# EECS 373
## Introduction to Embedded System Design

Robert Dick
University of Michigan

Lecture 10: Prototyping

7 March 2024

# Review

- ADCs and DACs
  - Value and temporal resolution.
  - Sampling frequency and averaging.

# Outline

- **Misc project-related applications and examples**
- Prototyping

# Timer review for project

- Initialize linked list with period, time remaining, and function pointer.
- Sort list in order of time left from shortest to longest.
  - Or create data structure that can never be out of order.
- Start hardware timer with shortest time left in hardware counter register.
- Hardware timer interrupts.
- Call callback via function pointer.
- Subtract elapsed time from time remaining.
- If current timer is repeating, add period to time left and re-insert.
- Otherwise delete.
- Sort list in order of time left.
- Start hardware timer with shortest time left in hardware counter register.
- Hardware timer interrupts.
- Repeat.

# What if you need things in sorted order?

- E.g., virtual timers.
- Create data structure that can never be out of order.
  - Elegant.
  - Efficient.
  - May not be flexible enough for some applications.
- Write your own sort?
  - Time consuming to debug, especially for efficient sorts like quick sort and merge sort.
- Reuse C standard library.
  - Make array of list nodes.
  - Write comparison routine that takes list node pointers.
  - Call qsort().
  - Reconstitute list from array.

- Design rule: If there's something close that is already written and debugged, and fast enough, use it even if you need a shim.

# Definition: bit banging

- Using software to directly set pin values instead of setting parameters in special-purpose hardware.
- Particularly for communication protocols.
- Instruction processor takes responsibility for timing and other aspects of protocol.
- Flexible.
- Keeps processor occupied.
- Wastes power (chainsaw when scissors might be better).

# Struct packing 1

```
struct wasteful {
    char b; // 1 byte
// What goes here?
    char *p; // 4 bytes
    char c; // 1 byte
// What goes here?
    int x; // 4 bytes
};
```

# Struct packing 2

```
struct compact {
    char *p; // 4 bytes
    int x; // 4 bytes
    char b; // 1 byte
    char c; // 1 byte
// What goes here?
};
```

# Singly-linked lists and head nodes

- This idea improves debugging and maintenance, not component efficiency.
- Use header node in empty list.
- Makes operations consistent.
- Removes conditionals.
- Reduces bugs.
- Example.

# Outline

- Misc project-related applications and examples
- **Prototyping**

# Prototyping: why?

Get this wrong → won't finish/debug design.

# Prototyping: what often happens

- Somewhat O.K. design.
- Prototyping errors dramatically increase space for bugs to hide in.
- Days to weeks of debugging.
- Mixture of prototyping flaws and design errors.

# Prototyping: what should happen

- Somewhat O.K. design.
- Methodical, flawless prototyping dramatically reduces hiding spaces for bugs while increasing prototyping time by only minutes.
- Hours of debugging.
- All of it on design errors and (rarely) faulty components.

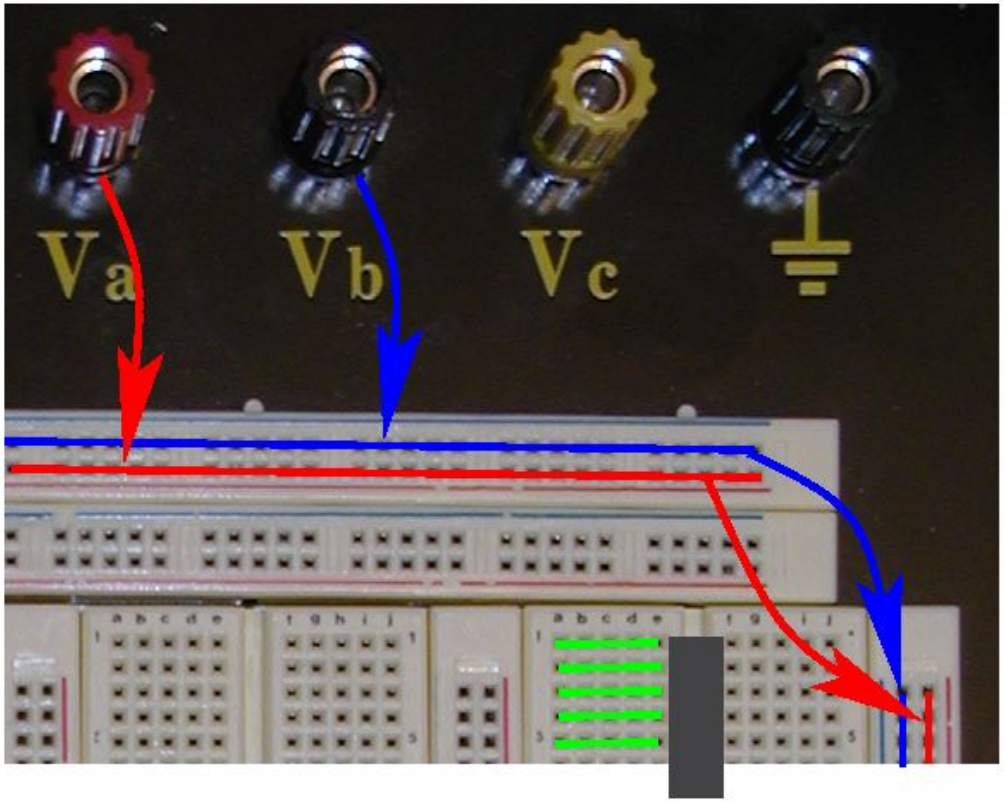
Be obsessive about knowing your tools!

# Prototyping topics

- Options.
  - Breadboards.
  - Soldering.
  - Wire wrap.
- Chilling.
- Noise sources.
- The guild handshake of computer engineers.
- ESD.

# Breadboards

- Quick.
- Easy.
- Horrible.
  - Unreliable.
  - Low-frequency.
    - <=1 MHz usually safe.
    - 10 MHz works, on good days.
  - Nasty parasitics.

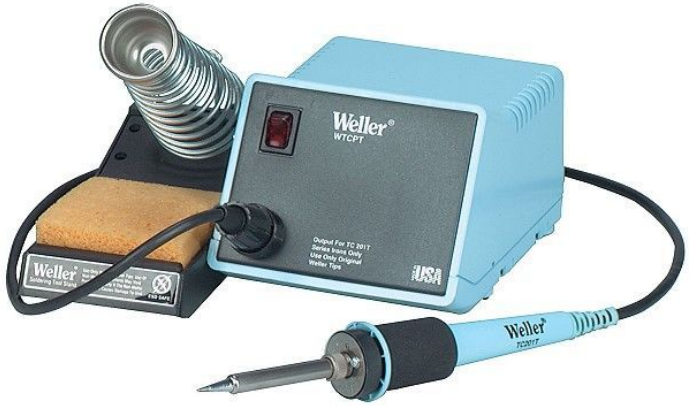


# Soldering

- Slow.
- Requires skill.
- Reliable.

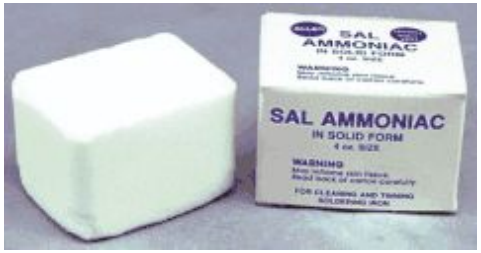
# Soldering tools

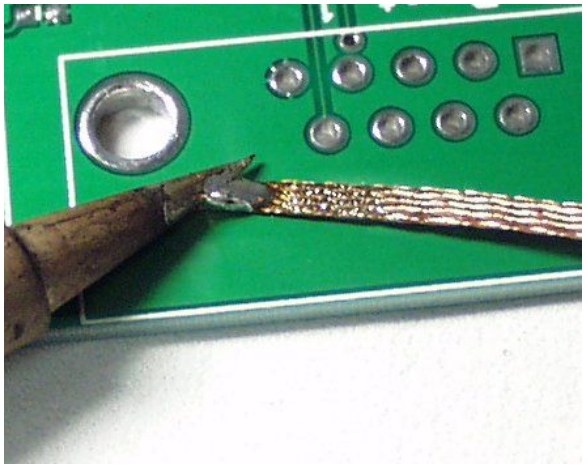
- Soldering iron.
  - Sheath: Check before heating.
  - Temperature: 370 degrees C is a good guideline for through-hole.
  - Allow to fully heat before starting.
  - Don't leave on unnecessarily.
    - Tip oxidizes.
  - Keep clean and tinned.
- Sponge or scrubber.
  - Used in tinning process.

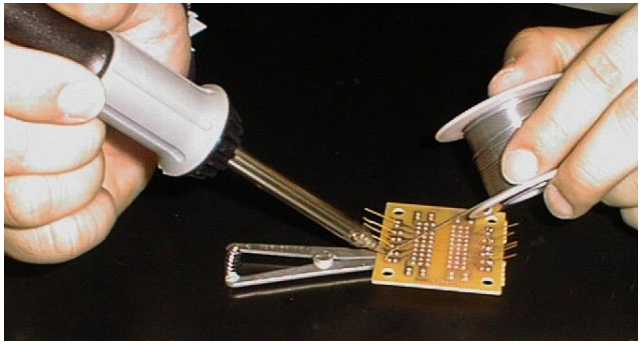


# Soldering tools

- Solder sucker used for desoldering.
- Wick used for desoldering.
- Tinning block.
- Heatsink: used to prevent component damage.





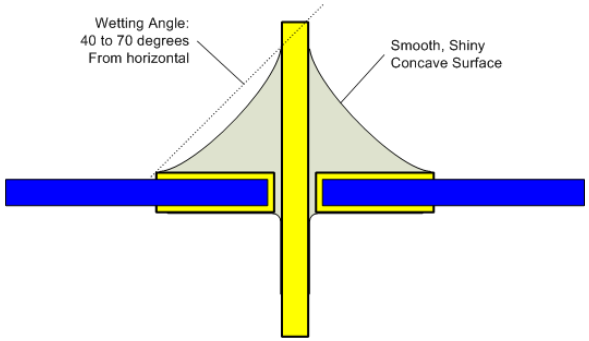
# Tinning

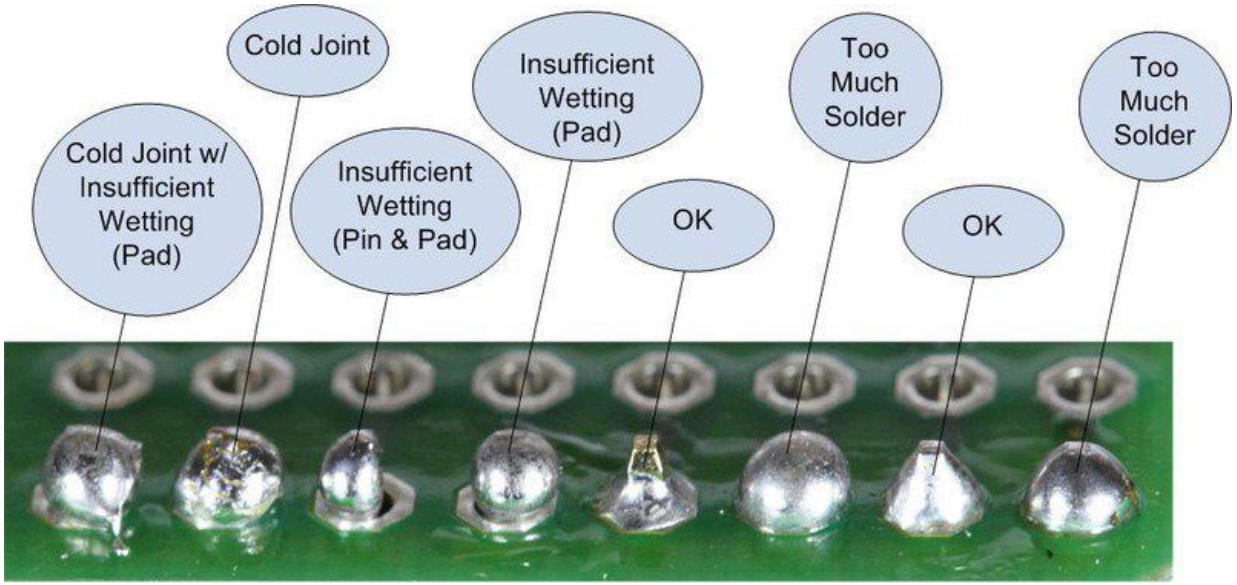
- Oxides on iron prevent adhesion of solder.
- Clean using sponge, scrubber, or tinning block.
- Apply tinning agent or solder.
  - Use solder sparingly when making direct contact with iron. Rosin can etch iron.
  - Should leave surface coated and shiny.
- Monitor: reclean and tin whenever oxides appear.
- Don't do this more frequently than necessary.
- Always do it before starting.

# Soldering

- Deposits on traces and leads can prevent capillary action.
- What soldering **is not!**
  - Melting solder and letting it drop onto traces and leads.
- What soldering **is**!
  - Heating traces and leads, allowing solder to wick into gaps due to surface tension.
- Iron should come into contact with trace and lead first, to preheat them.
- Solder should touch trace/lead junction and flow.
- Can use heatsink on delicate components.

# Soldering


Wetting Angle:
40 to 70 degrees
From horizontal
Smooth, Shiny
Concave Surface


Cold Joint w/
Insufficient
Wetting
(Pad)
Cold Joint
Insufficient
Wetting
(Pin & Pad)
Insufficient
Wetting
(Pad)
OK
Too
Much
Solder
OK
Too
Much
Solder

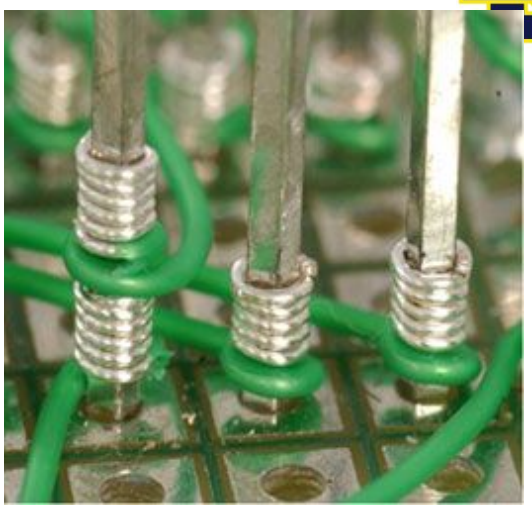
# Goal

Spires of success, not balls of failure.

# Desoldering

- Heat solder in joint and use solder sucker.
- Can clean with wick.
  - Heat the wick when in contact with joint.
  - Don’t rip trace off board.
- This is a pain: desoldering is slow and involved even when you are good at it.

# Wire wrap



- Amazingly fast, once practiced.
- Requires skill.
  - Lost art.
  - Few people know how.
- Reliable.
- Unfortunately, need sockets/adapters for everything.

# Wire wrap

- Measure strip length.
- Strip using slot.
- Insert wire to insulation in off-center hole.
- Put center hole over pin.
- Spin clockwise between fingers with **very gentle** pressure on back of tool.
- To unwrap, use opposite end counter-clockwise.
- Snip, leaving enough for stripping other end.
- Very slow at first.
- After 100, shockingly fast.
- Reliable. Thick.

# Using colors when wiring

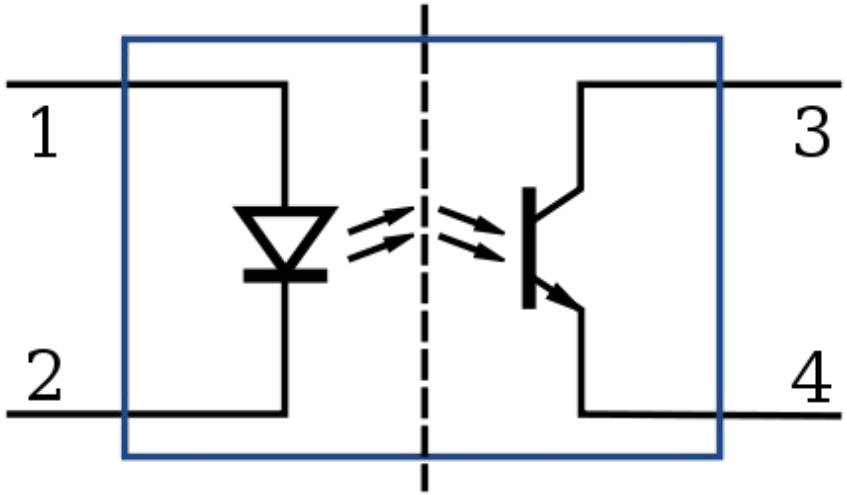
- Black: ground.
- Red: Vdd.
- Many would naturally use one color for addresses and another for data; there is a better way!
- If you only have two other colors to spare, distinguish odd/even, not data/address.
- Now you can trace the wires from pin to pin instead of getting lost in a spaghetti sea.

# Noise sources

- Capacitive coupling.
  - Avoid long, nearby, parallel wires or planes unless you want coupling capacitance.
  - When is coupling capacitance good?
  - Pay attention when low-voltage and high-voltage signals run close to each other.
- RF
  - Inductive coupling / antenna effects.
  - Less local than capacitive. Can be harder to debug.
  - May require shielding.
- High-f noise sometimes easy to filter with ferrite beads: essentially retrofit inductors on wires.

# Noise sources

- Motors.
- Solenoids and mechanical relays are often worse.
- Sparking is a bad sign.
- Common to need to isolate noise source and computer.
- Independent power supplies.
- Opto-isolators.



# Noise rules of thumb

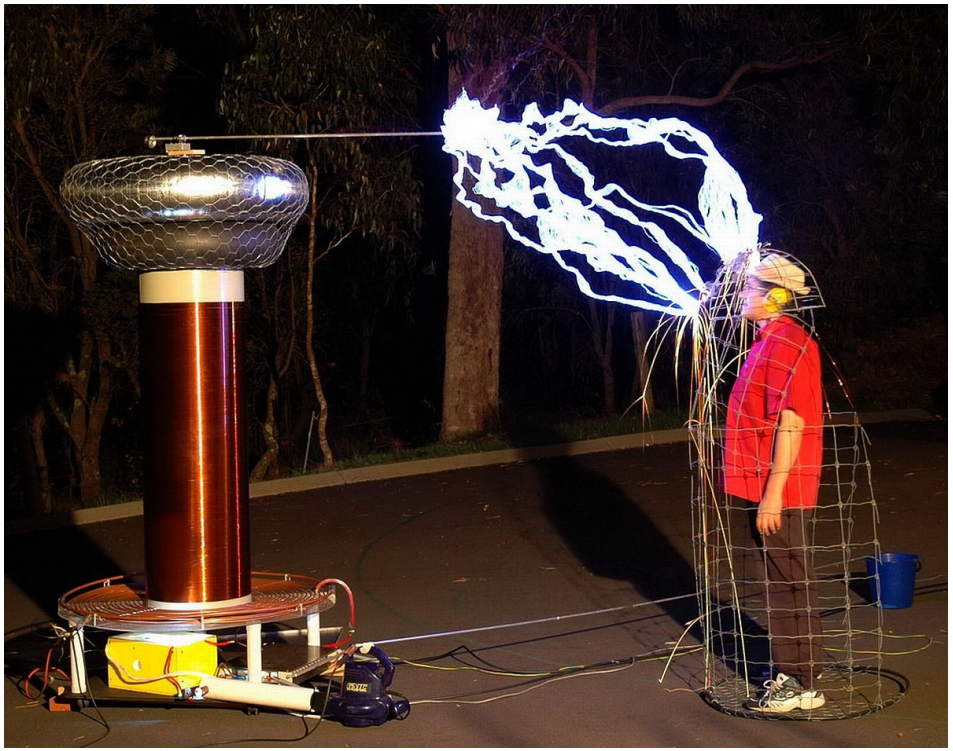
- Really tiny motors.
  - Might even be able to drive from GPIOs, but always check specifications.
- Tiny motors.
  - GPIO to FET.
- Small motors.
  - Independent power supplies, shared ground, FET.
- Big motors, solenoids, and sparky relays.
  - Separate power supplies and grounds, opto-isolators, FET.
- For many applications, H-bridges are easier than bare FETs.
- May need conductive shielding, too.
- Sometimes you can live with noise via clever design.
  - Reboot from safe memory.
  - ECC.

# Faraday cages

- A conductive sphere cancels the effects of external electrical fields.
- Mesh works for most fields we would care about.
- Needs to be highly conductive. Iron doesn't work well. Cu, Al do, but Al hard to connect electrically due to all the sapphire.

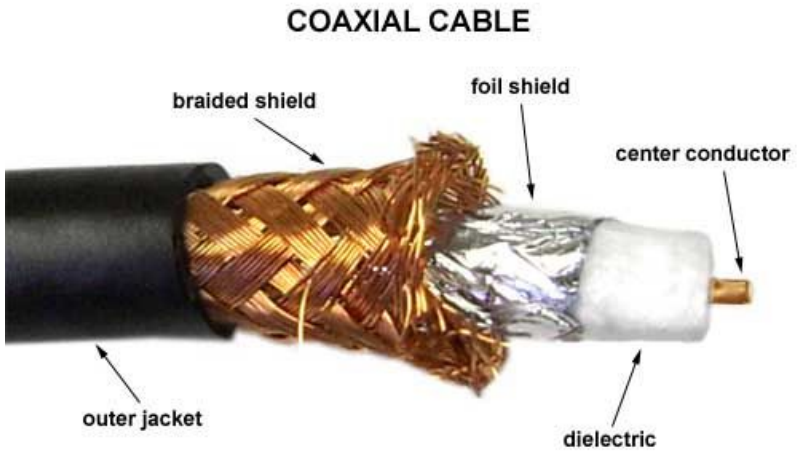
Positive: Safe from electrocution and hearing damage.

Negative: X-rays mutating his DNA. Ozone damaging his lungs.



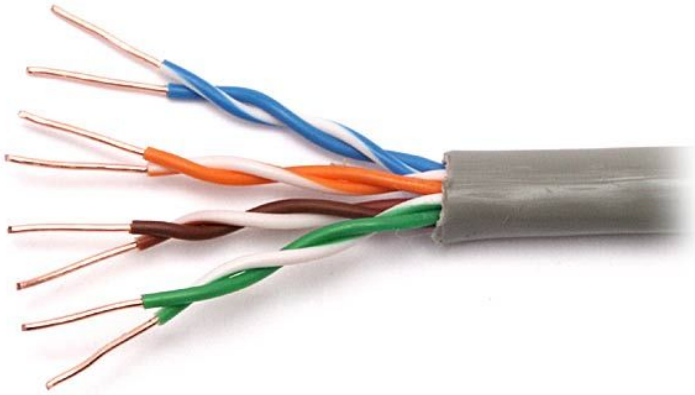
# Coaxial cables

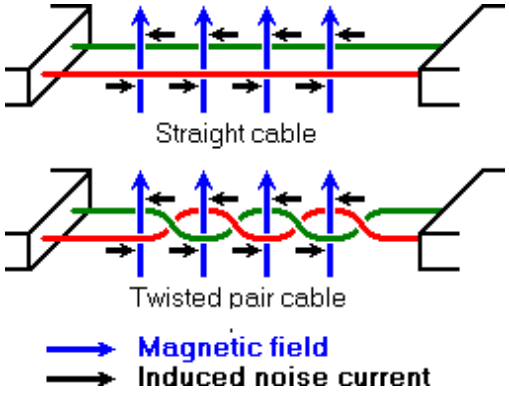
- Very high interference resistance.
- Expensive.



# Twisted pair

- High noise resistance.
- Inexpensive.
- Can roll your own.





# The guild handshake of computer engineers

- Some of you have high potential (7,000 V).
- Others have low potential (0V).
- Your potential changes a lot over time.
- If ESD sensitive component is on path between high-potential and low-potential student, it may be damaged or destroyed.

- Handshaking protocol
  - A holds component away from B in one hand.
  - A reaches out other hand and touches B's hand.
  - A hands over component while other hands still in contact with each other.

- Explain first to non computer engineers.

# Electro static discharge

- High potential difference results in high momentary current through ESD-sensitive structure.
- Examples.
  - Destroy gate oxide.
  - Erase non-volatile memory locations.
- Might cause consistent faults, but might cause rare intermittent faults.
- CMOS generally more susceptible than BJTs.
- Highest risk before PCB mounting.

# Electro static discharge prevention

- Least effective → most effective.
- Handshake, conductive foam/mylar, and nothing else.
- … and touching grounded equipment cases before starting work and periodically.
  - Don't touch while touching live circuit. Dangerous current path.
- … and using grounded mat.
- … and/or wearing grounded wrist strap.
- … and wearing shoe/ankle grounding straps and using grounded floor.
- Danger: Don't short to ground. Use >1 MOhm resistor.

Done.