MRSD Project Course I:
Printed Circuit Board Design and Fabrication

September 19, 2017

Acknowledgement

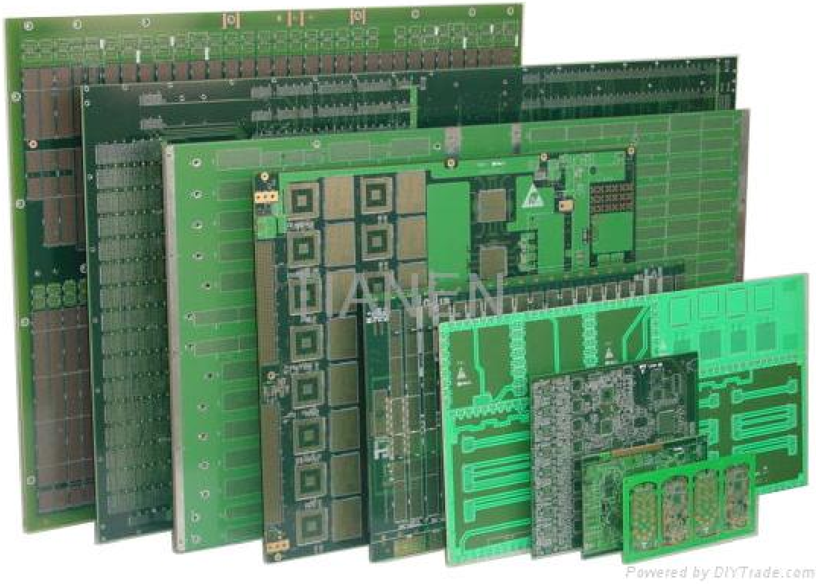
- Large portion of the material presented here was produced by Tom Lauwers

Questions to Answer

- **What is a Printed Circuit Board (PCB) and why are they instrumental to electronic devices?**
- **How do I design a PCB?**
  - What's an electrical schematic good for?
  - What's a "Layout"?
  - Where do parts come from?
- **How do I convert a system diagram into an electrical schematic?**
- **What are some important design considerations?**

The Printed Circuit Board (PCB)

- **Mechanical Mounting**
- **Electrical Connectivity**
  - **Traces can form components**
- **Many flavors**

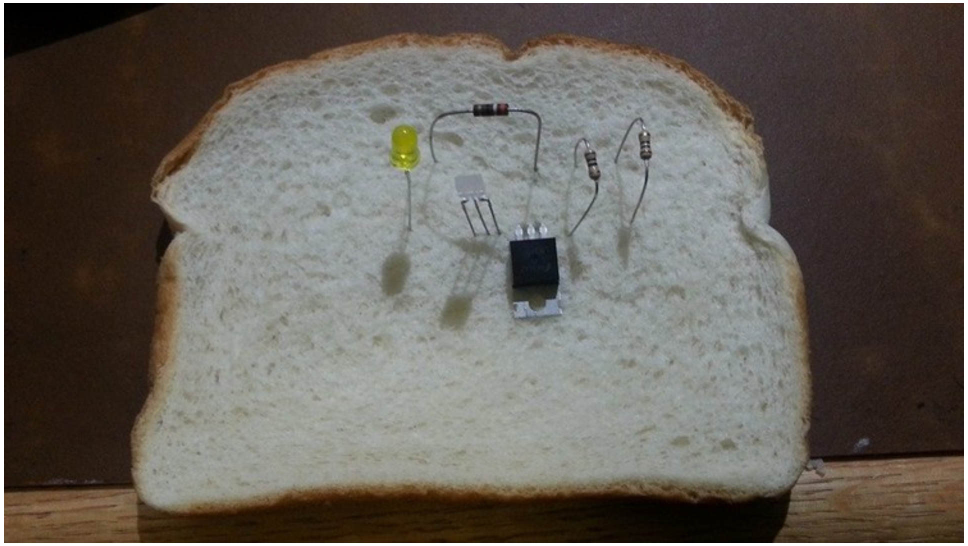


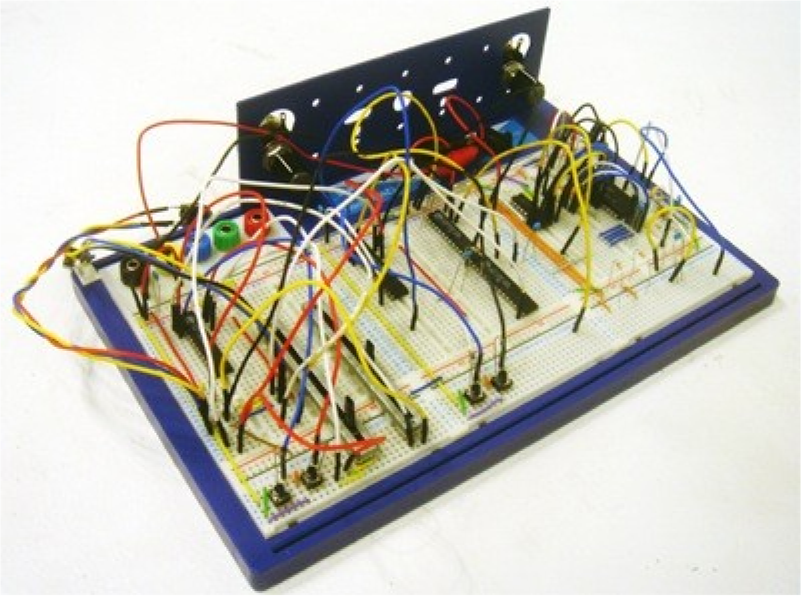
Who is this man?

- Paul Eisler (1907-1995)
- Invented the PCB in 1936
- Used it in his radio
- Also invented rear-window defroster

Which is Better?

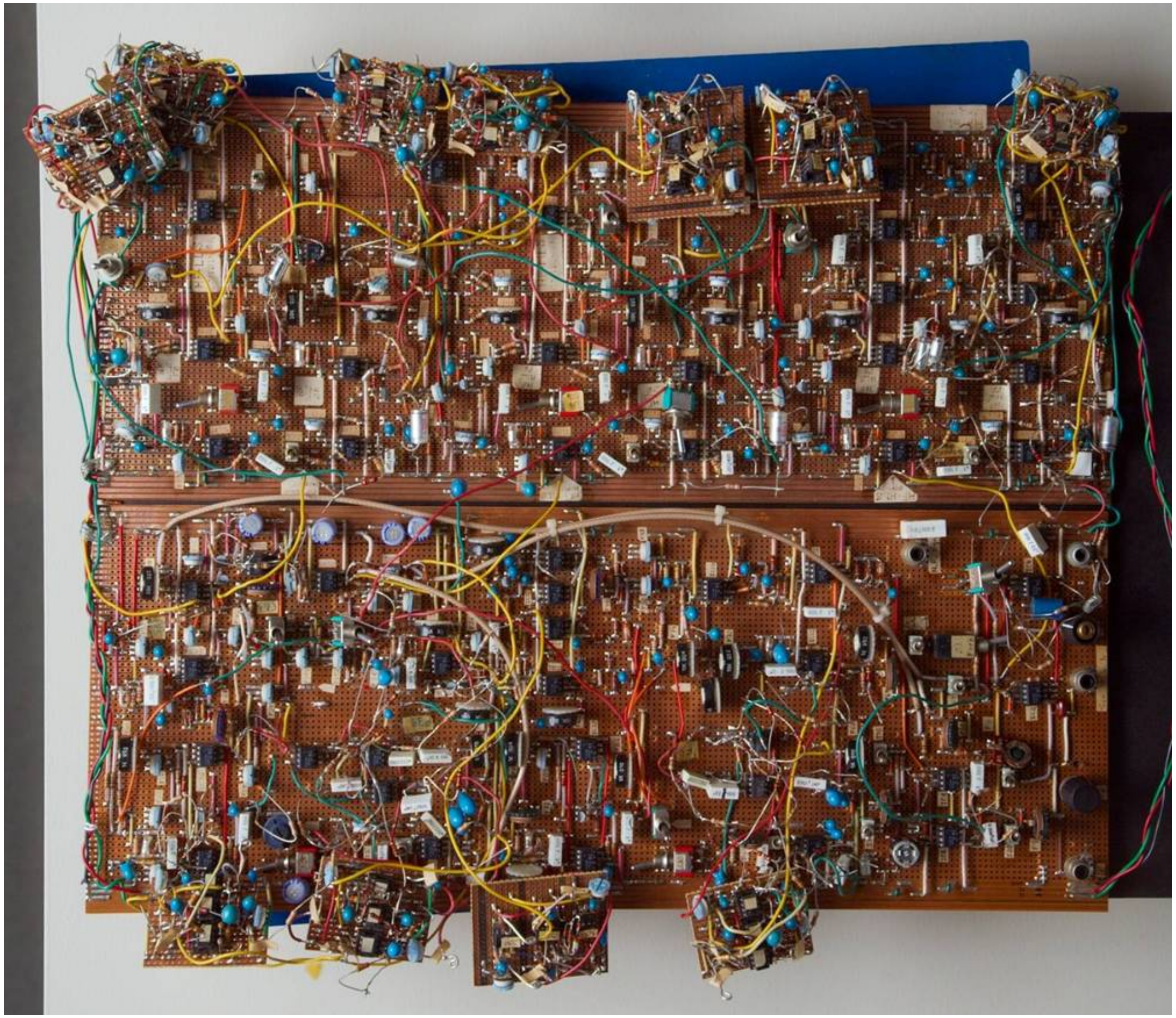
This???



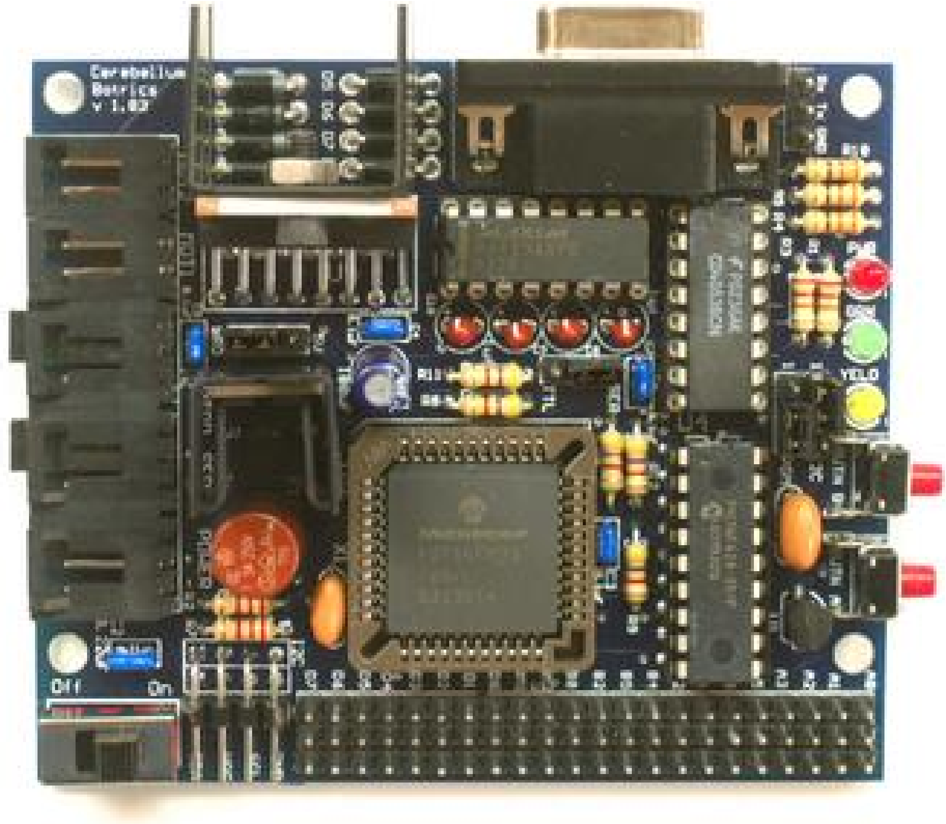


Which is Better?

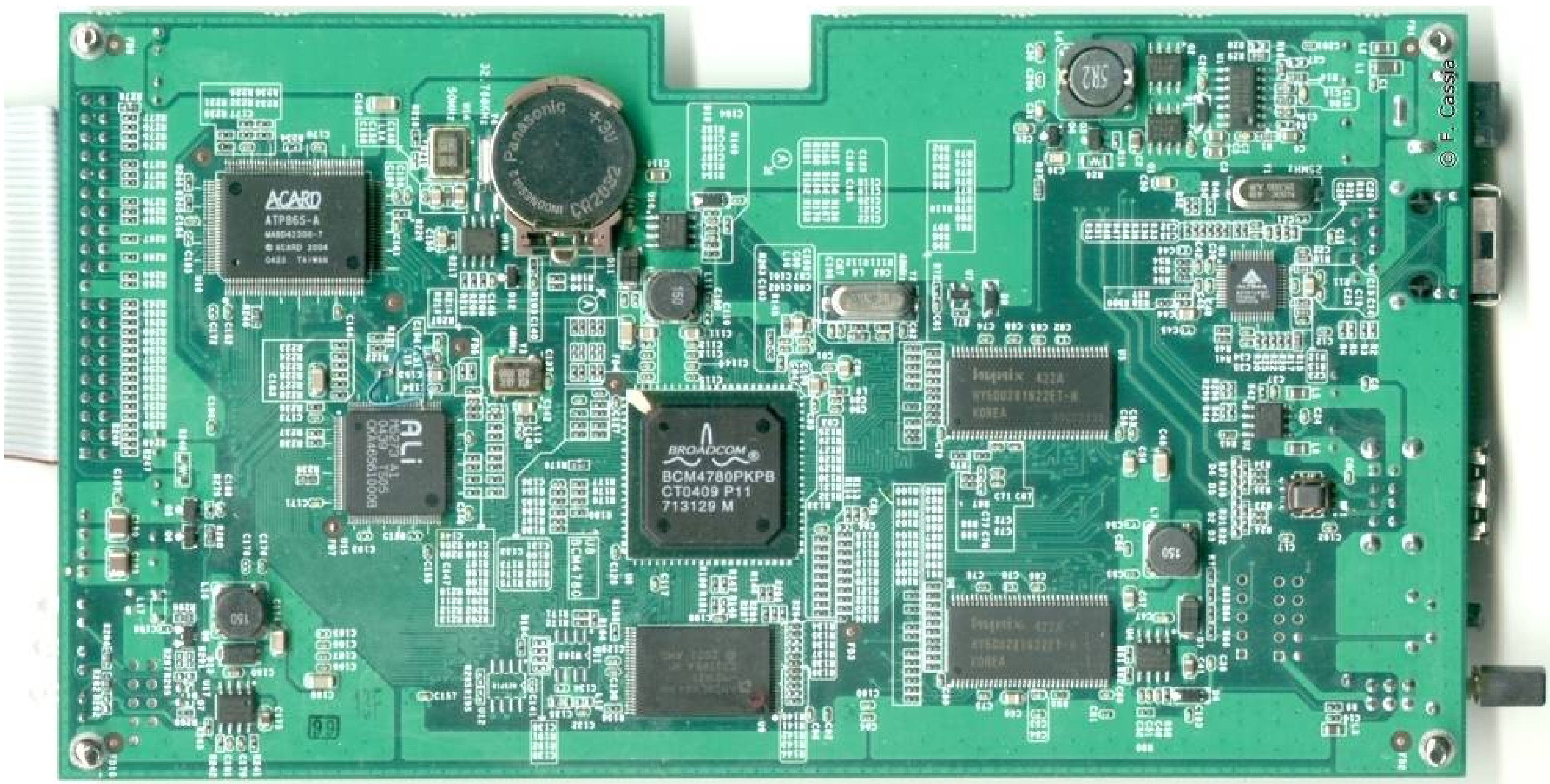
This???!!!?!!



This is Better.



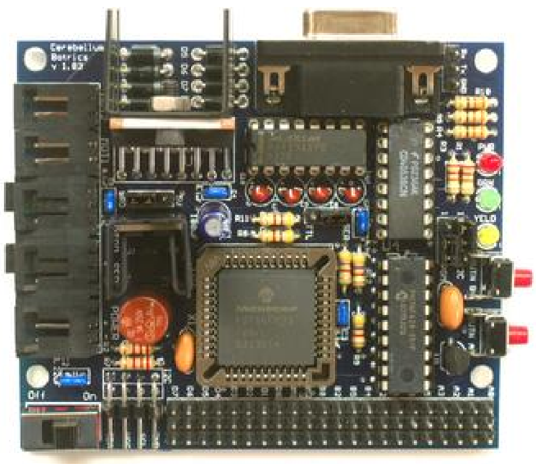
This is Even Better*.



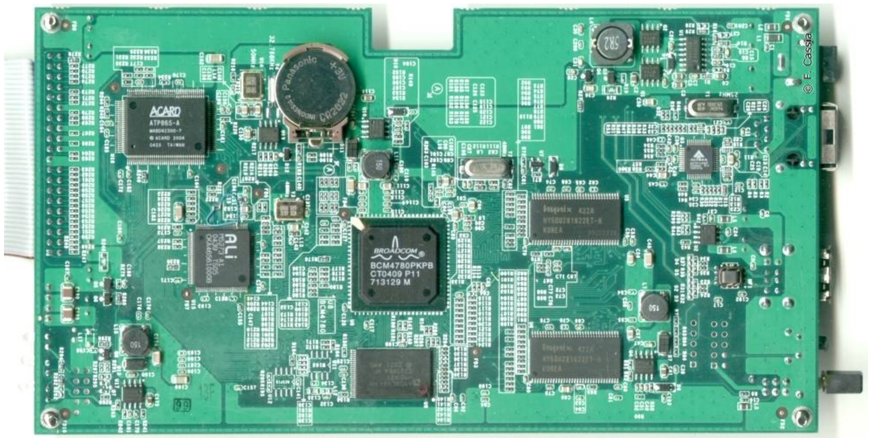
* - see next slide

Through-Hole Vs. Surface Mount

Through-Hole:
- Easy to solder
- Easier to fix circuits
- You don't lose components on the floor



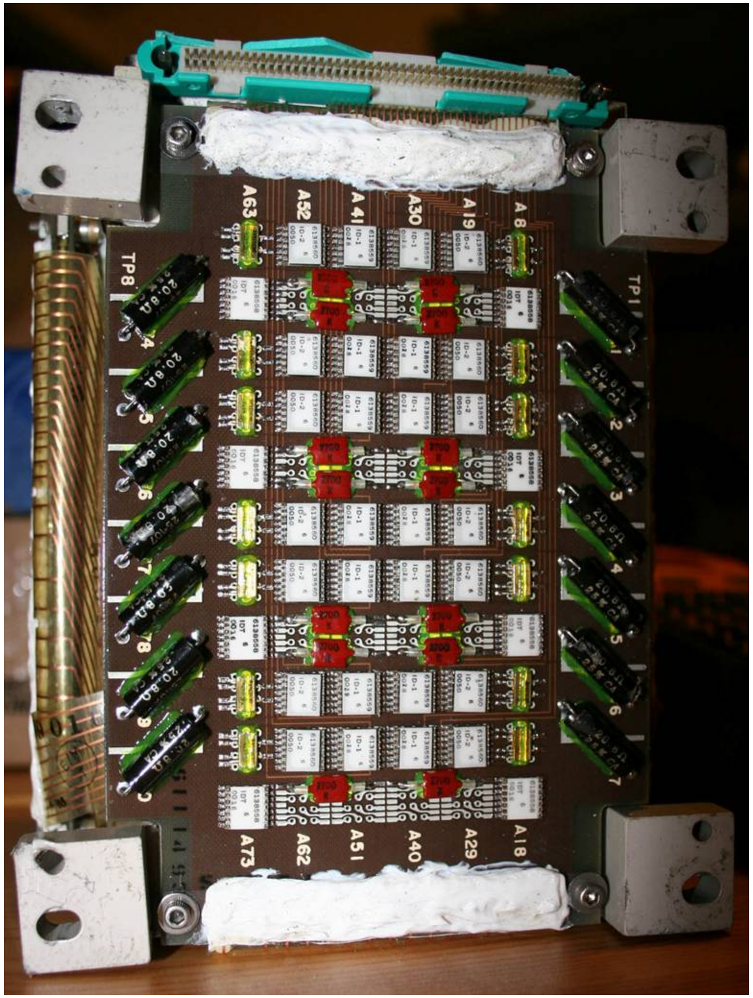
Surface Mount:
- Miniaturization
- From *small* to *ridiculous*
- Easier Automated Assembly
- PCB Fabrication is easier – fewer holes to drill
- Most cool parts are only Surface mount



Historical Digression

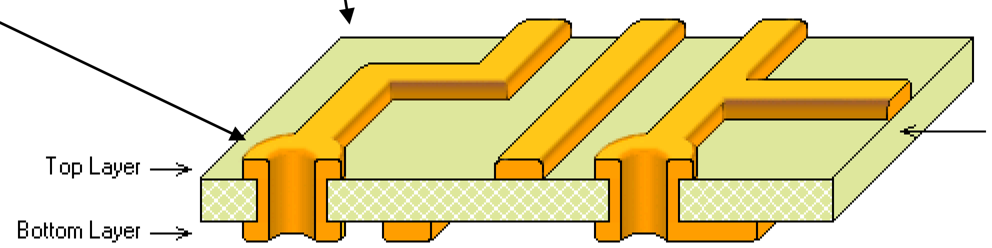
- **First surface mount boards were developed for Saturn V flight computers**
  - **In 1964!**
  - **By IBM**

- **This module can hold 14,336 bytes**
  - **The Saturn V had 8 of them**



Multi-layer Construction

- **PCBs have many layers:**
  - **The dielectric**
  - **Copper layers**
  - **Vias**



- **Other Common Features**
  - **Soldermask**: [usually green] overlay that insulates copper
  - **Silkscreen**: [usually white] with text, logos, etc.
  - **Multi-layer**: Many sandwiched copper/dielectric layers "planes" are often implemented this way. Complex PCBs like motherboards might have 16 copper layers—or even more

PCB Materials

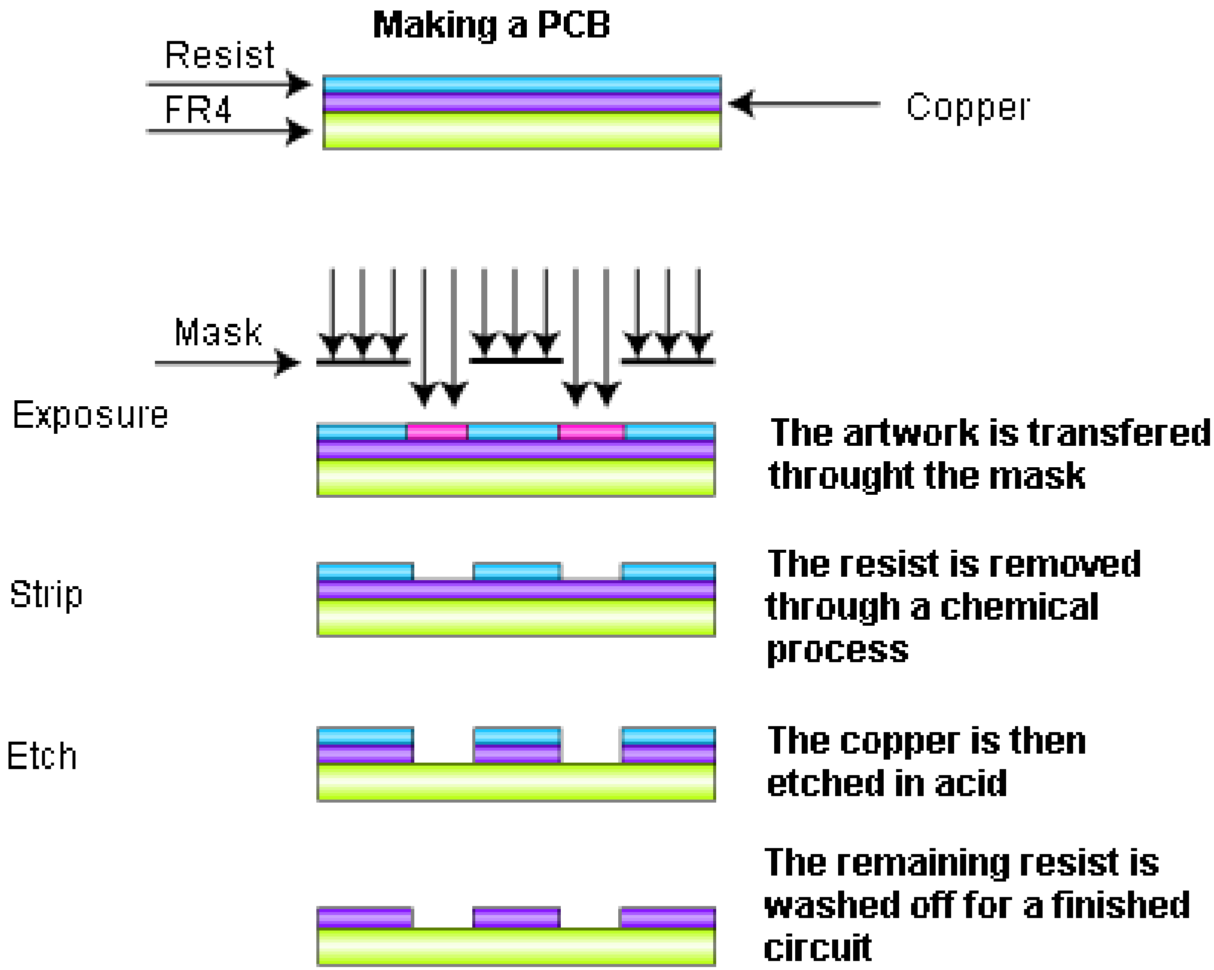
- **PCBs are available in several grades**

GRADE DESIGNATION	MATERIAL/COMMENTS
FR–1	Paper/phenolic: room temperature punchable, poor moisture resistance.
FR–2	Paper/phenolic: suitable for single-sided PCB consumer equipment, good moisture resistance.
FR–3	Paper/epoxy: designed for balance of good mechanical and electrical characteristics.
FR–4	Glass cloth/epoxy: excellent mechanical and electrical properties.
FR–5	Glass cloth/epoxy: high strength at elevated temperatures, self-extinguishing.
G10	Glass cloth/epoxy: high insulation resistance, highest bond strength of glass laminates, high humidity resistance.
G11	Glass cloth/epoxy: high flexural strength retention at high temperature, extreme resistance to solvents.

Modern PCB Fabrication Steps

- **Automated Fabrication:**
  1. Clad dielectric with copper foil (~0.0015" thick)
  2. Create copper traces using photolithography and chemical etching (remove excess foil)
  3. Combine all layers if making >2 layer PCB
  4. Drill holes in PCB
  5. Plating to electrically connect all layers to vias
  6. Soldermask applied to top/bottom
  7. All copper surfaces plated (typically lead/lead-free solder, nickel, or gold)
  8. Apply silkscreen
  9. Post-process milling/board de-panelization

Modern PCB Fabrication



Implementing Systems with PCBs

- PCBs are a good way to implement ***Interconnect***
  - Mechanical Mounting
  - Electrical Connectivity
- Many parts are designed to operate solely on PCBs
  - ICs (microcontrollers, regulators, motion controllers, modern sensors, etc.)
  - Discrete components shaped to fit
- Standalone devices interact with PCBs via electrical ***connectors***

Custom PCBs in System Design: Benefits

- **Interconnect organization**
  - Put all your electronic subsystems in one place
  - Combine many “breakout boards” into single PCB
  - Eliminate unnecessary electrical connectors
  - Can increase reliability

- **Form Factor**
  - Extreme miniaturization is possible
  - PCB shape and part placement is definable
  - Can fit very small or unusual shapes

- **Greater choice in component selection**
  - Most modern parts not designed for standalone use
  - Not limited to breakout boards, dev kits, or hobby parts
  - Some electrical designs (high speed or analog) not possible or reliable using breadboards and dev kits

Custom PCBs in System Design: Drawbacks

- **Learning Curve**
  - Takes a little more time to implement than COTS
  - Requires knowledge of electronics to implement
  - May require more testing/validation

- **Failure may be harder to mitigate if entire system is on a single PCB!**
  - Can add additional, optional connectors to salvage
  - Test complex subsystems beforehand if possible
  - Iterating PCB 2-3 times not uncommon

- **Balancing cost/effort against a COTS system**
  - For small design houses, turnaround on a design is generally 1-2 weeks
  - More effort in design, less effort in assembly / wiring
  - A single custom PCB may end up cheaper than 5 breakout boards

Designing PCBs with EagleCAD

Schematic Capture
and
Layout Design

PCB Design Software

- **PCB CAD provides two separate (but linked) tools:**
  - Schematic capture / verification
  - Layout artwork / verification

- **Also provides ability to create parts libraries**
  - Basic building block links logical (schematic) diagrams to physical (layout) parts

- **Some common software packages:**
  - Eagle (Easily Accessible Graphical Layout Editor)
  - ORCAD
  - Altium Designer
  - PADs

Eagle CAD

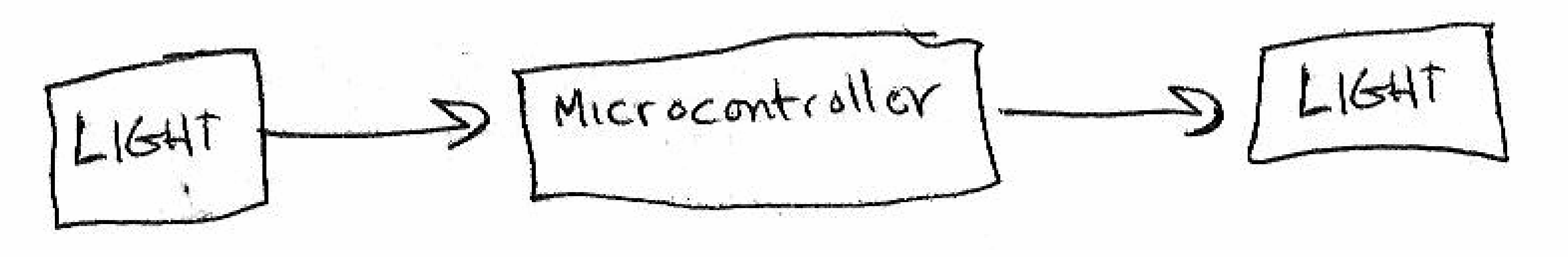
- **In this class we'll be using EagleCAD**
  - Available at http://www.cadsoftusa.com/
  - Used by many--including Arduino designers
  - Commercial software
    - "free" version for non-commercial use
    - "free" restricted: 2-layer, 100mmx80mm size

- **An interesting alternative is DIPTRACE**
  - Available at http://www.diptrace.com/
  - Free, restricted version is also available

- **Also worth considering:**
  - KiCad EDA http://kicad-pcb.org
  - Altium CircuitMaker http://circuitmaker.com/
    - MRSD Wiki: http://cmumrsdproject.wikispaces.com/Altium+Circuitmaker

Schematics: A Picture is Worth a Thousand Words

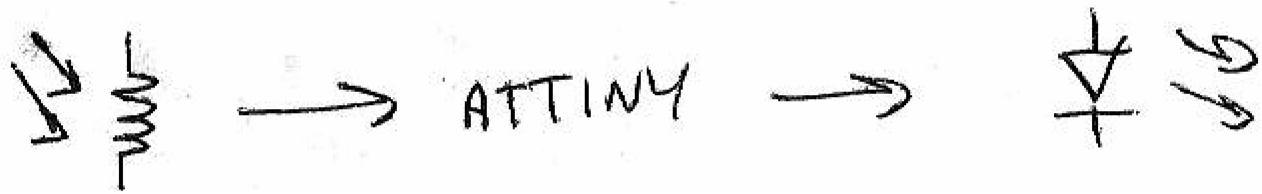
- Schematics represent circuits graphically
- Consider one very basic system:



- A little vague, but still a schematic
- A good schematic shows *what* and *how*

Adding Detail with Symbols

- Using specific symbols tells more information



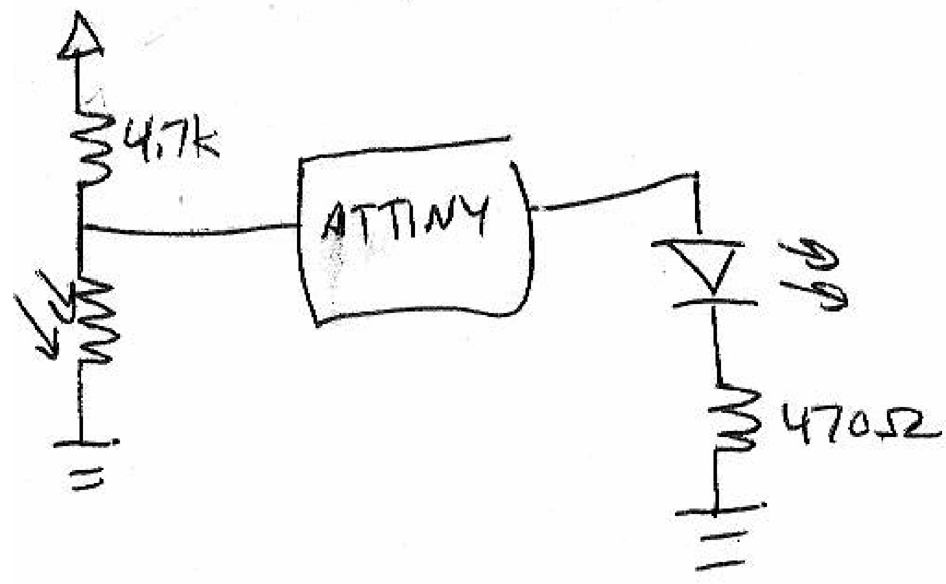
A photocell is read by an ATTiny microcontroller which lights an LED

- Shows you've thought about *what* you're using
- But  is not really a valid electrical device

Adding Detail with Basic Circuits

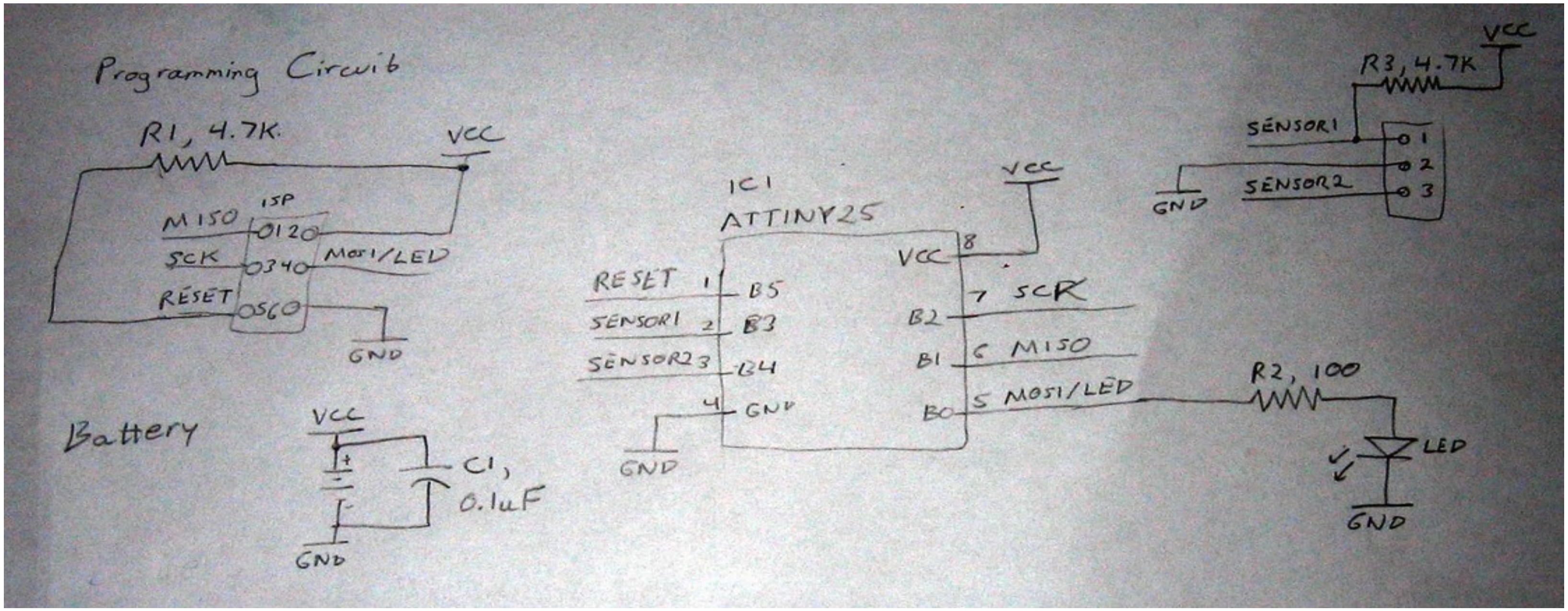
- **Explaining *how* requires circuit diagrams**

A photocell in a voltage divider is read by an ATTINY which outputs to an LED with a current-limiting resistor



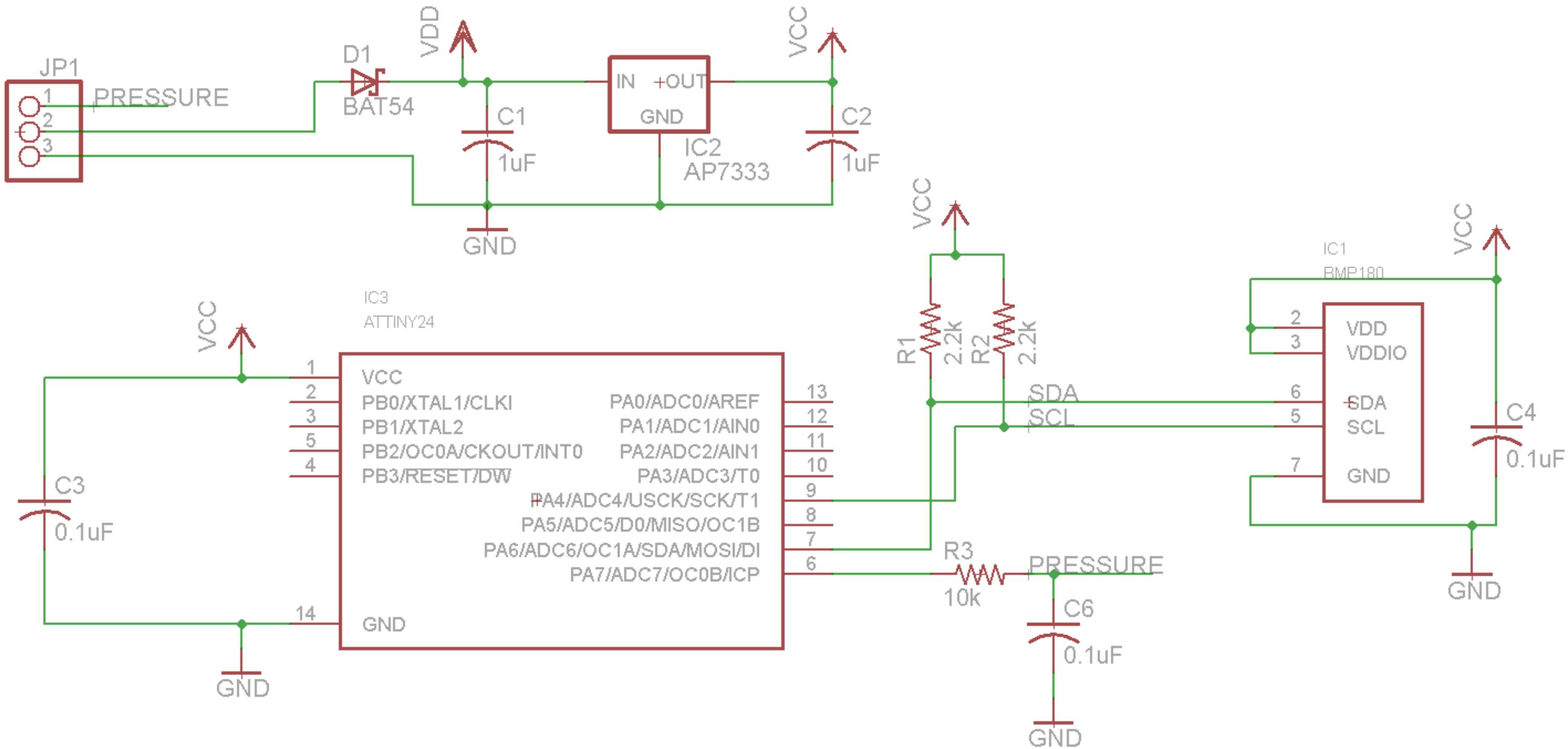
But is this enough?

A complete schematic



- **Lots going on! A *complete schematic.***
- **However, *layout* is not apparent**

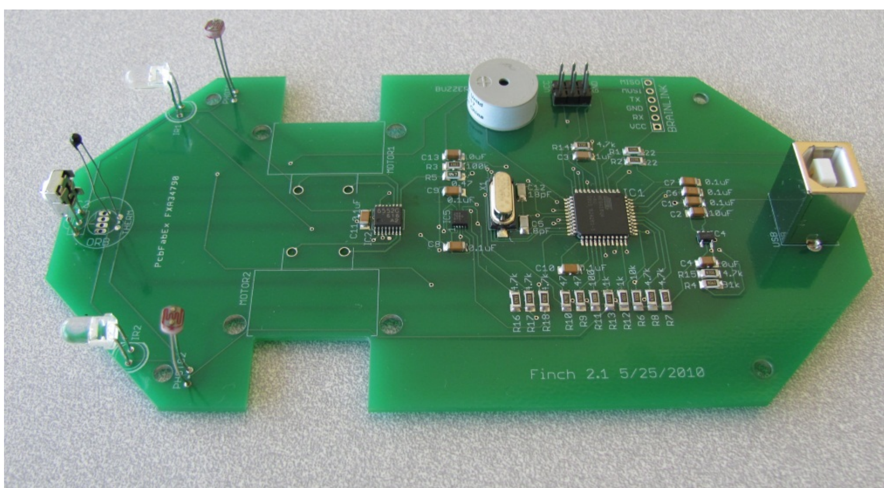
A Complete Schematic in Eagle

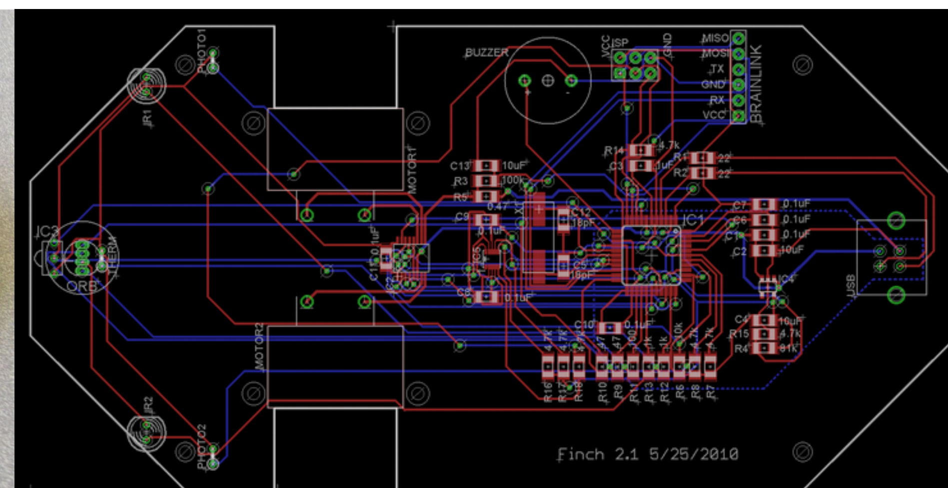


So I've Got my Schematic…

- **Time to make a "Layout"**
  - Scale representation of your PCB

- **Layers**
  - Copper in red and blue
    - Around holes it's in green
  - Designators in white
  - Holes marked with x's





Real World Layout Considerations

- Design Rules
- Surface Mount and Through-hole
- Heat sinks
- Shape, mounting, and 2D representations of 3D parts
- Routing

Design Rules

- Design Rules – "*Thou shalt nots*" for your circuit board

- Adjustable, but…they come from reality:
  - Advanced Circuits Capabilities

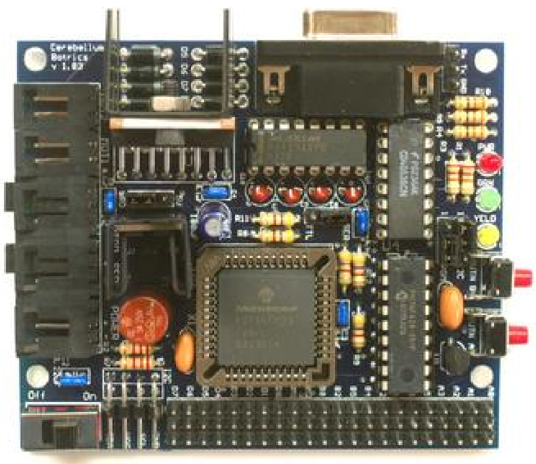
- Tell you things like:
  - How small traces can be
  - How small holes can be
  - Minimum distance between part, pad, trace, and the board's edge
  - Minimum distance between traces
  - And much more!

- You must address any violations flagged by the CAD software

Through-hole Vs. Surface Mount

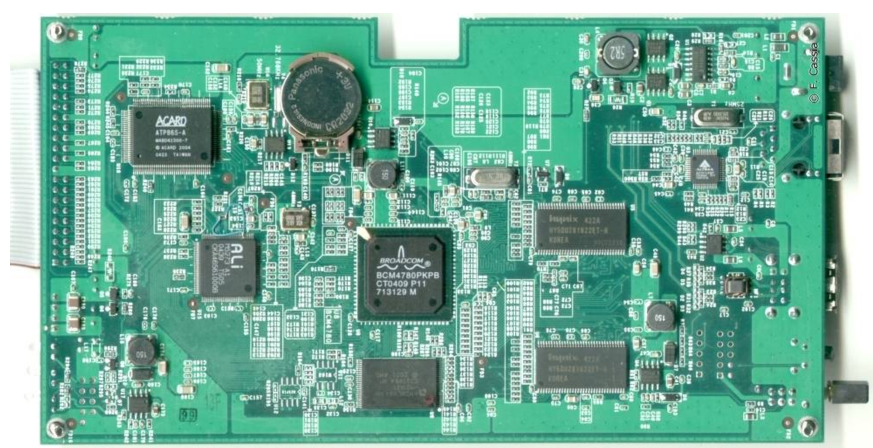
Through-Hole:

- Easy to solder
- Easier to fix circuits
- You don't lose components on the floor
- **Better for large connectors**



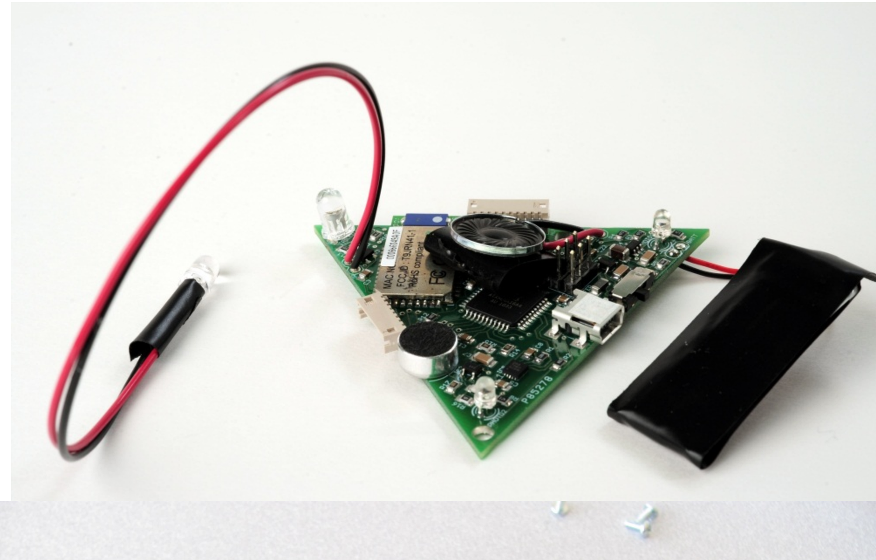
Surface Mount:

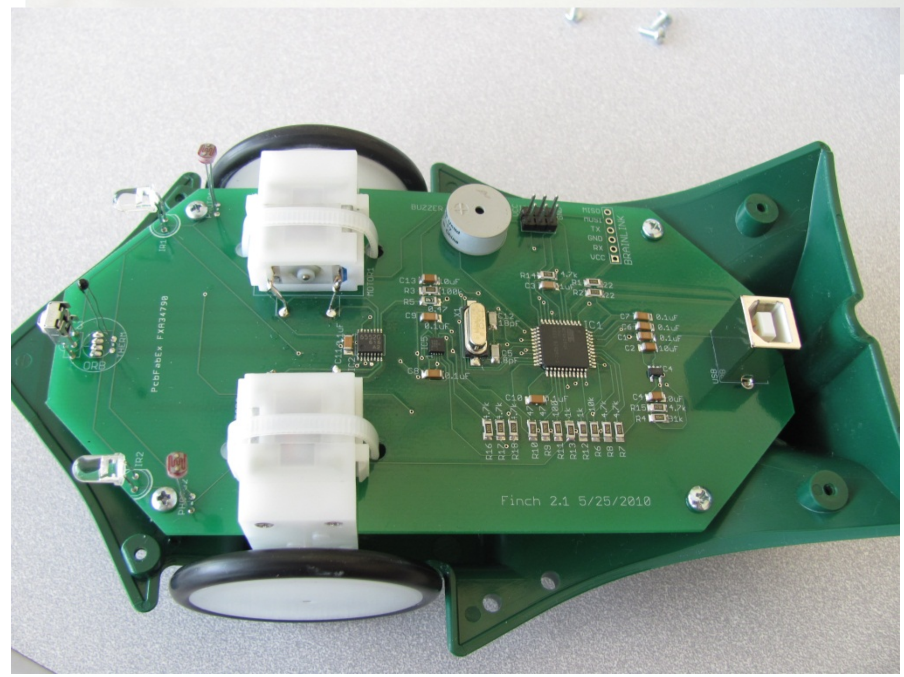
- Miniaturization
  - From small to ridiculous
- Easier Automated Assembly
- PCB Fab is easier – fewer holes to drill
- **Most cool parts are only Surface mount**



Shape, Mounting

- PCB does not need to be a rectangle
- Make sure you allow for a way to mount the board
- Keep in mind – 2D representation of a 3D part!





Heatsinking

- Some high current parts may require a heat sink
- Often much bigger than part
- Specified in datasheet
- Make sure to factor size of heat sink into your layout
- Location is also important
- Insulate electrically !!!

Routing

- **Definition:** ***The process of making electrical connections between the different components by editing the physical (copper) representation of the nets created in your schematic***

- **Your "frenemy," the autorouter**
  - **Computers do the darnedest things**
  - **Fixing things may take as long as routing by hand**

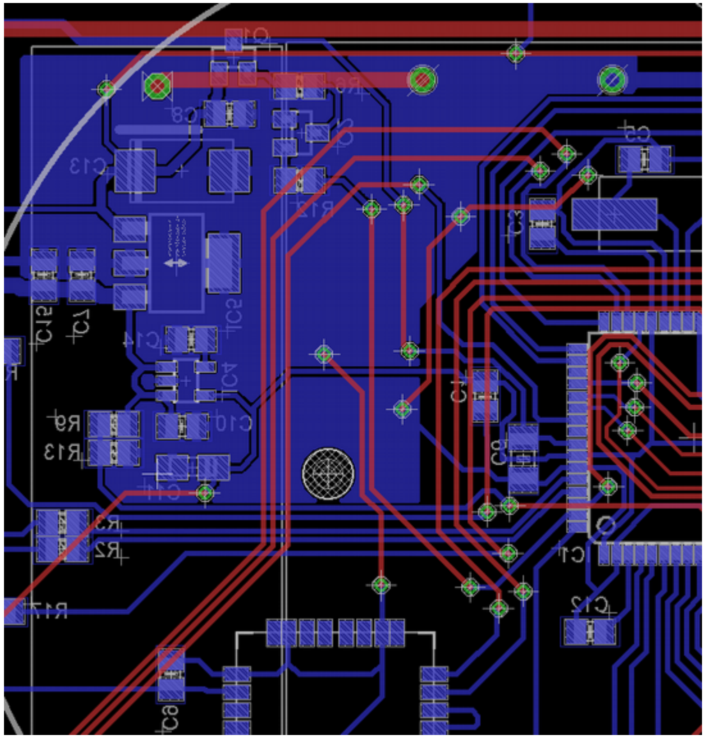
- **Routing by hand creates a much cleaner looking design**
  - **However, it's invaluable for large projects!**

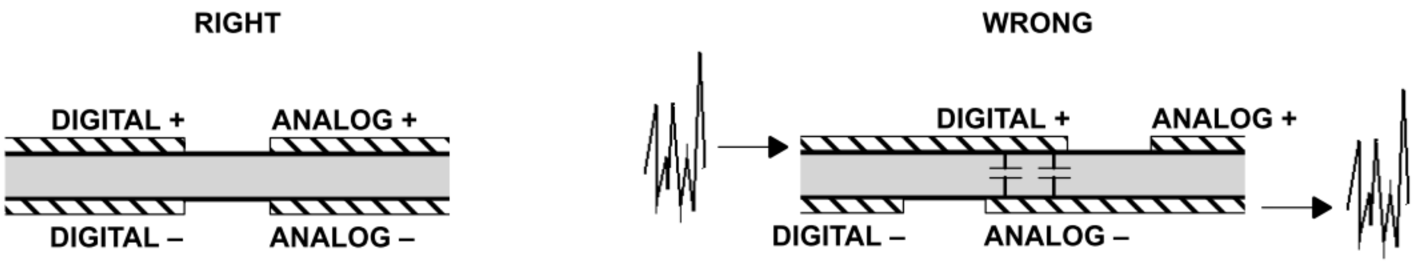
Trace Widths and Power

- All traces/wires have resistance
- Car jumper cables vs. Christmas lights
- Some traces need to be bigger than others
  - Wider traces carry more current (necessary for driving high-current devices like motors).
  - Thin trace carrying lots of current = FIRE
  - Wider traces have less noise (good for analog signals)
- Handy tool:
  - http://www.4pcb.com/trace-width-calculator.html
  - Understand issues with temperature

Planes

- Cover a large part of your board with copper connected to a net (typically ground)
- Used for:
  - Heat-sinking
  - Noise reduction
  - Simplify routing (one less net)

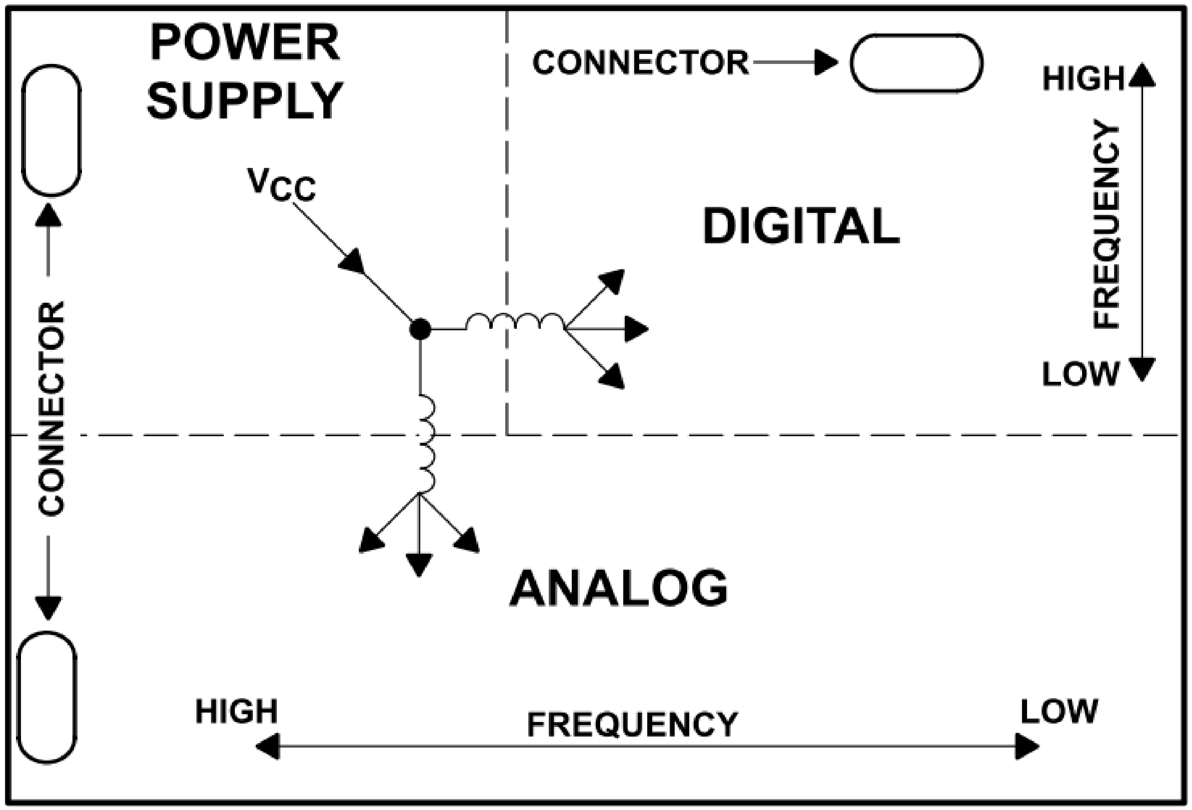




Avoid coupling high-frequency digital noise into analog circuitry

Board Layout

- Careful placement prevents a lot of problems

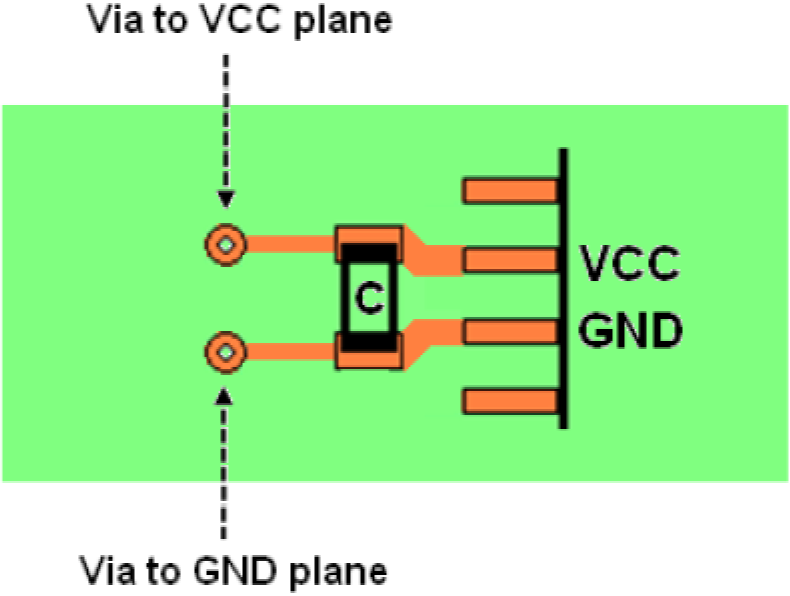


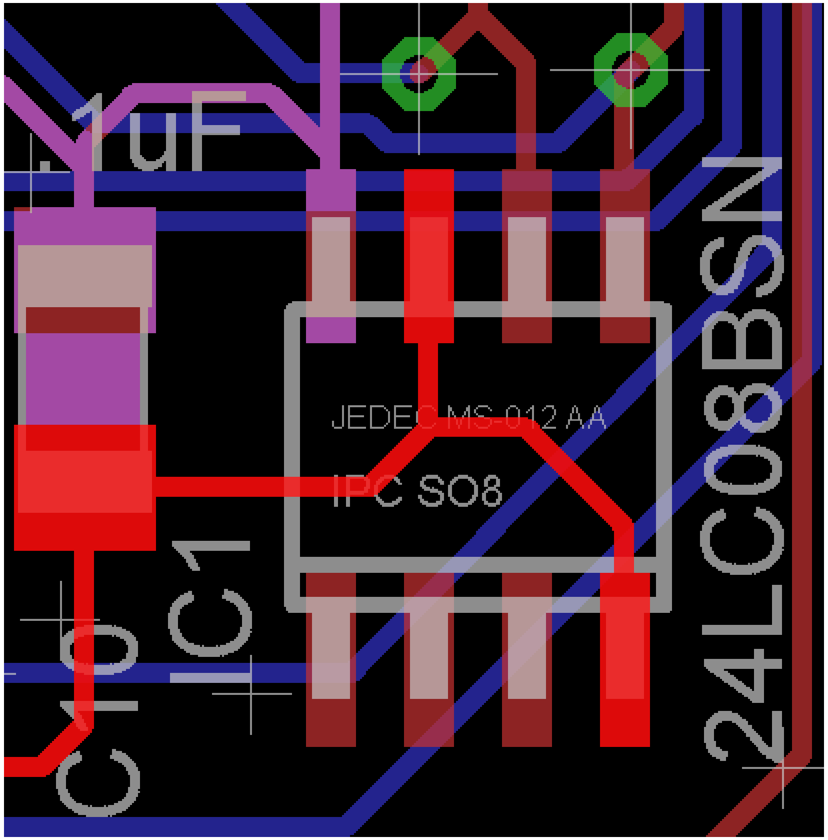
Slight Detour: Decoupling Capacitor Placement

- Short traces (less than 0.5”)
- At least one per IC
  - Usually one per pair of VCC/GND terminals on IC
- Can be on underside if you want to save space
- Following slides – a number of examples of good capacitor placement

Good IC Decoupling

- VCC = PURPLE
- GND = RED



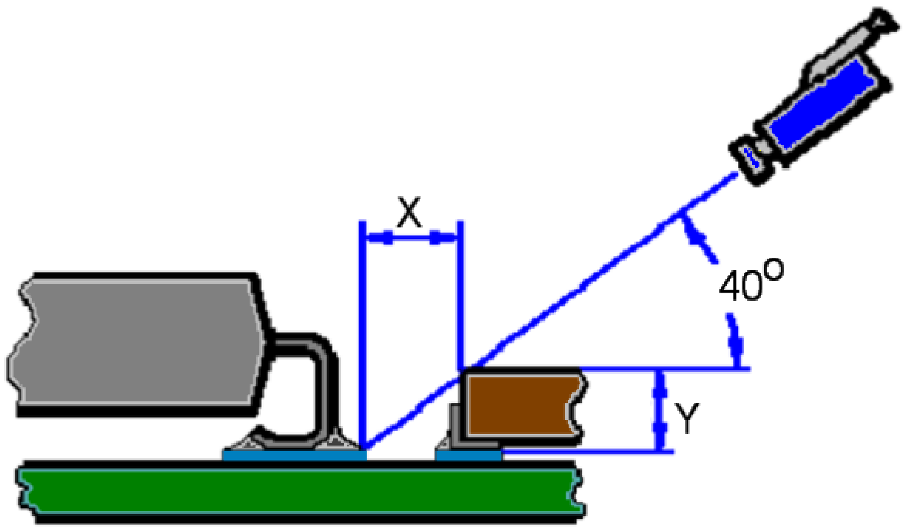


Good Input Capacitors for a Voltage Regulator

GND = Red, VDD = Yellow, VCC = Purple

Design for Testing

- Test points
- Tooling pins
  - Precise registration with test fixture
  - Push fingers
- Clearances for inspection



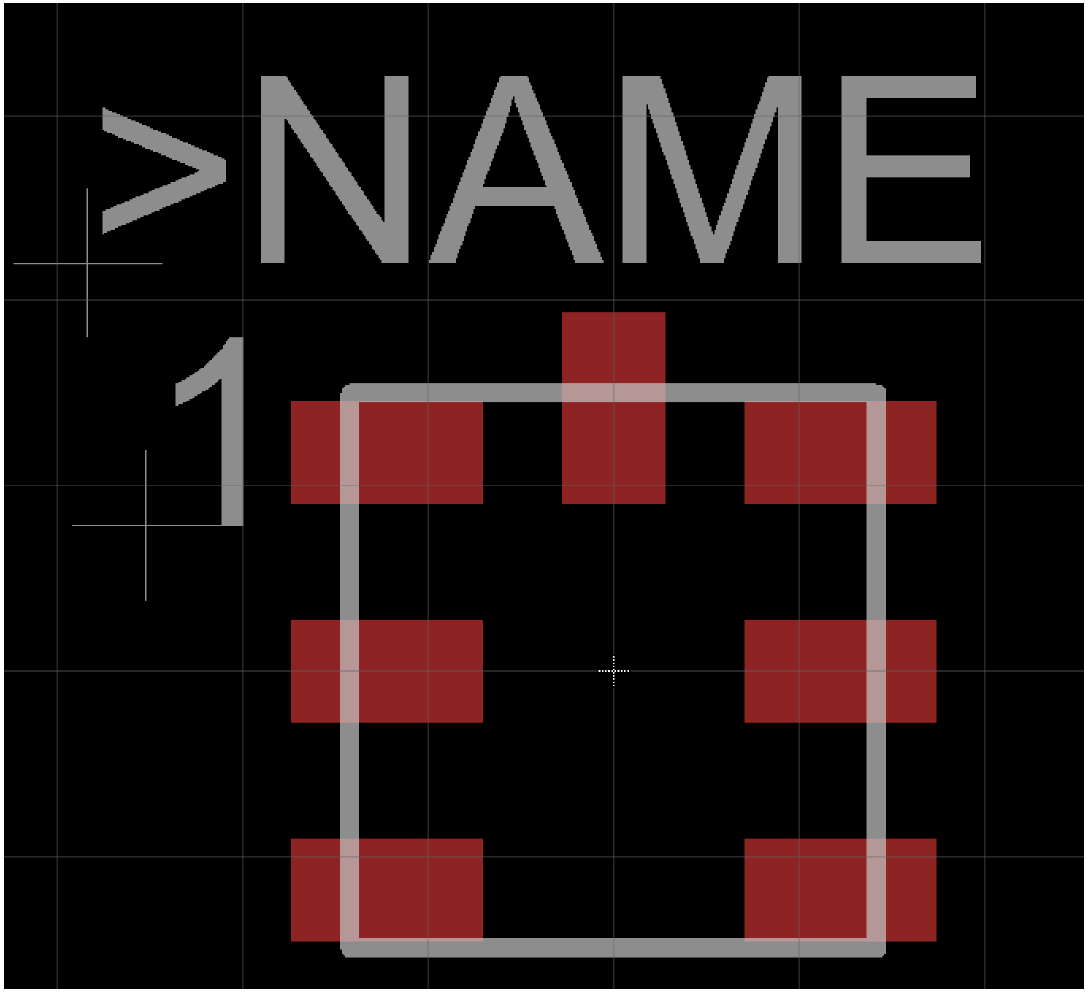
CAD Circuit Board Layout Process

- Set DRC
- Place components
- Route electrical connections between them
- Place designators
- Verify results

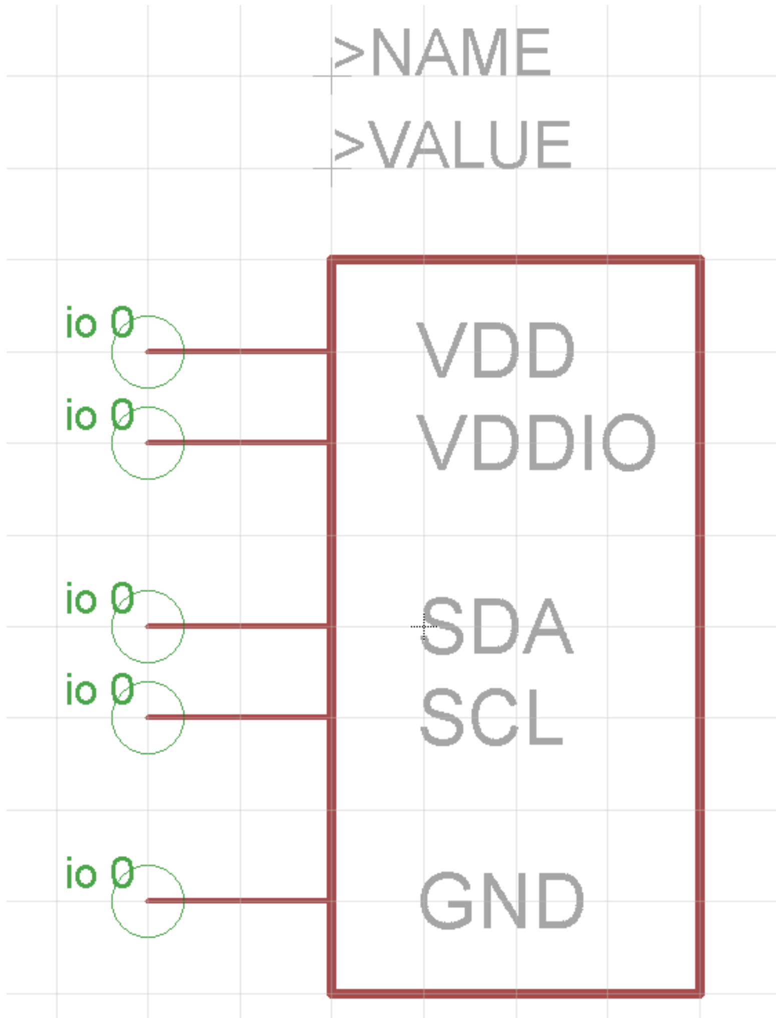
So where do parts come from?

- Answer: Libraries!
- Every part is divided into three representations
- Package
  - The physical representation of the part
  - What you see in Layout view
- Symbol
  - The logical representation of the part
  - What you see in Schematic View
- Device
  - A place to tie the package to the symbol
  - Lets you re-use common packages on different parts
  - Lets you re-use common symbols on package variants

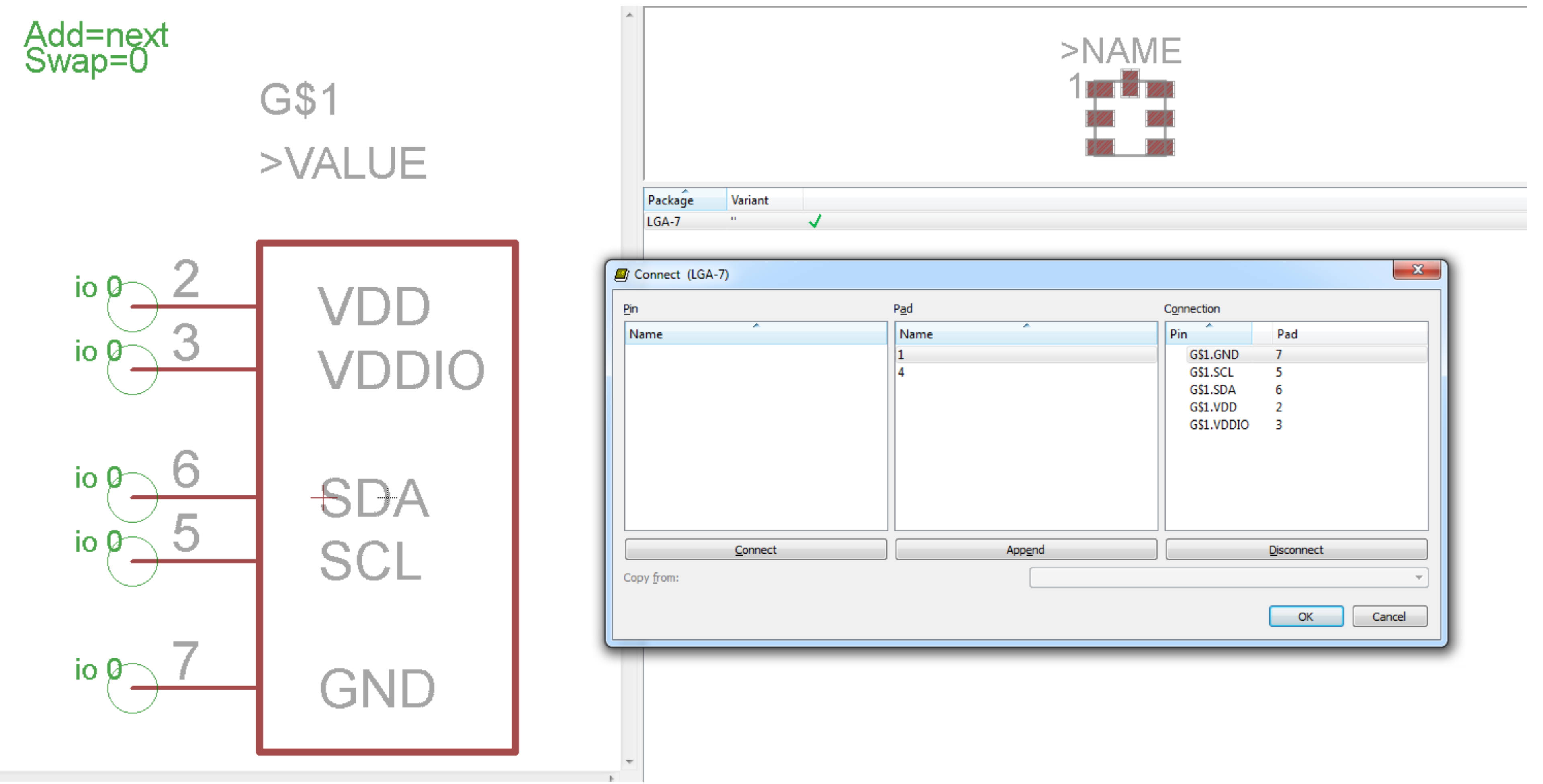
Sample Package View



Sample Symbol View



Sample Device View



Where do libraries come from?

- EAGLE standard libraries
  - Some are great, some date from the time of disco
- Online open-source libraries
  - Contributed libraries on Eagle's site
  - Sparkfun, Adafruit, etc.
- You!
  - Complex or unusual parts are probably not in any library

Converting a System Diagram into an Electrical Schematic

Implementation Guidelines for 4 System Types

- Control
- Sensing/Actuation
- Power
- Interconnect

Implementing Control

- Choosing a microcontroller
  - AVR, PIC are popular choices
  - TI OMAP, Freescale, ARM7, ARM9…
  - Older architectures like 8051 or 16HC11
- Also OK to plan on using a *microcontroller board* and interfacing to its connectors
  - “Arduino shield”
  - Sensor/actuator boards (*a la L293 driver board)*
- Others (higher degree of integration/capability)
  - Raspberry PI, Beaglebone…

Implementing Control

- Don't forget the baggage
  - Usually specific programming infrastructure required ("toolchain")
  - A bunch of extra circuitry, specifically:
    - Required voltage regulation / decoupling
    - Programming connector (even if using a bootloader!)

- We usually use AVR due to availability of OSS `avr-gcc` compiler and `avr-dude` compiler. AVR Studio (windows only) also available free.

- Arduino
  - Uses AVR for the same reason
  - Also has bootloader / Processing interface for programming
  - If you like/need Arduino interface, consider making your own
    1. Download open source eagle files for your favorite Arduino version
    2. Modify design to add additional sensor/actuator interfaces to PCB
    3. Make PCB and re-release design files

Important Things to Remember on Control

- Voltage Compatibility
  - Modern ICs run at 3.3V or 1.8V, not 5V
  - If you need to hook up to a junky old sensor, may need level-shifting to prevent damage

- Make sure to add debugging output
  - Even a blinking LED helps iron out software kinks
  - A serial (or USB<->serial) port makes it even easier

- Basic system testing – aim for a few simple milestones
  - "Blink test" is a good first program
  - Each subsystem should be able to be independently tested and verified, even if only in debugging mode

Sensing and Actuation

- Sensing/Actuation lumps together subsystems that either:
  - Measure the physical world and produce digital or analog values for a controller, or
  - Take digital or analog values from a controller and effect change on the physical world
- Some sensors:
  - Light sensors (photocells, cameras, photodiodes, etc.)
  - Temperature sensors (thermistors, thermocouples, etc.)
  - Contact sensors (buttons, limit switches, pressure pads, etc.)
  - Rangefinders (IR pairs, IR rangefinders, sonar, laser rangefinders, etc.
- Some actuators:
  - Light sources (LEDs, lamps, neon, etc.)
  - Motion (Motors, servos, steppers, solenoids, piezo elements, etc.)
  - Sound (speakers, buzzers, piezo elements, etc.)

Support Circuitry

- Most sensors/actuators require additional circuit elements to operate, such as:
  - Discrete components (capacitors, resistors)
  - Additional, non-logic level voltage sources (especially for motors)
  - Level translation
    - Communications protocols (e.g. RS232/485 to TTL)
    - Also commonly for old 5V components interfacing to modern ICs
  - Electrical connectors to connect off-board components
- Most devices explicitly tell you what support circuitry is needed in their **datasheets** (more on future lecture)

Actuators Brief: Motion Control

- Always use the least amount of control needed!
  - Only need to drive one direction? MOSFET/relay
  - Bi-directionality requires H-bridge & support circuits

- Aim for reasonable voltage/current
  - 5V/500mA is easier than 50V/50A
  - 24V@2A is easier than 2V@24A   [why?]

- Many monolithic options available
  - Driving 1-4 motors can be as simple as one IC, a few capacitors, and your connectors
  - L293 == L298 with integrated flyback diodes
  - Many MOSFET designs / ICs do not require flyback

- For high power (>10W), it may be easier to look into hobby servos, off-board modules or motion control amps.

Implementing Power

- **Power** is mechanical work and also [**voltage X current**]
- All electromechanical devices require power to operate
- All modern electronics perform logical operations while generating heat
- When implementing a power system, understanding the **bounds** is essential:
  - Voltage ranges (min / max)
  - Current draw (quiescent / peak)
  - Storage capacity (mass / volume)
  - Runtime

Classifying Different Power Systems

Consider:

Voltage Current Capacity Runtime

- Digital watch
- Wireless sensor node
- R/C Car
- R/C Quadrotor
- Computer performing video analysis
- Electric car

How Power Affects PCBs

- May need multiple voltage sources
  - Motors may need higher voltage than logic
  - Different logic may require different voltages
- Current draw affects trace width and component size
  - Big currents require big traces
  - Regulators for 100mA much smaller than for 2A

Voltage Regulation

- Most systems designed to operate from a single source
  - Wall (A/C) via AC/DC converter
  - Battery pack
  - Source voltage is usually optimized for largest power consuming subsystem (e.g. motors)
- DC/DC converters used to create other subsystem voltages
  - Logic voltage levels (frequently 1.8V, 3.3V, 5V)
  - Analog supplies (+-5V, +-15V)
  - Other (500V photoflash bank, 5kV neon ballast)
- DC/DC converters also used to **regulate** voltages
  - Stable voltages are essential for logic ICs
  - Helps isolate subsystems from each other

Voltage Regulation

- **Linear Regulators**
  - Can only step down voltages
  - Dissipates excess as heat [voltage X subsystem current]
  - Usually very clean signals
  - Low cost and complexity
  - Excellent for small voltage drops

- **Switching Regulators**
  - Allows output voltage to be above and/or below input voltage
  - Efficiencies typically 80-96%
  - Higher cost
  - Variable complexity – but Point of Load replacements available!
  - Excellent for large voltage differences

Minimizing current can minimize power losses / regulator size

Linear vs. Switching Efficiency

- **Desired Output Voltage: 5V**

Input Voltage	Linear Efficiency	Switching Efficiency
5.5V	91%	80-90%
6V	83%	80-90%
12V	42%	80-90%
24V	21%	80-90%
3.3V	Not possible	80-90%

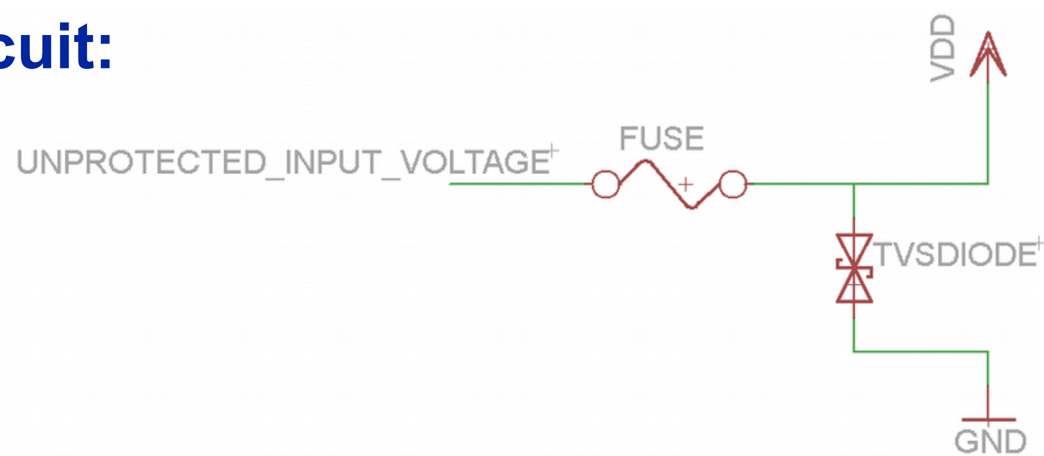
- **Efficiency is not everything. 21% efficiency might be just fine if you're drawing 500uA.**

Assembling a Power System

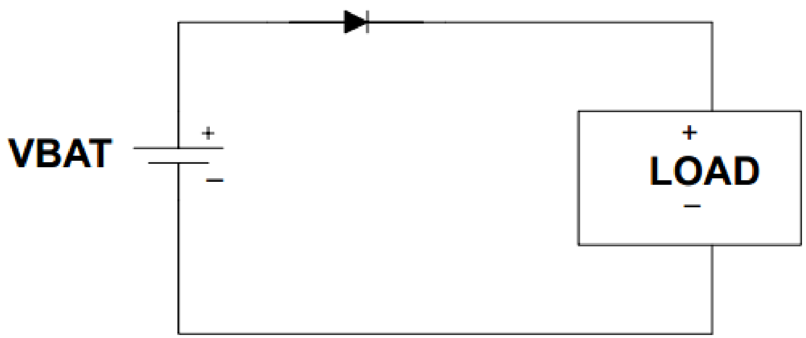
- Determine source voltage and current requirements
  - Optimize for primary power using subsystems
  - Add current to account for other subsystems *and conversion efficiencies*
- Determine voltages and current limits of regulated voltages
  - Primary power systems frequently run *unregulated*
  - May need several logic voltages or other supplies
- Other useful power system features
  - Switches (e.g. ON/OFF) are helpful
  - Fuses / breakers may be needed
  - Voltage and current monitoring
  - *Battery charge management*

Overvoltage and Reverse-voltage Protection

- It's 3 AM, and you just plugged your robot in backwards. Would you like to:
  - A: Change a fuse?
  - B: Put out a small fire, go through the stages of grief, and then solder together new PCBs, order new parts, and explain to your professor why your project is going to be a week late because nobody from your group remembered this slide
- Here's a circuit:

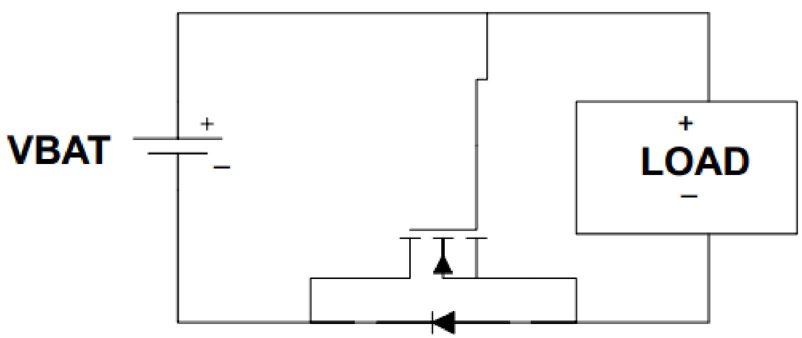


Reverse-Voltage Protection

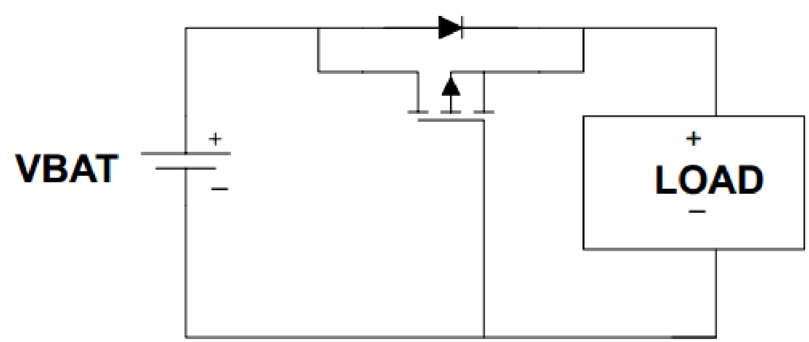


Diode in series with load

- Voltage drop
  - Batteries
  - Efficiency
- Use Schottky diode
  - Cost



NMOS FET in the ground return path



PMOS FET in the power path

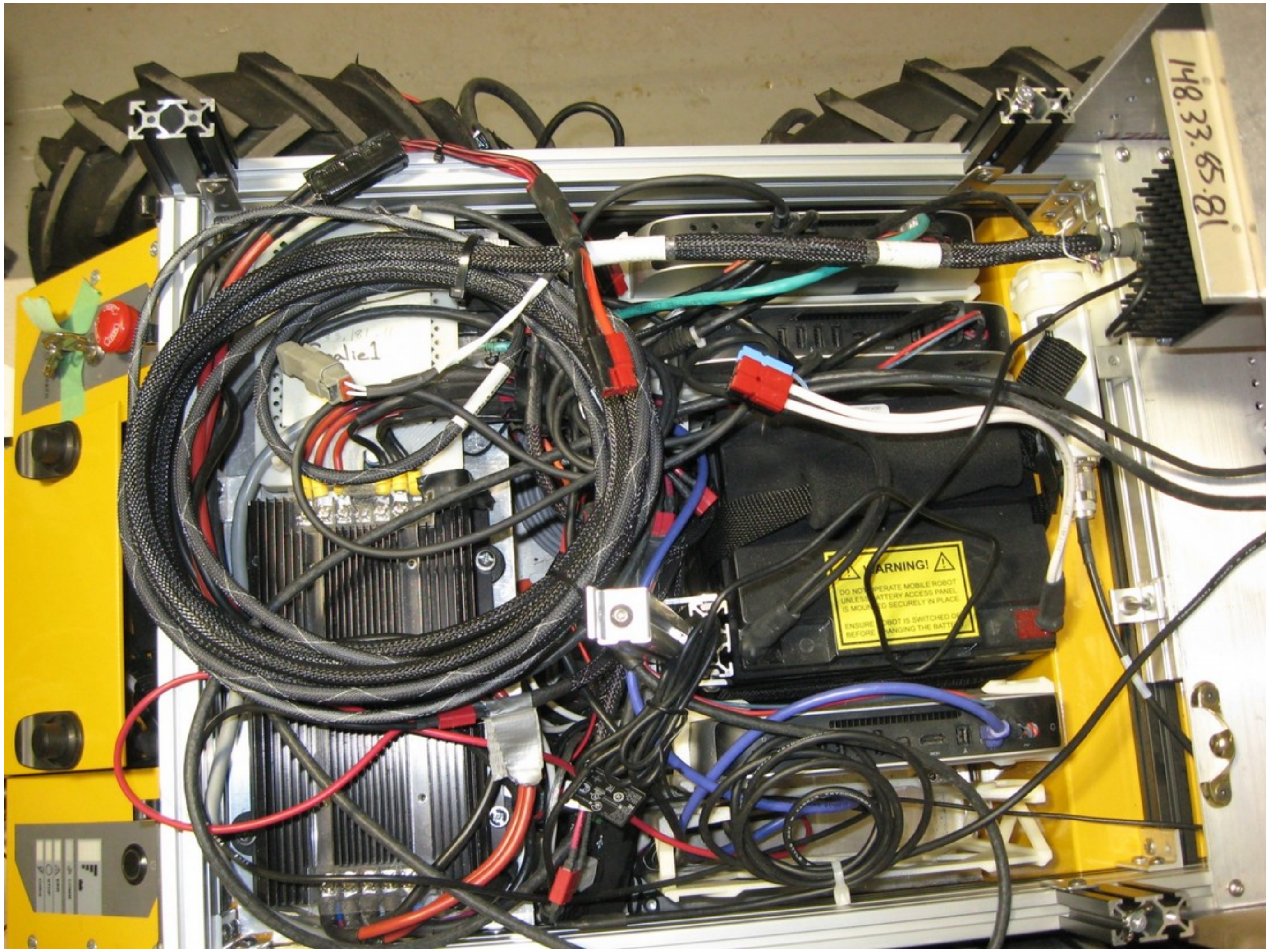
Don’t Forget the Capacitors!

GND = Red, VDD = Yellow, VCC = Purple

Using Interconnect Appropriately

- Wire (and PCB traces) should be sized appropriately for current loads
  - ‘PCB Trace Width Calculator’ is invaluable
  - Many wiring tables for various gauges show current capacity
  - Use stranded wire for connections!
- Avoid analog effects on your digital designs
  - Use short, thick traces for analog signals
  - Keep analog away from fast switching digital signals
  - Use ground planes or guard bands when appropriate
- Use appropriate connectors
  - Every connector is a potential point of failure
  - Observe current and connection cycle ratings
  - Use board-to-board when possible instead of board-wire-board

System Integration



Assignment Overview: Individual Assignment (Task3)

- **Due: Tue Sep. 26th (One week from today)**
- **Capture a hand-drawn schematic into Eagle, and make two layouts for it:**
  - One all through hole
  - One all surface mount
- **Create Eagle library versions of two parts from their datasheets**
- **Make sure to check out the two tutorials posted on Blackboard, and send questions to us**
- **Please, make sure that you**
  - Follow the file naming guidelines
  - Address any issues reported by ERC, DRC

Assignment Overview: Group Assignment (Task 6)

- **Due: Fri Oct. 6th**

- **Create a schematic and layout for a power distribution PCB**
  - Will need to choose appropriate regulators for each subsystem
  - Will need to design (from datasheet) Eagle library parts for those regulators
  - Will need to include OVP and RVP protection, and keep in mind the design considerations (like trace widths) mentioned in this presentation
  - Should be good preparation for the PCB you create for your own robots later in the semester

Teamwork and PCBs – Divvying efficiently

- Schematic accumulation
  - Everyone can contribute subcircuits
  - One person in charge of overall schematic / collating contributions
- Part hunter
  - At least one person should be searching for parts
  - Parts should be **available** and **captured into Eagle**
- Verifier
  - Dedicated to finding logical errors in schematics or layout
  - Goes through internet/team created libraries and checks parts
- Layout and physical space planner
  - Determine board size and shape
  - Board mounting hole placement
  - Determine where all connectors should be located
  - Ensures parts sticking off board won't interfere with chassis