Modular Design & Circuit Debugging Tips

ECE 445/ME 470 ZJUI Senior Design

Dean Biskup dbiskup2@illinois.edu

What is Modular Design?

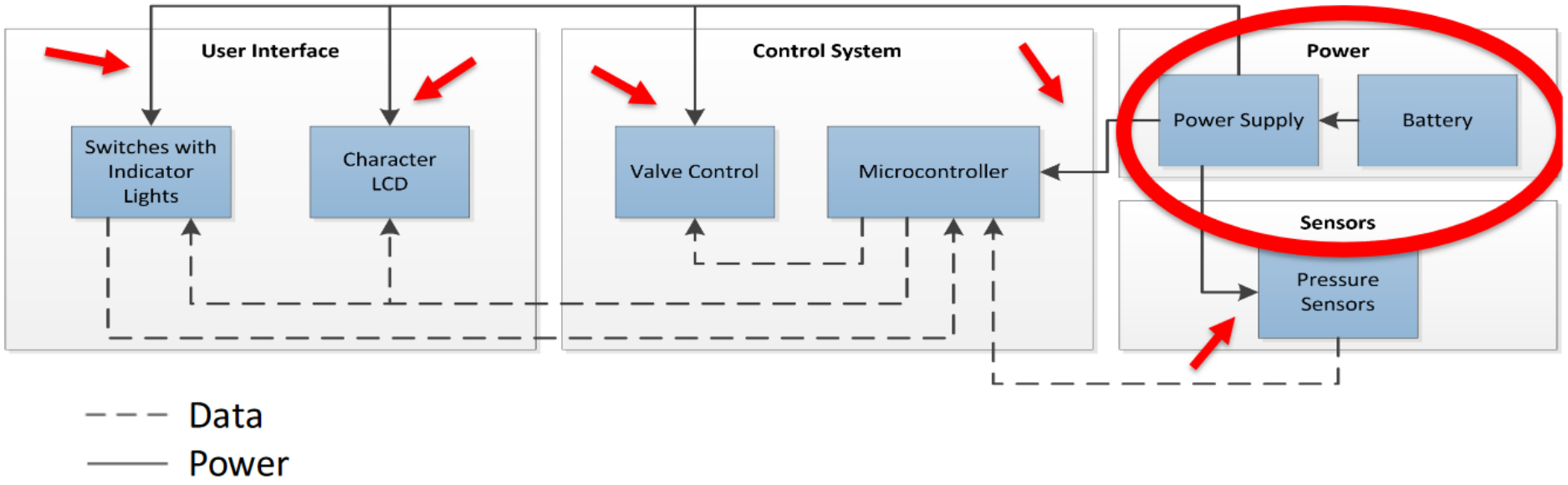
- Modular - Employing or involving a module or modules as the basis of design or construction.
- Each section of your design is defined with **inputs** and **outputs.**
  - Break down your project into well-defined blocks.
  - You already did this in your Block Diagram!
- Make sure you continue with this philosophy as you design the implementation of your project.

Why Modular Design?

- Makes complexity manageable.
- Enables parallel work (very important!)
  - Your team has 4 members that should all be working concurrently!
- Accommodates future uncertainty.
  - If your modules work but your full project doesn't, you can still get partial credit in the Demo.

Good Modular Design

- You should have seen a few of these already in Proposal information.
  - Key is that nothing is ambiguous. All information is clear.



Key Benefits of Modular Design

Helps with:

- Separation of work
- Communicating design to others
- Defining Requirements and Verification
- Debugging
- Partial credit in case your project doesn't fully work!

Circuit Debugging

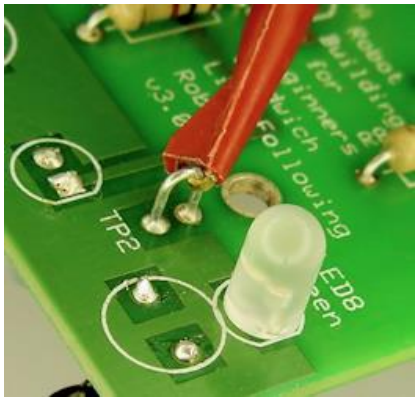
Best to test each module separately

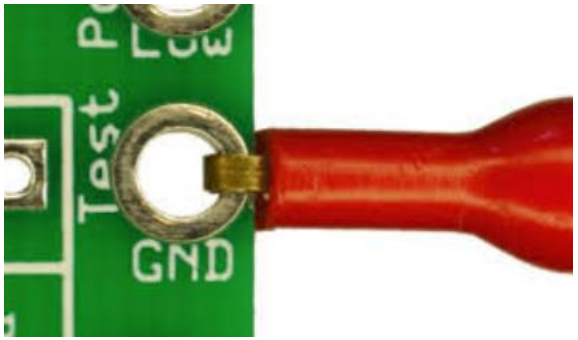
- Easier than troubleshooting the entire project at once.
- Requires a well-defined block diagram to be useful.

Circuit Debugging

Add “test points” to your circuit.

- Places on your circuit you can probe for certain signals.





Becoming a Good Design Engineer

- Break complex circuits into manageable blocks.
  - With accessible I/O for probing!
- Troubleshoot problems at a modular level.
- Understand previous approaches to the problem.
  - Both the pros AND cons!
- End goal of Senior Design:
  - Solve new problems in innovative ways.
  - Learn about the process of creating real projects.

Circuit Design Tips, Tricks, and Common Issues

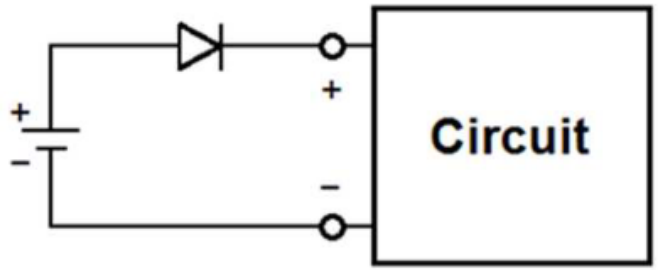
ECE 445/ME 470 ZJUI Senior Design

Dean Biskup dbiskup2@illinois.edu

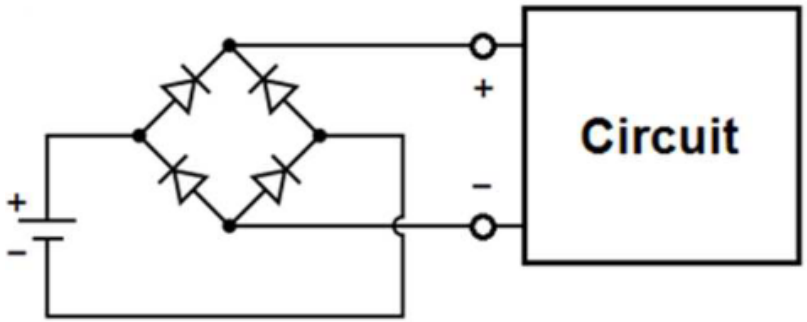
UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Reverse Polarity Protection

- 2 common methods



Simple Diode
- Circuit will not operate with incorrect polarity



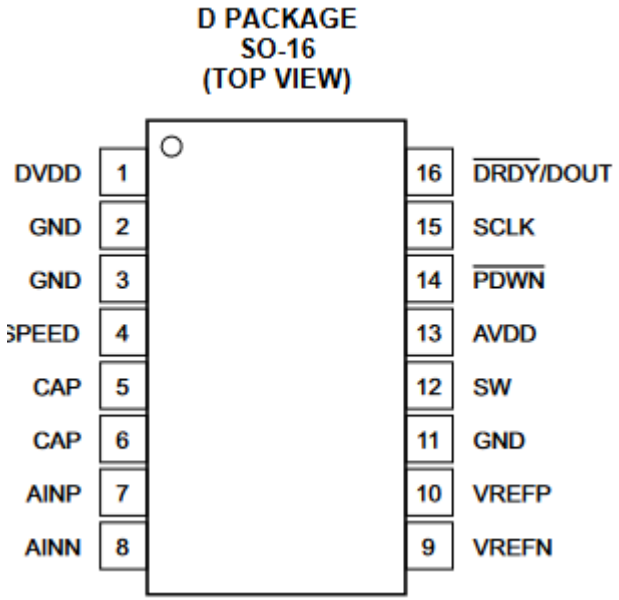
Diode Bridge
- Circuit will operate under either polarity
- Higher losses

Driving High-Current Loads

- Most microcontrollers can drive a max of 20mA.
  - Interface with a gate
- Methods:
  - Relays
    - Simple but may wear out
    - Are “slow” (compared to transistors)
  - Transistors
    - Fast switching, but more expensive/amp
  - H-Bridge
    - More involved but can do both forward and reverse current.
    - Good for motors, but you probably want a dedicated motor driver IC.

Must: Read the Datasheet!

- Datasheets often have common implementations.
  - Will tell you if you need decoupling capacitors or other passives.
- Also contain ordering information and sizing.
  - QFP packages are hand-solderable. Smaller packages are a little tougher.
- Also includes pinouts.



https://www.ti.com/lit/ds/symlink/ads1131.pdf?ts=1646869185550

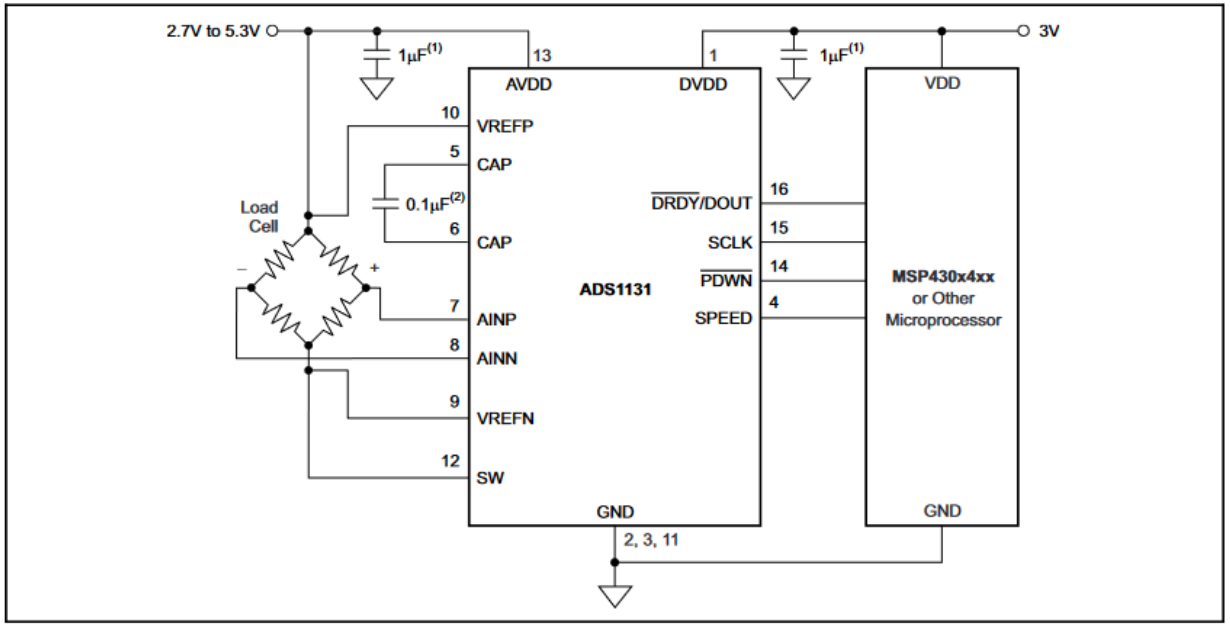
Must: Read the Datasheet!

- Ex. ADS1131 18-bit ADC from Texas Instruments

APPLICATION EXAMPLE

Weigh Scale System

Figure 10 shows a typical ADS1131 application as part of a weigh scale system.



(1) Place a 0.1µF or higher capacitor as close as possible on both AVDD and DVDD.

(2) Place capacitor very close to the ADS1131 CAP pins for optimal performance.

Figure 10. Weigh Scale Example

https://www.ti.com/lit/ds/symlink/ads1131.pdf?ts=1646869185550

Troubleshooting Steps

1. PCBs: Remove/disconnect power and measure DC resistance at power rails (VDD-to-GND) with a multimeter.
   - If reading is less than 100Ohms, you may have a damaged part somewhere – DO NOT POWER ON.
   - If possible, try removing parts 1 by 1 to see if the reading increases.
2. Power on. Check supply voltages with multimeter.
   - If any rails show 0V: is power actually plugged in? Fuse blown?
3. Probe signals at intermediate stages or at test points.

Troubleshooting Steps

4. Check interconnections.
    - Signals incorrectly wired?
    - Are any wires loose/contacts bad?
    - Is any signal floating?
5. Double check the design.
    - Check the pinouts.
    - Check that you have the correct datasheet or part number.
    - Ensure correct polarity.
6. Possible burnt components.
    - Last resort! This is usually **not** the case.
    - Isolate parts from circuit and test individually.

Common Circuit Checklist!

- ❑ Use connectors!
  - ❑ These connectors should be standardized.
- ❑ Proper labels so that you know what you are doing later.
- ❑ External oscillator required if you have strict timing?
- ❑ Do you have enough memory on your microcontroller?
  - ❑ Do you need persistent storage e.g. flash?
  - ❑ Do you need dual-core or is 1 core enough?
- ❑ Do you have capability for In-System Programming (ISP)?
- ❑ If you have a battery, how do you charge it?
- ❑ How will your PCB be mounted?
- ❑ Component packages all sized correctly?
- ❑ Debugging/Test points?
  - ❑ You may also want to add status LEDs.