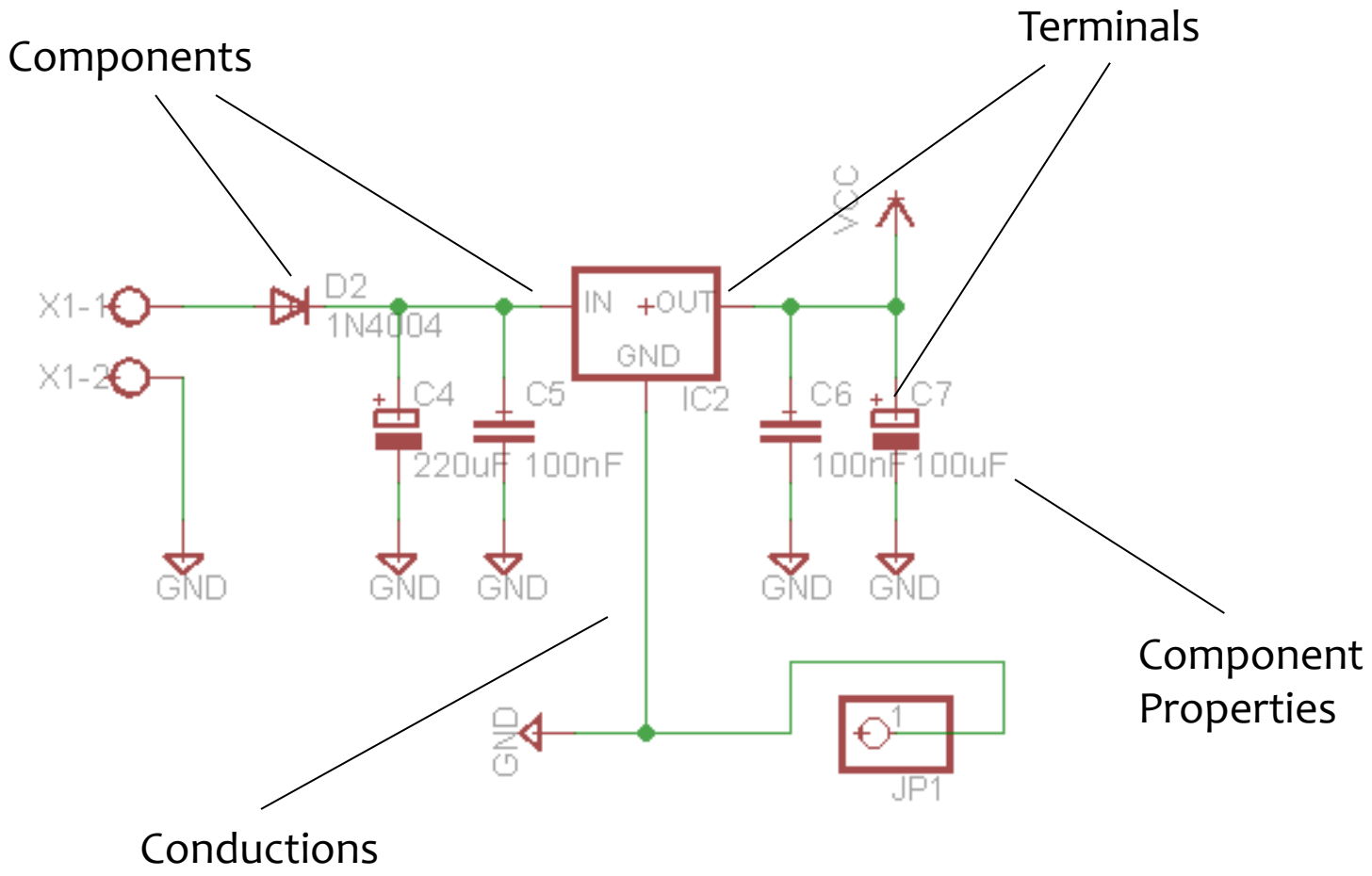
- **Schematic Design:** describes implementation of required functionality in terms of components and their connections



- **Physical Layout:** implements schema by assigning physical positions on to (semi-) conductor boards



Schematic Design/Capture



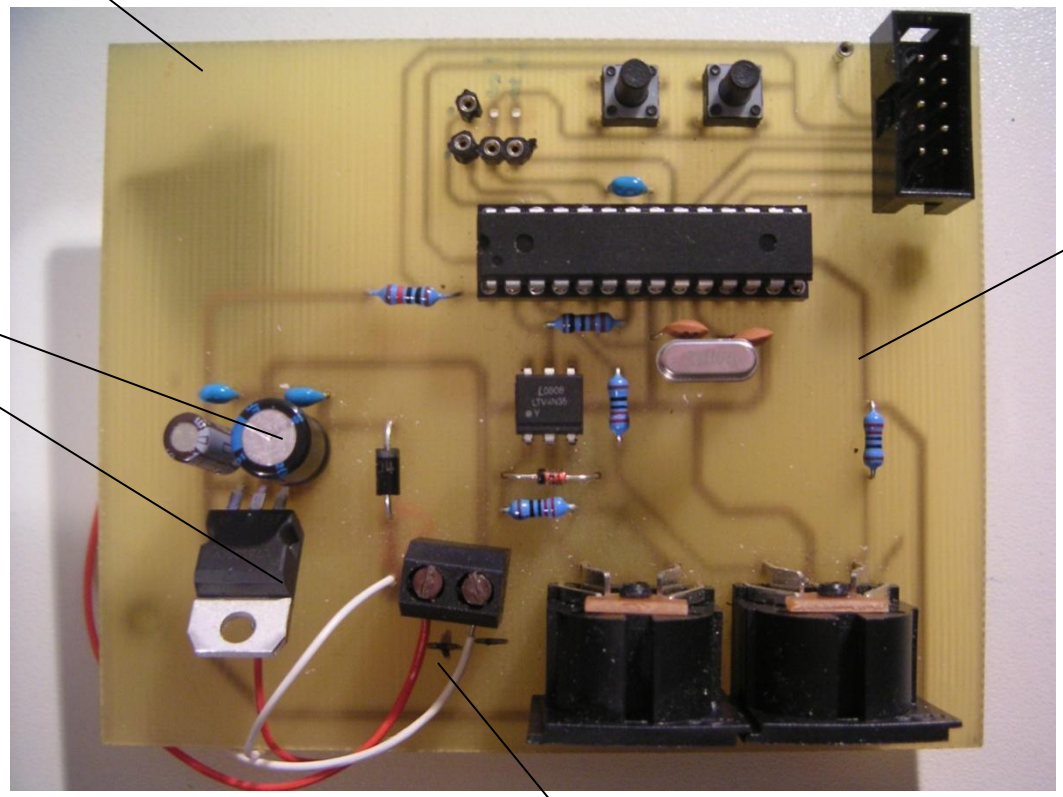
Requirements of Schematic Design /1

- Many types of **standard electronic components** with varying properties of different data types
  - Resistors, capacitors, inductors, power sources, diodes, transistors, integrated circuits, operational amplifiers, …
- **User-defined components** with application specific properties
- Formal description of functionality may be required for simulation
  - Function
  - Logical description

Requirements of Schematic Design /2

- Components have several connection points (**terminals**, leads, pins)
  - Often 2 (e.g. resistor, capacitor), but also 1 (e.g. antenna) or more than 2 (e.g. transistor)
  - Each terminal may have defined functionality
- Electronic circuit: conduction of terminals builds complex networks
- Correctness of the design can be verified partially (e.g. sufficient power sources, no short circuits

Board Design/Layout



Requirements of Board Layout /1

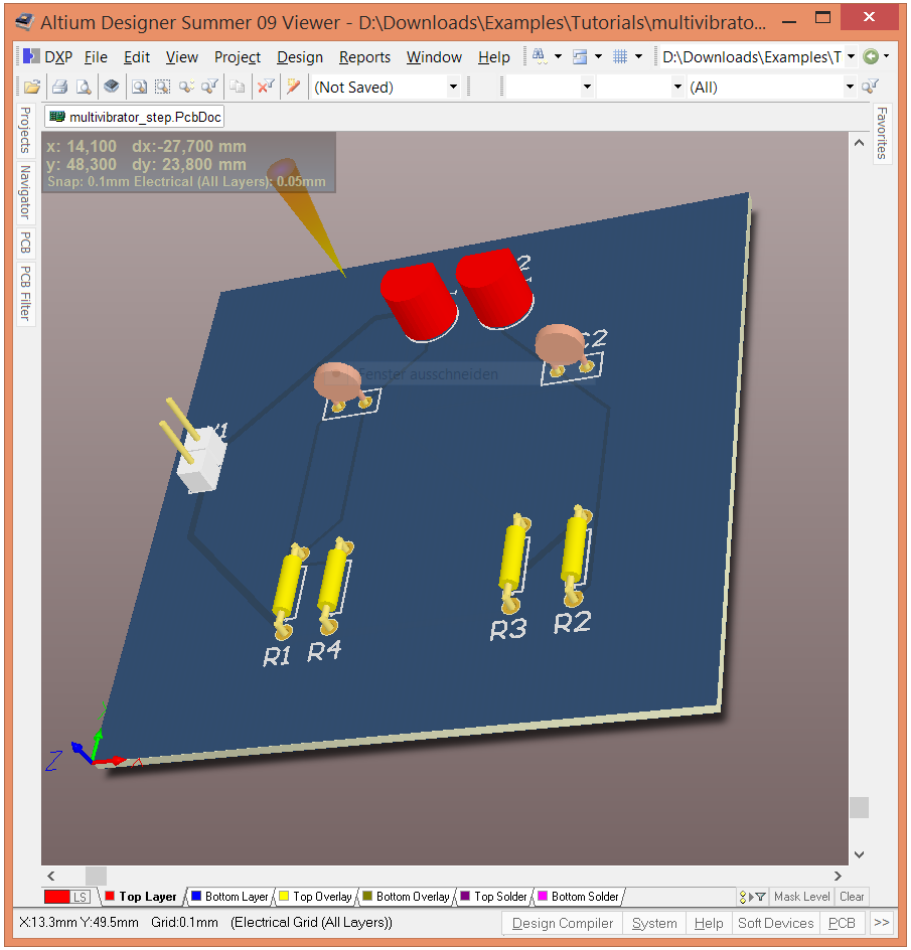
- Printed Circuit Boards (PCB) implement conductions of electronic circuits
  - Assign geometry to conduction paths
  - Paths: laminated copper sheets
  - Single sided, double sided, or multi-layered boards
  - Additional concerns: connection points (pads), holes, connection between layers, etc.
- Integrated Circuits (IC) as very dense electronic circuits with specific design and production techniques = Very-Large-Scale Integration (VLSI)

Requirements of Board Layout /2

- Phases of physical board layout
  - Placement of components
  - Routing of conduction paths
  - Optimization and validation
- Supported by tools, partly automated = Electronic Design Automation (EDA)
  - E.g. automated component placement based on specified constraints
  - Auto-routing of conduction paths
  - Most often requires manual adjustments or iterative tuning

Requirements of Board Layout /3

- Additional data structures
  - Size of components
  - Positions of components
  - Layout concerns: temperature, possible physical damage, etc.
- EDA supported by some CAD solutions
  - AutoCAD
  - SolidWorks Electrical
  - Altium Designer
  - …



Data Management for Electronic Data

- Several Proprietary file formats of EDA tools
  - Possibly separation of schematic and board design
  - 2D or 3D board design
- Presented here:
  - STEP AP 210: ISO-defined, trustworthy, complete solution
  - EAGLE: open, XML-based industry standard (human-readable, suitable for data exchange)
  - IDF: text-based format for exchanging information between 2D-PCB-Design and CAD
- EDA databases (see [2]) not (yet) covered in this lecture

Electronic data in STEP

- Several, partially overlapping solutions
  - **AP 210**: Electronic assembly, Interconnect and packaging design
  - **AP 212**: Electrotechnical design and installation (current analysis, equipment, lighting, cable sizing, electrical connectivity checks and cable tray interference detection)
  - **AP 220**: Process planning, manufacture, and assembly of layered electronic products.

STEP AP 210

- Data structures for exchanging electromechanical design requirements, functional specification, pin mapping, etc.
- With or without 2D or 3D model information
- The largest standard within STEP: About 900 individual concepts and 95 high level areas of functionality

Features of AP 210

- Data suitable for
  - Interconnect validation
  - PCB duration analysis and optimization
  - Component library
  - Simulation
  - 3D Assembly
  - Meta-Information about the product
  - …
- Complete list:
  http://www.wikistep.org/index.php/AP210_Use_Cases

STEP AP 210 file structure: Header

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (( 'STEP AP210' ),
    '1' );
FILE_NAME ('PCB-Example-AP210',
    '',
    ( '' ),
    ( '' ),
    'STEP 3.0',
    'TARGET 3001!',
    '' );
FILE_SCHEMA ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN' ));
ENDSEC;
```

STEP AP 210 file structure: Product

```
...

#5483=PRODUCT_CONTEXT('NONE',#5481,'mechanical');
#5484=PRODUCT('PCB-Example_Board','PCB-
Example_Board','',(#5483));
#5485=PRODUCT_DEFINITION_CONTEXT('',#5481,'design');
#5486=PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE('','NONE
',#5484,.NOT_KNOWN.);
#5487=PRODUCT_DEFINITION('NONE','NONE',#5486,#5485);
#5488=PRODUCT_DEFINITION_SHAPE('NONE','NONE',#5487);

...
```

STEP AP 210 file structure: Parts

```
...
#28=PRODUCT_DEFINITION_CONTEXT('part definition',#3,'');
...
#93=PRODUCT('CAP-1','CAP-1',$,(#20051));
#94=PRODUCT_DEFINITION_FORMATION('6',$,#93);
#95=PACKAGE('CAP-1',$,#94,#28,'',$,*);
...
#118=CLASSIFICATION_ROLE('conductive',$);
#119=CLASS('conductive','electrical conductivity');
#121=PRODUCT_DEFINITION_FORMATION('6',$,#910);
...
#910=PRODUCT('JAN1N757A','STANDARD',$,(#20051));
```

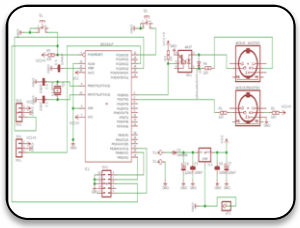
STEP AP 210 file structure: Placement

```
...
#95=PACKAGE('CAP-1',$,#94,#28,'',$,*);
...


#5714=PACKAGE_TERMINAL('5','part generic feature',#95,.T.);
#5715=USAGE_CONCEPT_USAGE_RELATIONSHIP('',#17731,#5716,'',$,#5714
,#559,*);
#5716=AXIS2_PLACEMENT_2D('',#5718,#5717);
#5717=DIRECTION('',(1.0,0.0));
#5718=CARTESIAN_POINT('',(0.4,0.0));
```

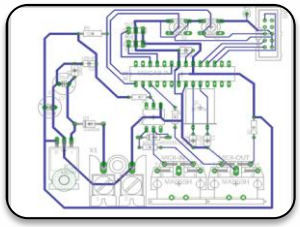
EAGLE file format

- EAGLE (Easily Applicable Graphical Layout Editor) commercially available EDA tool
- Propriatry, but open XML-based file format; defined by publically available *eagle.dtd*
- Every project is described by two XML files



project.sch

- Stores information about the electronic circuit



project.brd

- Stores information about the board layout

EAGLE file format: DTD

- Many definitions for all kinds of standard PCB elements
- Very general; allows not only the „usual“ elements but also custom objects, the position of screwholes for the case etc.

```
<!ELEMENT wire EMPTY>
<!ATTLIST wire
          x1 %Coord; #REQUIRED
          y1 %Coord; #REQUIRED
          x2 %Coord; #REQUIRED
          y2 %Coord; #REQUIRED
          width %Dimension; #REQUIRED
          layer %Layer; #REQUIRED
          extent %Extent; #IMPLIED
          style %WireStyle; "continuous"
          curve %WireCurve; "0"
          cap %WireCap; "round"
          >
```

EAGLE XML format – Basic structure

```
<eagle version="6.5.0">
    <drawing>

        ....

        <schematic>

        .....

        </schematic>
    </drawing>
</eagle>
```

EAGLE XML format – Schematic

```
<schematic>
  <parts>
    <!-- A list of all used parts. E. g.: -->
    <part name="SUPPLY1" deviceset="GND" device=""/>
    <part name="C1" deviceset="C-EU" device="050-024X044" value="22pF"/>
  </parts>
  <sheets>
    <sheet>
      <instances> <!-- Positions the parts on the board. E. g.: -->
        <instance part="SUPPLY1" gate="GND" x="132.08" y="187.96"/>
        <instance part="C1" x="-50.8" y="200.66" rot="R270"/>
      </instances>
      <nets>
        <-- Nets connect the parts with each other. A bunch of nets create the
        complete design. E. g. : -->
        <net name="N$1" class="0">
          &lt;segment&gt;
            <wire x1="9.44" y1="19.04" x2="8.9" y2="19.04" width="0.15"/>
            <wire x1="8.9" y1="19.04" x2="8.9" y2="20.66" width="0.15"/>
            <wire x1="8.9" y1="20.66" x2="2.4" y2="20.66" width="0.15"/>
            <pinref part="C1" pin="5"/>
            <pinref part="SUPPLY1" pin="1"/>
&lt;/segment&gt;</net></nets></sheet></sheets></schematic>
```

IDF: Intermediate Data Format

- CAD exchange format, especially created for importing/exporting PCB-Data
- Allows the exchange of information between the PCB creator and the case designer
- IDF describes only 2D-information, the CAD software builds the 3D-representation using additional libraries

IDF Example: Board assembly

```
.HEADER
BOARD_FILE 2.0 “Title" 2014.01.20.15:21:50 1
“example.sch" MM
.END_HEADER
...
.DRILLED_HOLES
0.812000 69.850000 31.750000 PTH C1
0.812000 69.850000 26.670000 PTH C1
0.812000 64.770000 31.750000 PTH C2
...
.END_DRILLED_HOLES
.PLACEMENT
"" "#" "C1"
69.850000 29.210000 0 TOP PLACED
"" "#" "C2"
64.770000 29.210000 0 TOP PLACED
"" "#" "C3"
62.230000 63.500000 0 TOP PLACED
...
.END_PLACEMENT
```

IDF Example: Component definition

```
.ELECTRICAL
C1      pn-cap MM 150.0
0 -55.0 55.0 0.0
0 -55.0 -55.0 0.0
0 135.0 -55.0 0.0
0 135.0 -80.0 0.0
0 565.0 -80.0 0.0
0 565.0 -55.0 0.0
0 755.0 -55.0 0.0
0 755.0 55.0 0.0
0 565.0 55.0 0.0
0 565.0 80.0 0.0
0 135.0 80.0 0.0
0 135.0 55.0 0.0
0 -55.0 55.0 0.0

PROP
CAPACITANCE 100.0
.END_ELECTRICAL
```

- Every component is described by a simple closed curve made up of arcs and lines
- Additional meta-data can be included using the „PROP“ section

Literature / Further Readings

[1] ISO 10303-210: *STEP AP 210 Electronic assembly, Interconnect and packaging design*

[2] Mark Bales: *Design Databases*. Chapter 12 in *Electronic Design Automation For Integrated Circuits Handbook (Volume 2)* by Martin, Scheffer, and Lavagno.