Department of Computer Science, NTNU

TDT4295 COMPUTER DESIGN PROJECT

PCB Layout

Asbjørn Djupdal

2022-09-09

From Idea to PCB

The road from...

- ...a rough idea of your system
- ...to a finished circuit board

Todays Lecture

- Workflow
- EFM32
- Artix FPGA
- Power Supply
- Final Tips

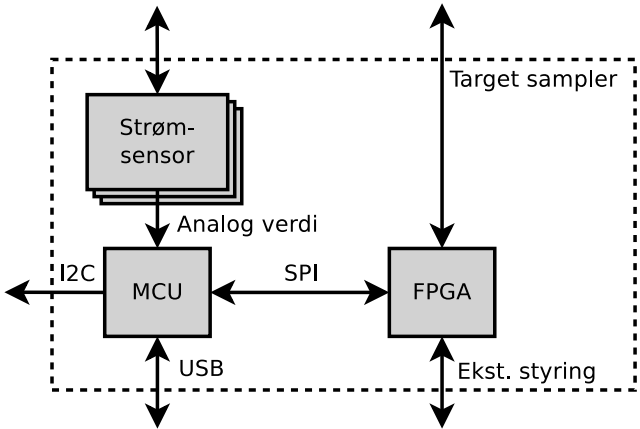
Typical Workflow

1. Decide on functionality
2. Decide on the main components
3. Draw schematics
4. Draw the PCB layout

Typical Workflow

1. ***Decide on functionality***
2. Decide on the main components
3. Draw schematics
4. Draw the PCB layout

Functionality

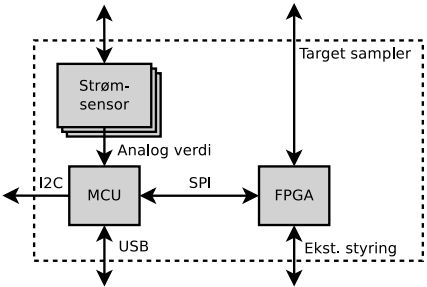


Typical Workflow

1. Decide on functionality
2. ***Decide on the main components***
3. Draw schematics
4. Draw the PCB layout

Choosing Components

- Identify the important ones first
  - Microcontroller
    - EFM32GG
    - I/O requirements?
    - Available: EFM32GG990F1024-BGA112
  - FPGA
    - Xilinx Artix 7
    - Available: XC7A100T
  - Other important components
- These components affect the detailed PCB design
  - Power supply
  - Support (smaller) components



Choosing Components

Choose components that are easy to use:

- Some components require complex support circuits and / or complicated PCB layout
- Some components can be difficult to drive from software
  - Are there drivers available you can use?
- Make sure that your components are compatible with each other
  - Voltage levels: Try to keep to 3.3V for all signals on the PCB
  - Communication standards (SPI/I2C/...)
- *Read and understand the datasheets*

Choosing Components

Choose components that are easy to solder:

- The same component is often available in different packages
- Easy to solder by hand:
  - QFP
  - SOIC / *SOP etc
  - SOT
- Require solder oven:
  - BGA
  - QFN
- Choose a large pitch (distance between pins)
  - Preferably 0.5mm or more (1mm for BGA)
- Resistors / capacitors etc:
  - Follow the datasheets
  - If not specified, choose 0805 or 1206 as standard
  - Avoid smaller than 0603

Choosing Components

Choose components you can get:

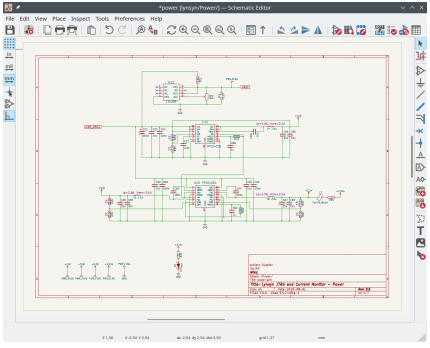
- Check availability on our main suppliers
  - no.farnell.com
  - digikey.no
- *You do not have time to wait for months for components*
- Send us your shopping lists as soon as possible
  - You don’t have to order everything at once
  - Order enough for at least 3 boards

Typical Workflow

1. Decide on functionality
2. Decide on the main components
3. ***Draw schematics***
4. Draw the PCB layout

Schematics

- KiCAD
- Drawing of your system
  - Specifies how components are connected
  - Does *not* show anything about physical placement, trace width etc.
- This is the most important part of the PCB design
- The schematic netlist can then be transferred to the PCB tool

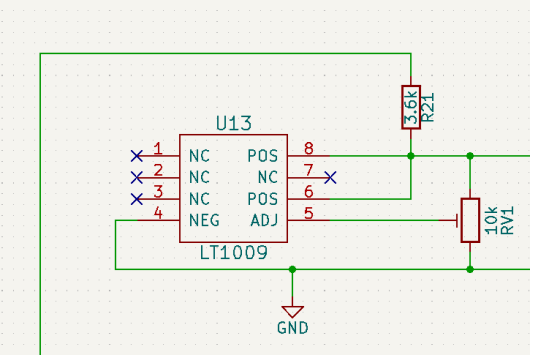


Schematics

- Follow the examples in the datasheets
  - Almost all components have an example that can be modified to your needs
- Use hierarchical boxes / schematics
  - Microcontroller
  - FPGA
  - Power supply
  - …
- Run ERC (Electrical Rules Checker) before transferring netlist to the PCB tool

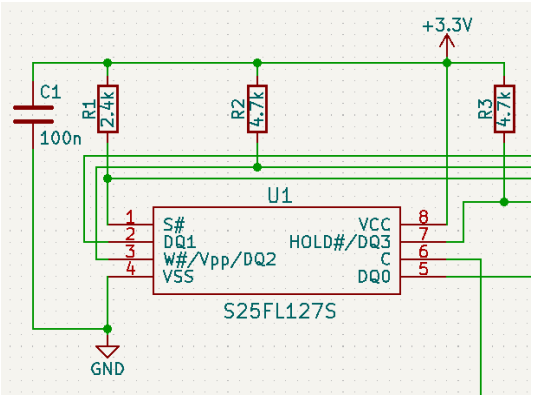
Symbols

- Symbols represent a PCB component in the schematic tool
- KiCAD has symbols for many components
- You probably have to make some
  - Make sure the pin numbering is correct



Decoupling Capacitors

- Most chips need one or more decoupling capacitors
- Keeps supply voltage (VCC) constant
- Placed close to the VCC pins (on the PCB, in the schematic it does not matter)
- Copy the datasheet!

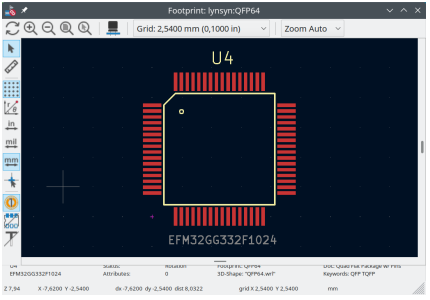


Design for Testing and Bugfixing

- Add test points for connecting measurement equipment
  - *But not on fast signals*
- Route some spare GPIO pins out on pin headers
  - Flexibility for expansion / backup solutions
  - Both from FPGA and the microcontroller
- Add redundancy
  - Want HDMI output? Add VGA also
  - Want SPI bus between FPGA and MCU? Add some extra wires
  - ...

Connect Schematic Symbol to Footprint

- Each symbol must be assigned a *footprint*
- The physical drawing of the component
  - Soldering pads
  - Mounting holes
  - Text printed on the PCB
- KiCAD has most standard footprints
- If you have to draw your own:
  - Be accurate
  - Do not mirror the pins
  - Follow the datasheets
  - Print your footprint on paper and check

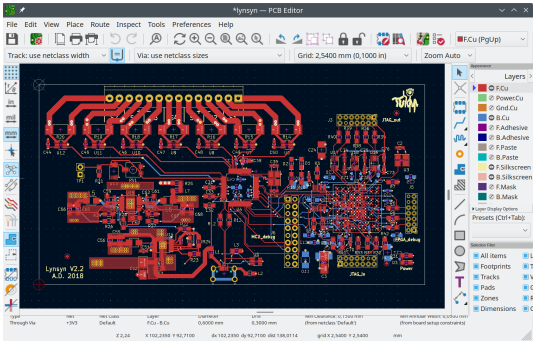


Typical Workflow

1. Decide on functionality
2. Decide on the main components
3. Draw schematics
4. ***Draw the PCB layout***

PCB Layout

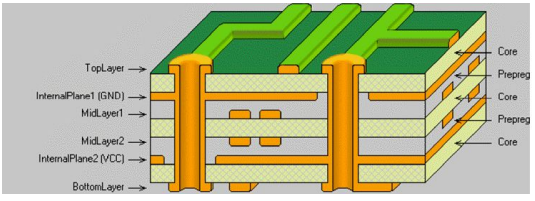
- KiCAD
- Physical layout of the PCB
- Place components
- Routing: Draw the metal traces between component pins
- Uses the netlist from the schematic to make sure everything is correctly connected



PCB: Setup – Number of Layers

A PCB has several layers with copper traces

- Use 4–8 layers
- One internal ground layer
- At least one internal VCC layer
- At least two signal layers (top and bottom)
- Number of layers:
  - Few layers: Cheap
  - Many layers: Easyer to route
  - FPGA is difficult to route with only 4 layers



(allaboutcircuits.com)

PCB: Setup – Design Rules

The rules you have to follow to make your PCB possible to produce

- KiCAD lets you specify the rules, and then enforces you to follow them
- Get information from the factory (`https://www.elprint.no/kapabilitet`) or use the following:
  - Clearance: 0,125 mm
  - Track width: 0,125 mm
  - Via dia: 0,6 mm
    - Via is a connection between one layer to another
  - Via drill: 0,3 mm
- Tighter design rules can also be fabricated, but remember that small is expensive.

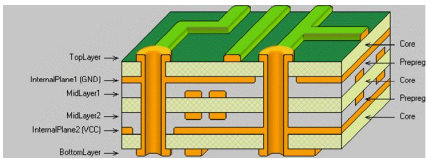
PCB: Placement of Components

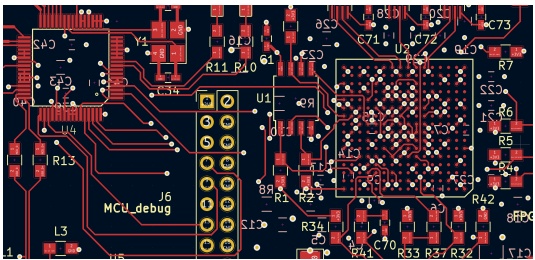
You are now ready to start laying out the PCB

- Place all components manually
- Then draw the edges of your board
  - Don't make the board too small
- Think about placement and rotation
  - Important for how easy routing is later
  - Do not place components too close to other components
    - Difficult routing and difficult soldering
- Group your components according to your schematics
- Follow any recommendations from the dataseheets

PCB: Routing

- I recommend manual routing
  - Autorouter does not produce a good result
- Don't use mimimum trace width, unless necessary
  - Use one standard width for signal traces, another thicker width for power traces
- *Don't use "buried" or "blind" vias*. Only normal through-hole vias
- *Don't put a via in a solder pad*





PCB: Signal Integrity

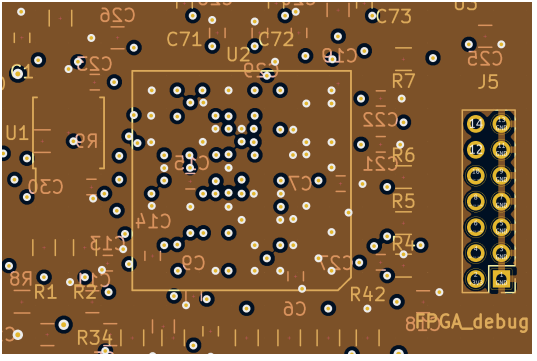
- Wire traces on the PCB are not perfect
  - Act as transmission lines at high frequencies
  - Has an *impedance*
  - May *interfere* with close-by traces
- Some designs need a specific impedance and/or length matching
  - DDR memory
  - RF circuits
  - Advice: Avoid if you can
    - If not, read up on it *and* discuss with me
- Differential pairs (e.g. USB) should be routed correctly
  - Use KiCAD differential pair router
- Keep high speed lines short
  - Place crystals and oscillators close to the chips they feed
- Don’t route sensitive signals through connectors or via pin headers

PCB: Analog and Digital

- For minimum amount of noise: Separate analog part of PCB from digital part
- Put the analog part towards an edge of the PCB
  - Not between power supply and digital part
- Normal to have separate VCC for the analog circuits (see EFM32 AN0002 for an example on isolating the analog VCC)
- Big and complex topic
  - Don't speed too much time thinking about it
  - Audio will work anyways (possibly with worse sound quality), even if you ignore all analog advices

PCB: Finalization

- Fill all planes
  - Gnd and VCC layers are “planes”
  - *Check that your vias have not damaged any of your power planes*
- Run DRC (Design Rules Checker)
- Print to paper and check that:
  - Components fit physically
  - There is enough space for the soldering iron



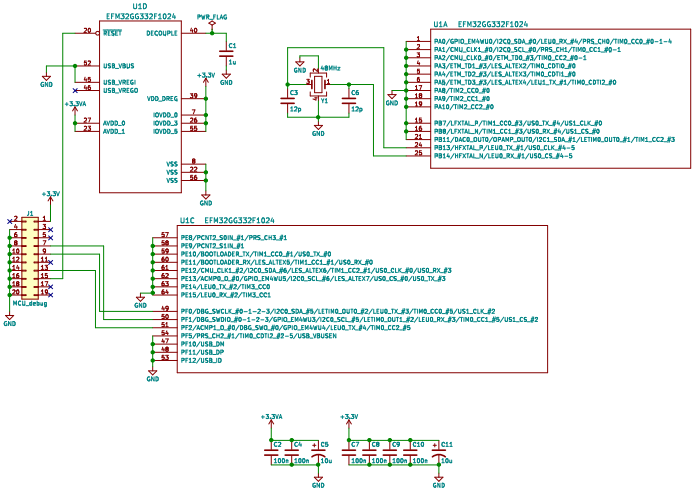
EFM32GG

- Easy to use
- Power supply: 3.3V
- Clock: Can use an internal oscillator, or external crystal
  - Recommend 48MHz external crystal
- Exist in many variants (parts): Choose the part with the properties you need
  - I/O
  - Flash
- Remember to include the debug connector
- Relevant documents for PCB:
  - SiLabs EFM32 AN0002 (Hardware Design Considerations)
- Prefer package QFP64

EFM32GG – Availability

- Many parts are difficult to get these days
- Available in our lab: EFM32GG990F1024-BGA112
  - ARM Cortex M3
  - Flash: 1024kB
  - RAM: 128kB
  - USB support
  - ADC: 8 channels, 12 bit
  - DAC: 2 channels, 12 bit
  - SPI: 3
  - I2C: 2
  - GPIO: 87
  - *112 pin BGA package*
  - ...

EFM32GG


U1D
EFM32GG332F1024
PWR_FLAG
RESET
DECOUPLE
C1
1u
USB_VBUS
USB_VREGI
USB_VREGO
VDD_DREG
AVDD_0
AVDD_1
IOVDD_0
IOVDD_3
IOVDD_5
VSS
GND
+3.3V
+3.3VA
U1A EFM32GG332F1024
PA0/GPIO_EM4WU0/I2C0_SDA_#0/LEU0_RX_#4/PRS_CH0/TIM0_CC0_#0-1-4
PA1/CMU_CLK1_#0/I2C0_SCL_#0/PRS_CH1/TIM0_CC1_#0-1
PA2/CMU_CLK0_#0/ETM_TD0_#3/TIM0_CC2_#0-1
PA3/ETM_TD1_#3/LES_ALTEX2/TIM0_CDTI0_#0
PA4/ETM_TD2_#3/LES_ALTEX3/TIM0_CDTI1_#0
PA5/ETM_TD3_#3/LES_ALTEX4/LEU1_TX_#1/TIM0_CDTI2_#0
PA8/TIM2_CC0_#0
PA9/TIM2_CC1_#0
PA10/TIM2_CC2_#0
PB7/LFXTAL_P/TIM1_CC0_#3/USO_TX_#4/US1_CLK_#0
PB8/LFXTAL_N/TIM1_CC1_#3/USO_RX_#4/US1_CS_#0
PB11/DAC0_OUT0/OPAMP_OUT0/I2C1_SDA_#1/LETIM0_OUT0_#1/TIM1_CC2_#3
PB13/HFXTAL_P/LES_TX_#5/USO_CLK_#4-5
PB14/HFXTAL_N/LEU0_RX_#1/USO_CS_#4-5
48MHz
C5
12p
C6
12p
Y1
J1
MCU_debug
U1C EFM32GG332F1024
PE8/PCNT2_S0IN_#1/PRS_CH3_#1
PE9/PCNT2_S1IN_#1
PE10/BOOTLOADER_TX/TIM1_CC0_#1/USO_TX_#0
PE11/BOOTLOADER_RX/LES_ALTEX6/TIM1_CC1_#1/USO_RX_#0
PE12/CMU_CLK1_#2/I2C0_SDA_#6/LES_ALTEX6/TIM1_CC2_#1/USO_CLK_#0/USO_RX_#3
PE13/ACMP0_O_#0/GPIO_EM4WU5/I2C0_SCL_#6/LES_ALTEX7/USO_CS_#0/USO_TX_#3
PE14/LEU0_TX_#2/TIM3_CC0
PE15/LEU0_RX_#2/TIM3_CC1
PF0/DBG_SWCLK_#0-1-2-3/I2C0_SDA_#5/LETIM0_OUT0_#2/LEU0_TX_#3/TIM0_CC0_#5/US1_CLK_#2
PF1/DBG_SWDIO_#0-1-2-3/GPIO_EM4WU3/I2C0_SCL_#5/LETIM0_OUT1_#2/LEU0_RX_#3/TIM0_CC1_#5/US1_CS_#2
PF2/ACMP1_O_#0/DBG_SWO_#0/GPIO_EM4WU4/LEU0_TX_#4/TIM0_CC2_#5
PF5/PRS_CH3_#1/TIM0_CDTI2_#2-5/USB_VBUSEN
PF10/USB_DM
PF11/USB_DP
PF12/USB_ID
C2
C4
C5
100n
100n
10u
C7
C8
C9
C10
C11
100n
100n
100n
100n
10u

Xilinx Artix 7 FPGA

- Complex chip
- Needs a lot of support circuits on the PCB
  - Complex power supply
  - Lots of decoupling capacitors
  - Need a flash chip for the configuration
  - JTAG connector
- Preferably choose the FTG256 package
  - Available in the lab: Artix 100T (FTG256)

Artix 7 – Relevant Documents

- UG483: 7 Series FPGAs PCB Design Guide
  - Information about decoupling capacitors
- XAPP586: Using SPI Flash with 7 Series FPGAs
  - How to connect the Flash chip and JTAG connector
- UG908: Vivado Design Suite User Guide
  - Appendix C has a list of supported Flash chips
- Other documents:
  - UG470: 7 Series FPGAs Configuration
    - More about JTAG and configuration
  - XAPP427: Implementation and Solder Reflow Guidelines for Pb-Free Packages
    - Soldering of Xilinx BGA chips

Artix 7 – Decoupling Capacitors

- Needs lots of decoupling capacitors
- Sizes, number and max distance to chip is specified in UG483
  - One possibility is to place these on the bottom of the PCB

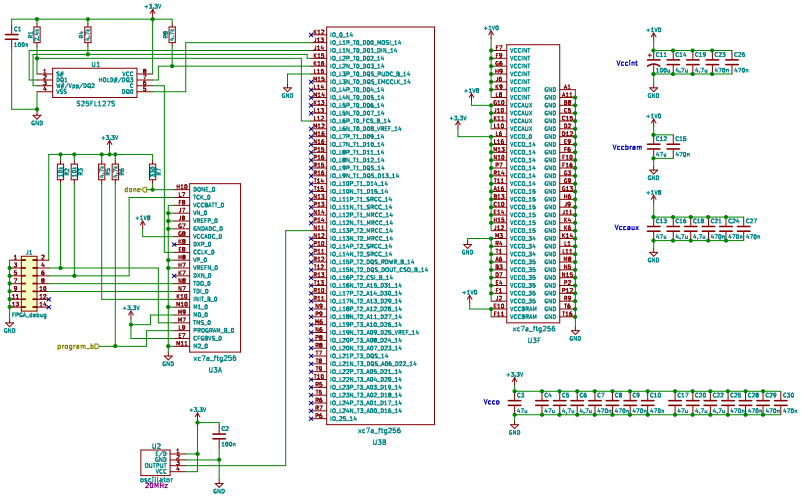
Artix 7 – Flash

- FPGA needs a separate Flash chip for its configuration
- Choose one which is compatible:
  - Choose from appendix C in UG908
- Copy schematic from XAPP586

Artix 7 – JTAG and Reset

- Connect as in XAPP586
- Add 10k pullup resistor on TMS and TCK to avoid problems when JTAG cable is disconnected
  - A pullup resistor is a resistor connected to VCC. Makes sure the line is high if nothing else drives it low

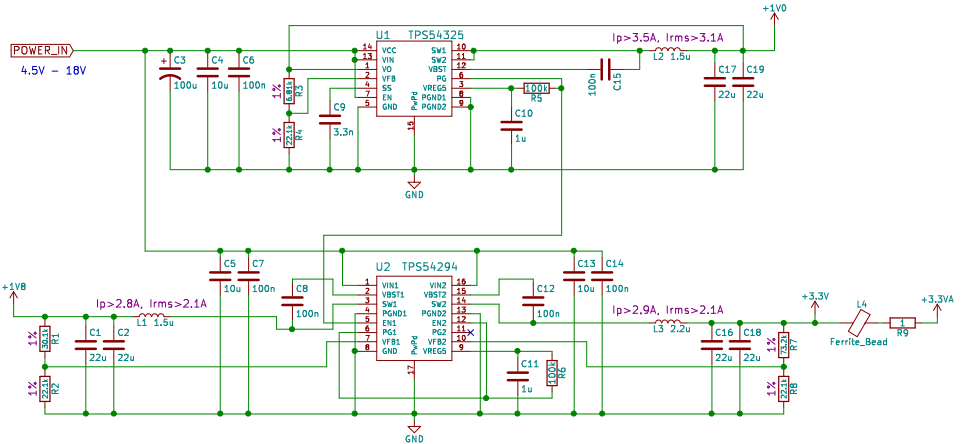
Artix 7


+3.3V
C1
100n
R1
4k7
R2
4k7
R3
4k7
U1
HOLD#/DQ3
C
DQ0
S25FL127S
GND
+3.3V
R4
4k7
R5
4k7
R6
4k7
R7
4k7
R8
4k7
done
+1V8
J1
FPGA_config
program_b
U3A
xc7a_ftg256
+3.3V
C2
100n
U2
OUTPUT
VCC
oscillator
20MHz
U38
xc7a_ftg256
+1V0
+1V8
+3.3V
U3F
xc7a_ftg256
VccInt
Vccbram
Vccaux
Vcco
C3
C4
C5
C6
C7
C8
C9
C10
C17
C30
C22
C28
C29
C30

Power Supply

- MCU needs 1 voltage: 3.3V
- FPGA needs 3 voltages: 1.0V, 1.8V, 3.3V
  - Must be switched on in the correct sequence
- Switching converter (buck converter) or linear?
  - Switching is more efficient
  - Linear is easier to use

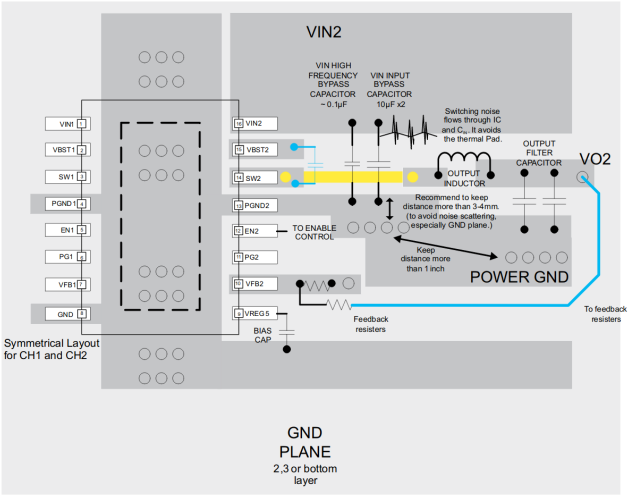
Power Supply Schematics



Power Supply Advice

- Power supply is complex, and critical
- Incorrect voltages damages the rest of your components
- Have a backup plan
  - Use jumpers to connect the voltages to the rest of the board
  - Allows you to disconnect a faulty power supply and add a lab supply

TPS54294 Layout



Final Tips

- Start with schematics for the power supply, FPGA and MCU
  - Adapt schematics from this lecture to your needs
- Then add other chips and components to your schematics
  - Read the datasheet for info about decoupling and usage
  - Add the support components as you go
- When using git: *Don't merge your KiCAD files*
  - One person per schematic file
- Check these slides regularly

Delivery

Send me the complete design by the end of the month

- Complete functionality
- Passes all tests