Lecture 25

Dealing with Noise and Scanning Examples

Agenda

01. Noise and how to deal with it
02. Distance Ranging Scanning
03. Extra: details on I2C master/slave (if we have time)

Requirements Rubric

Up to 3 minutes per task

• Wireless Control Robot	30pt controlled to arbitrary point
• 4 Robot ID UDP transmission	4pt partial credit: # of ID transmitted
• Vive XY UDP transmission	6pt partial credit: XY and datarate
• Follow wall – full circuit	20pt partial credit: % of full circuit
• Autonomous navigation to XY	20pt partial credit getting close
• Beacon tracking – 1 field away	20pt partial credit: % of field
• Total possible	100pts

Design Review in Recitation Friday

- Contents of presentation
  1. Overall concept: abilities you want your robot to achieve
  2. Port allocation (which pins will do what function) and CPU architecture – e.g. a block diagram
  3. General software approach
  4. Mechanical design overview (include any particularly hard issues)
  5. Should have most purchase done by now – include list of purchases in submission (don't need to present this)

Rubric for Design Review

- Primary grade based on teaching staff interpretation of your likelihood of success in the competition.
  - Mostly depends on your progress to date.
  - Also your designs and plan.
- Secondary grade based on your participation
  - Team presentation
  - Suggestions to help other teams
  - Understanding what others are doing
  - Attendance

01

Noise and how to deal with it

What is noise?

- How does noise effect our circuits?
- Has anyone had noise issues ?
- Noise can come into sensor readings, but can they affect digital logic?
- How can you get rid of noise?
- How does noise get in to your circuits??

Standard Debugging process

1. Start with most reliable:
    1. Check power and ground, trace power to and ground to your system.
2. Hardware components
    1. Check each hardware component output individually.
    2. Check inputs have proper effect (manually control input lines).
3. Software components
    1. Write short routines that you know work to check individual functions
4. Combine individual components one at a time until you isolate failure.
5. When everything is working separately except sometimes when assembled, **suspect noise as the problem**

Order of trust (correct and working properly)

Finding what is working

1. Corporate documentation
2. Test equipment:
   - E.g. DMM, Power Supply, Oscilloscope
3. Passive parts
   - Resistors, caps, wires
4. Simple active parts
   - LEDs, diodes, voltage regulators
5. Oscillosorta
6. Lecture documentation
7. Web documentation

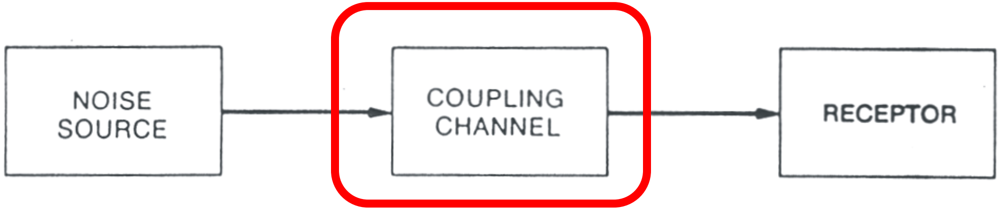
90% of ERRORS DUE TO THESE:

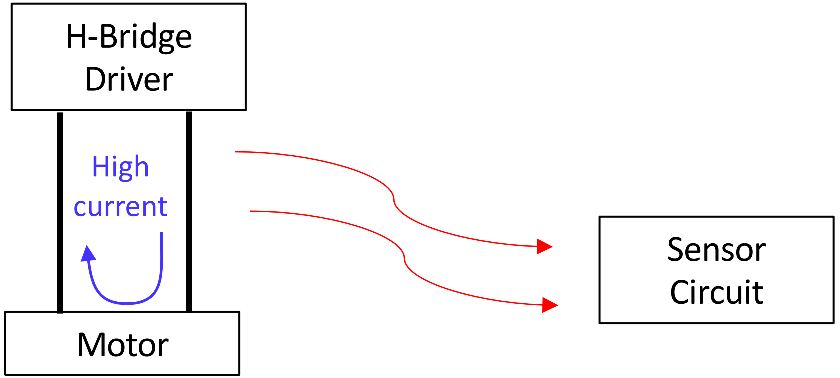
12. Blown port on MCU
13. External hardware
    - H-bridge, logic, opamps, sensor, motor, driver
14. Wiring
    - breadboard interface, connectors,
15. Software

Finding what is broken

Intermittent failure. It's either interrupts or NOISE…

How noise gets into your circuits





<u>Four types of coupling</u>

1. Conductive coupling
2. Capacitive coupling
3. Inductive coupling
4. Radiative coupling

Characteristics of the noise sources

- Distance from the victim

Most common, easiest to fix

Distance = 0 → Direct contact → Conductive coupling

RF noise

Distance > wavelength → Probably Radiative 80Mhz = ~4m

0 < Distance < Wavelength → Capacitive or Inductive

- Voltage noise source

Any two conductors with shared area

Strong electric field → Capacitive coupling

Examples?

Parallel wires

'High' dV/dt → Conductive and Capacitive coupling

Digital signals!

- Current noise source

'High' current → Strong magnetic field → Inductive coupling

Examples?

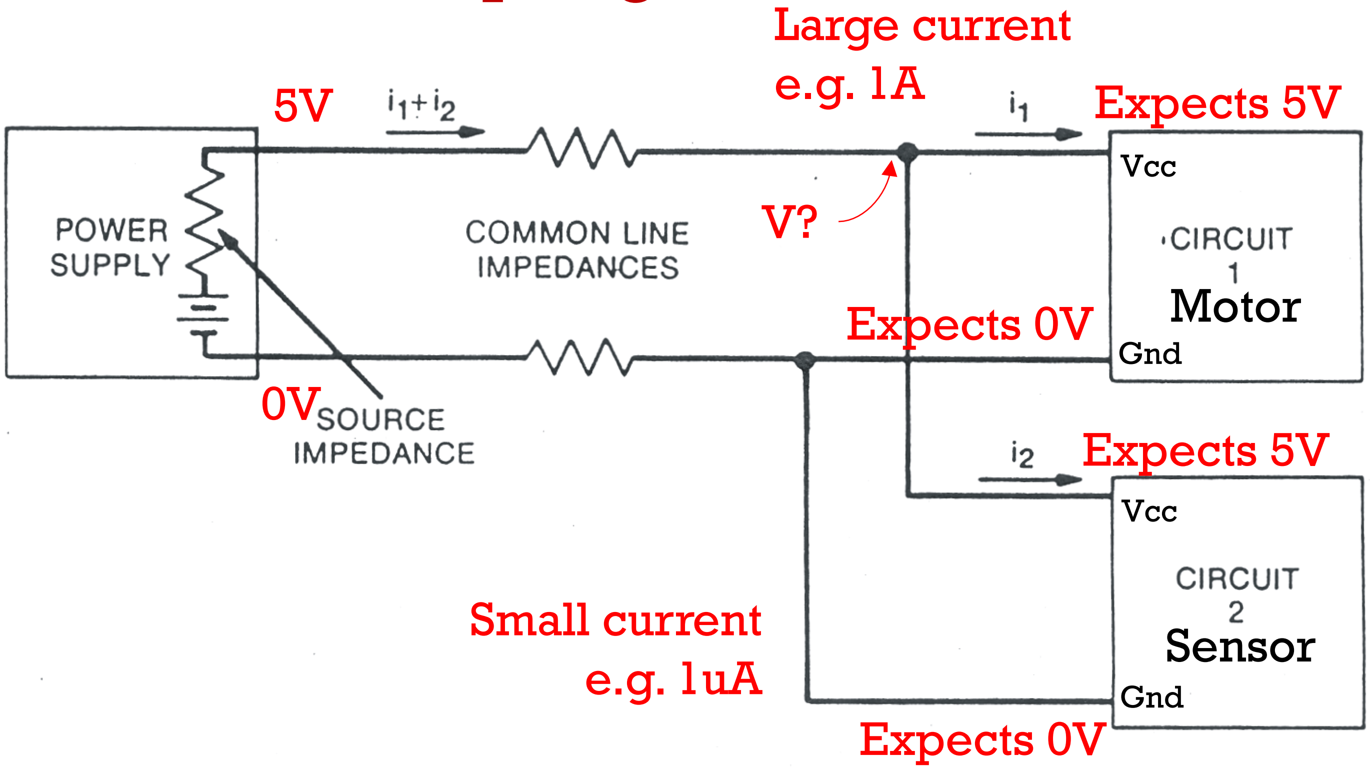
$V = L\, dI/dT$

Motors

- Frequency noise source

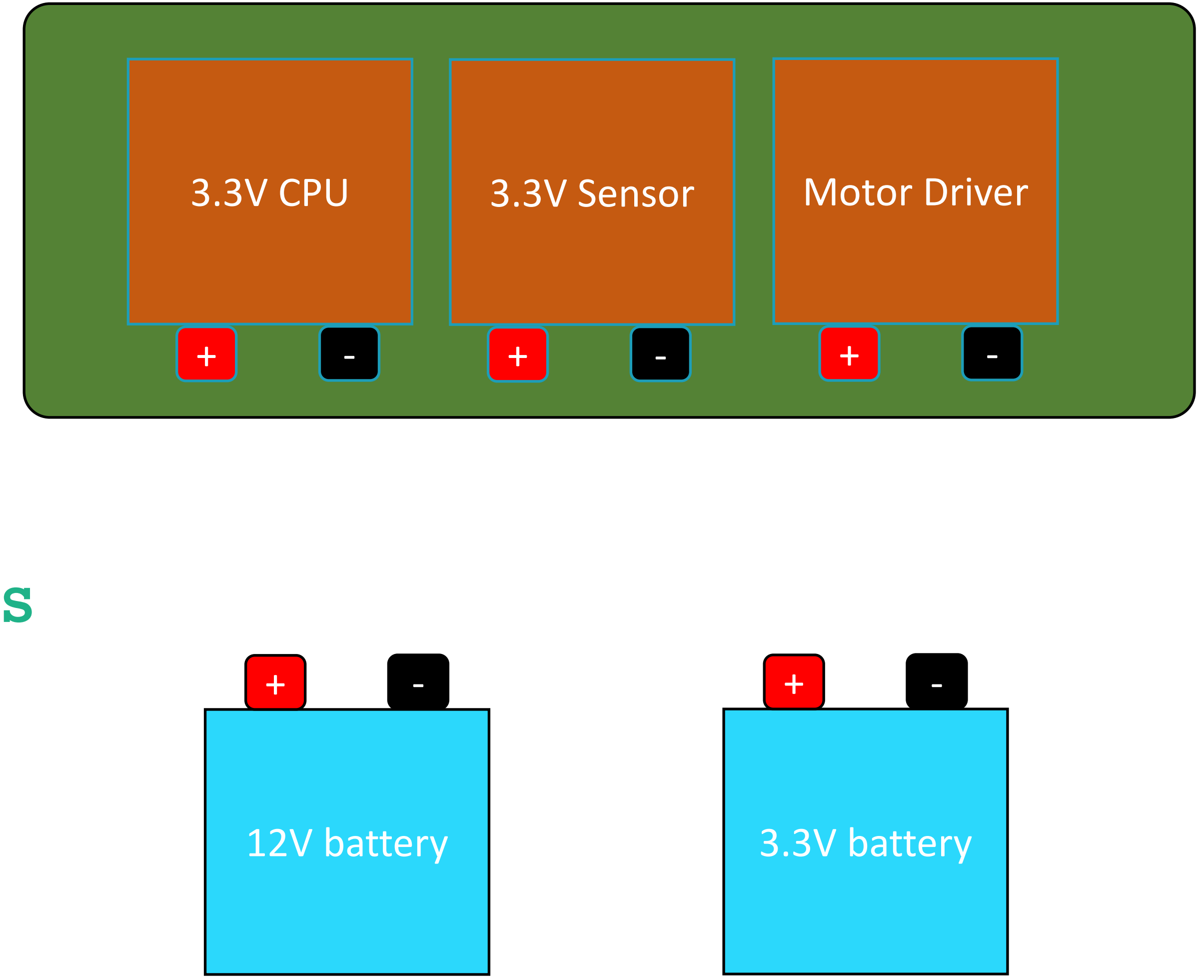
'High' frequency → Radiation → Radiative and inductive coupling

Conductive Coupling

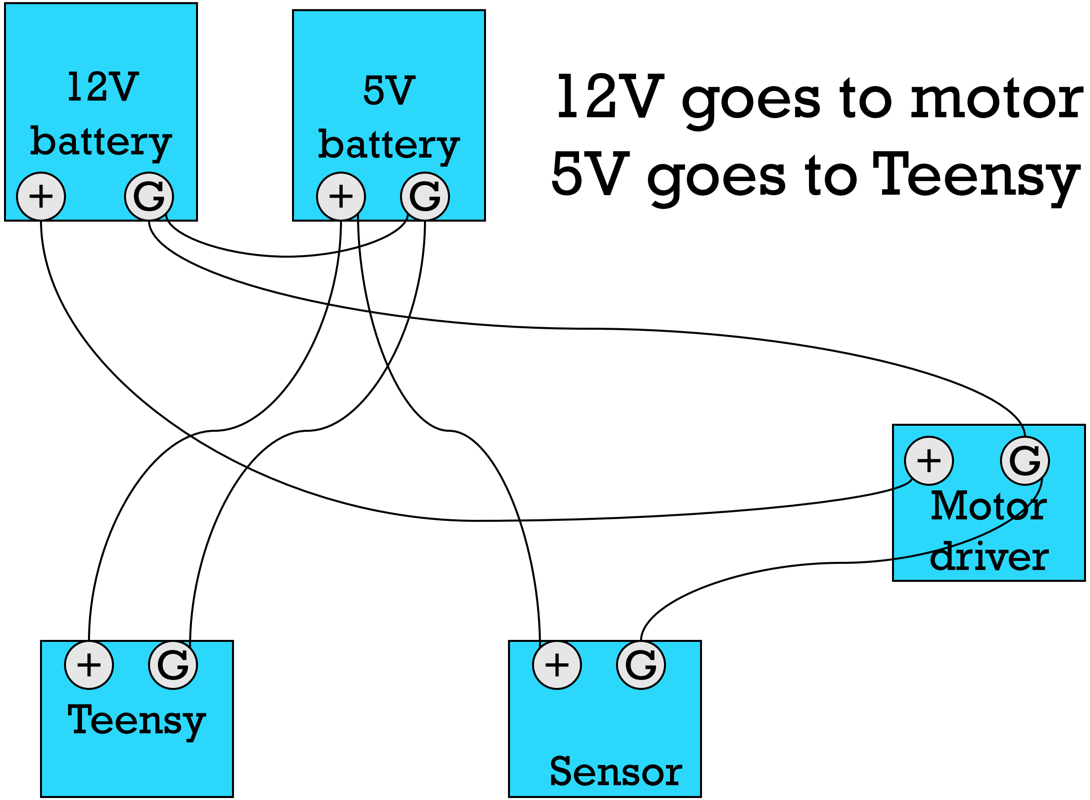


Q1: If cabling has 0.5Ω resistance, what voltage will the sensor see at its Vcc?

Q2: Draw power and ground sources to the devices. Considering wires have non-zero resistance



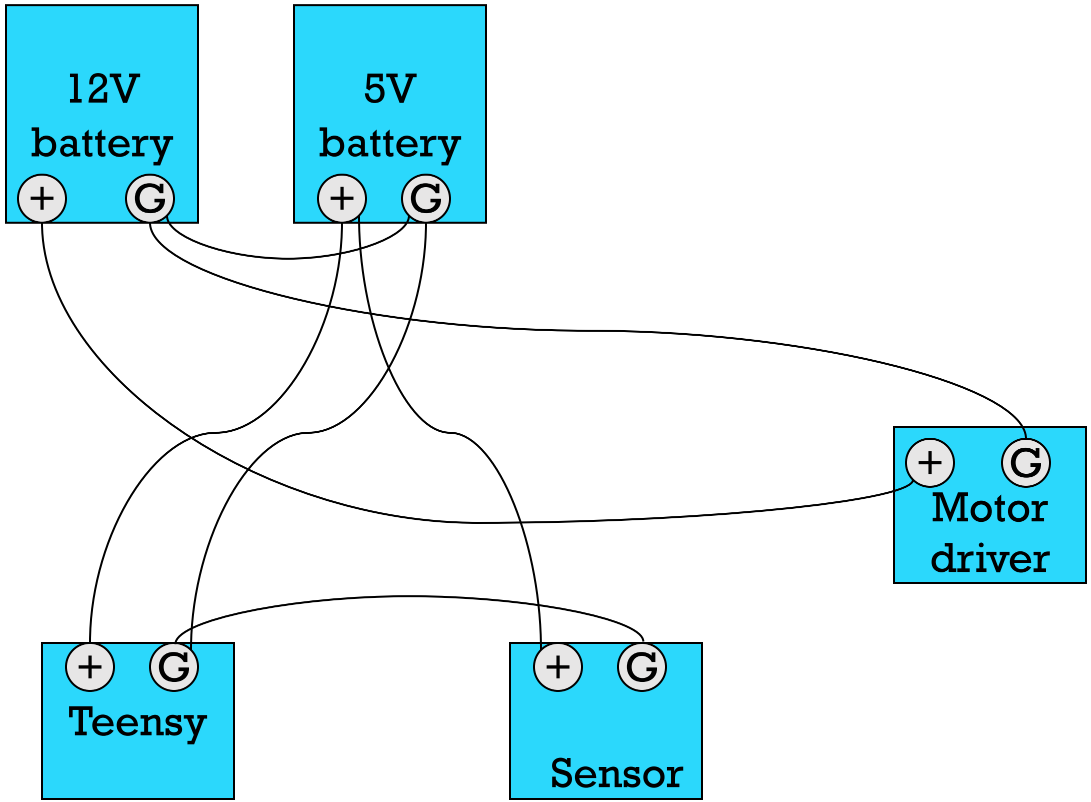
Q3: How could we make this better?



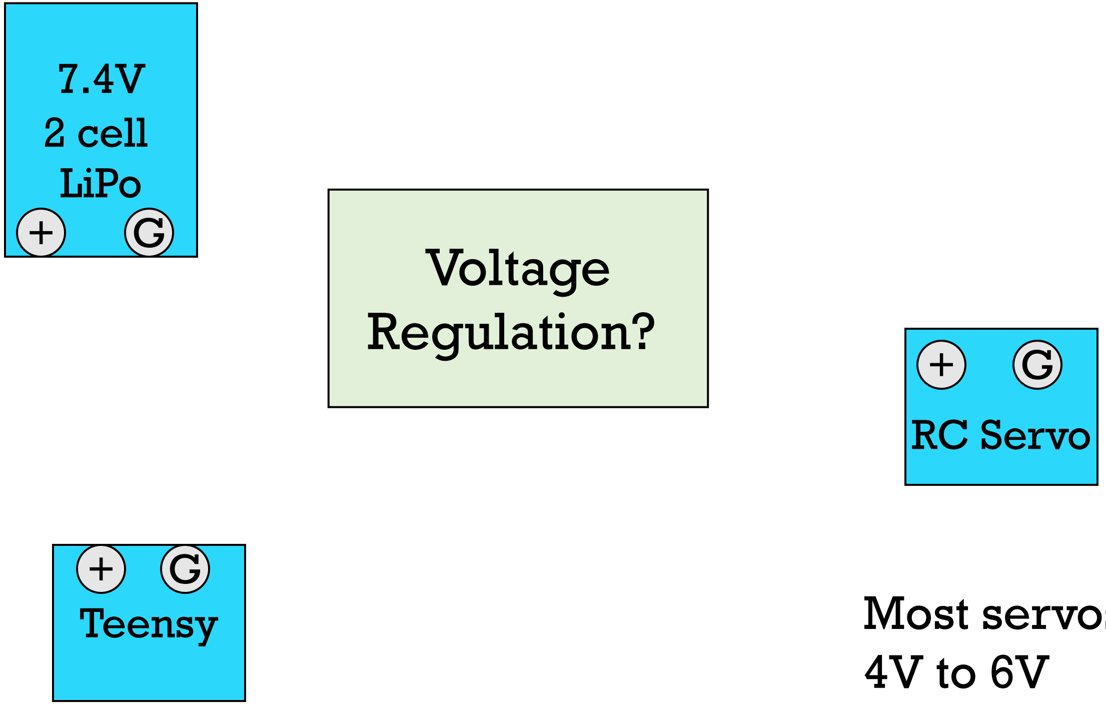
12V goes to motor driver

5V goes to Teensy and sensor

How about this?



If we use a 2S1P two cell 7.4V LiPo, how do we get 5V?



Requires 5V @ ~200mA max
No onboard regulator

Most servos will take
4V to 6V
Max current varies
Often 1A – 3A

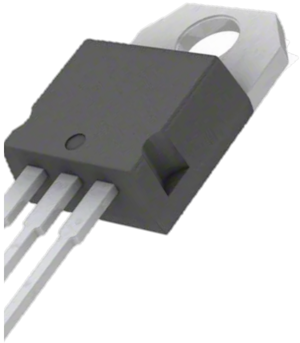
LM7805 Voltage Regulator
Will this work?

6.6 LM340 / LM7805 Electrical Characteristics, V_O = 5 V, V_I = 10 V

0°C ≤ T_J ≤ 125°C unless otherwise specified(1)

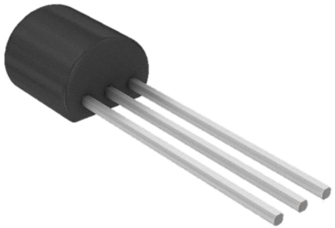
PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_O	Output voltage	T_J = 25°C, 5 mA ≤ I_O ≤ 1 A		4.8	5	5.2	V
		P_D ≤ 15 W, 5 mA ≤ I_O ≤ 1 A 7.5 V ≤ V_{IN} ≤ 20 V		4.75		5.25	V
ΔV_O	Line regulation	I_O = 500 mA	T_J = 25°C 7V ≤ V_{IN} ≤ 25V		3	50	mV
			Over temperature 8V ≤ V_{IN} ≤ 20V			50	mV
		I_O ≤ 1 A	T_J = 25°C 7.5V ≤ V_{IN} ≤ 20V			50	mV
			Over temperature 8V ≤ V_{IN} ≤ 12V			25	mV
ΔV_O	Load regulation	T_J = 25°C	5 mA ≤ I_O ≤ 1.5 A		10	50	mV
			250 mA ≤ I_O ≤ 750 mA			25	mV
		Over temperature, 5 mA ≤ I_O ≤ 1 A				50	mV
R_O	Dropout voltage	T_J = 25°C, I_O = 1 A			2		V
	Output resistance	f = 1 kHz			8		mΩ
	Short-circuit current	T_J = 25°C			2.1		A
	Peak output current	T_J = 25°C			2.4		A
	Average TC of V_{OUT}	Over temperature, I_O = 5 mA			−0.6		mV/°C
V_{IN}	Input voltage required to maintain line regulation	T_J = 25°C, I_O ≤ 1 A		7.5			V

(1) All characteristics are measured with a 0.22-µF capacitor from input to ground and a 0.1-µF capacitor from output to ground. All characteristics except noise voltage and ripple rejection ratio are measured using pulse techniques (t_w ≤ 10 ms, duty cycle ≤ 5%). Output voltage changes due to changes in internal temperature must be taken into account separately.

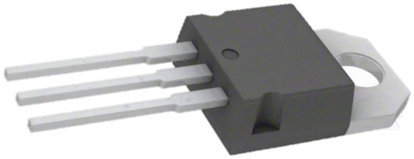


Other GMLab options for generating ~5V

- MCP1702 LDO regulator
  - Vin_{max}=13.2V, Vin_{min} =Vout + 0.62V
  - Vout = 5V (also have 3.3V version)
  - 250mA max output



- LM317 Adjustable Regulator,
  - Vin_{min} = Vout + 3V
  - 1.5A max output

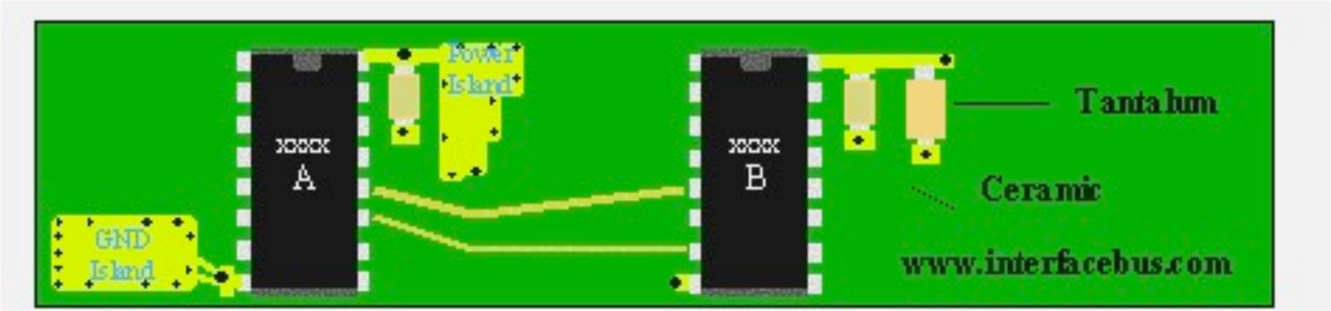


- TCA0372 Dual Power Op-Amp,
  - Vin_{min} ~= Vout + 1V
  - Won't be fixed Vout
  - 1A max output

- DF005M Bridge Rectifier, 3W max
  - Vf = ~ 1V (not regulating, but will lower V, like opamp)

Digital Circuitry Noise

- All digital circuits add noise potentially on the power lines.
- Add decoupling capacitors as standard practice
  - 10nF monolithic in parallel with 10uF tantalum or electrolytic cap at power rails
  - 10nF monolithic at every digital chip as close as possible to power-gnd

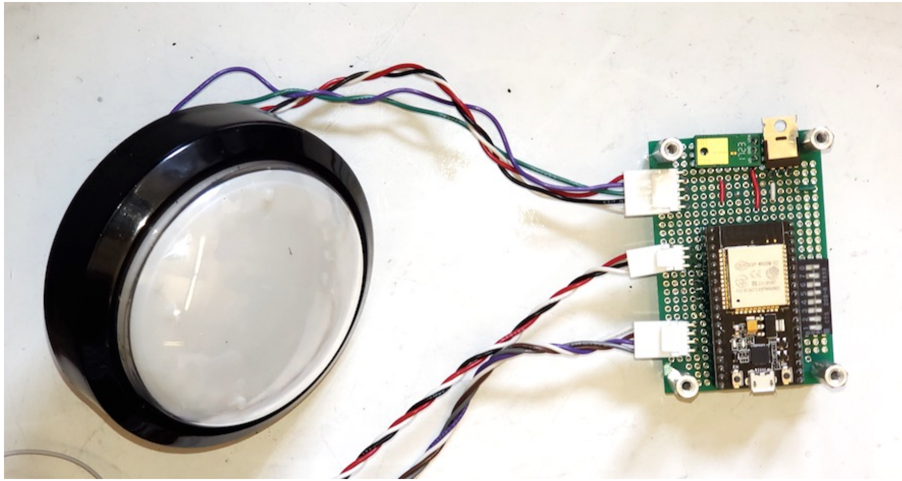


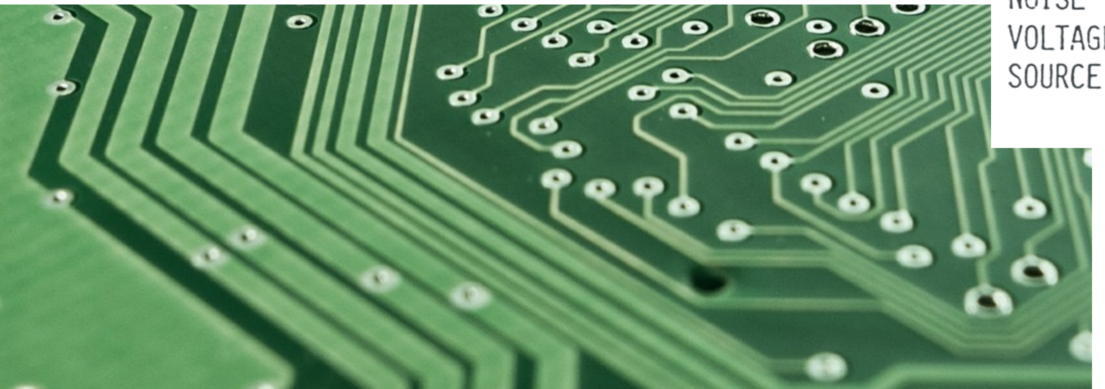
Reducing conductive coupling

- Change the conductive paths
- Power and ground are common paths for noise.
- Reduce common impedance (star configuration connections all join at battery).
- Use separate power supplies (noisy vs clean) if possible.
- Use lower resistance wires (fatter) for high currents.

Low freq noise is hard to get rid of

Capacitively coupled noise

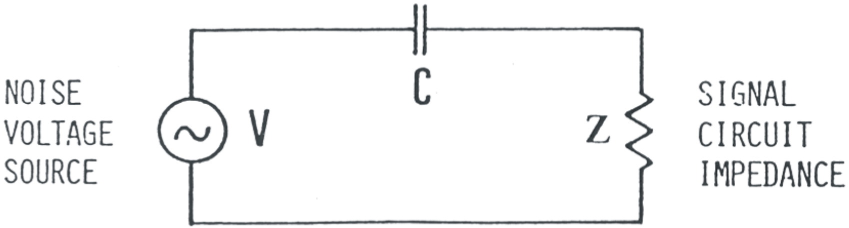




Simplified circuit representing all types of capacitive coupling

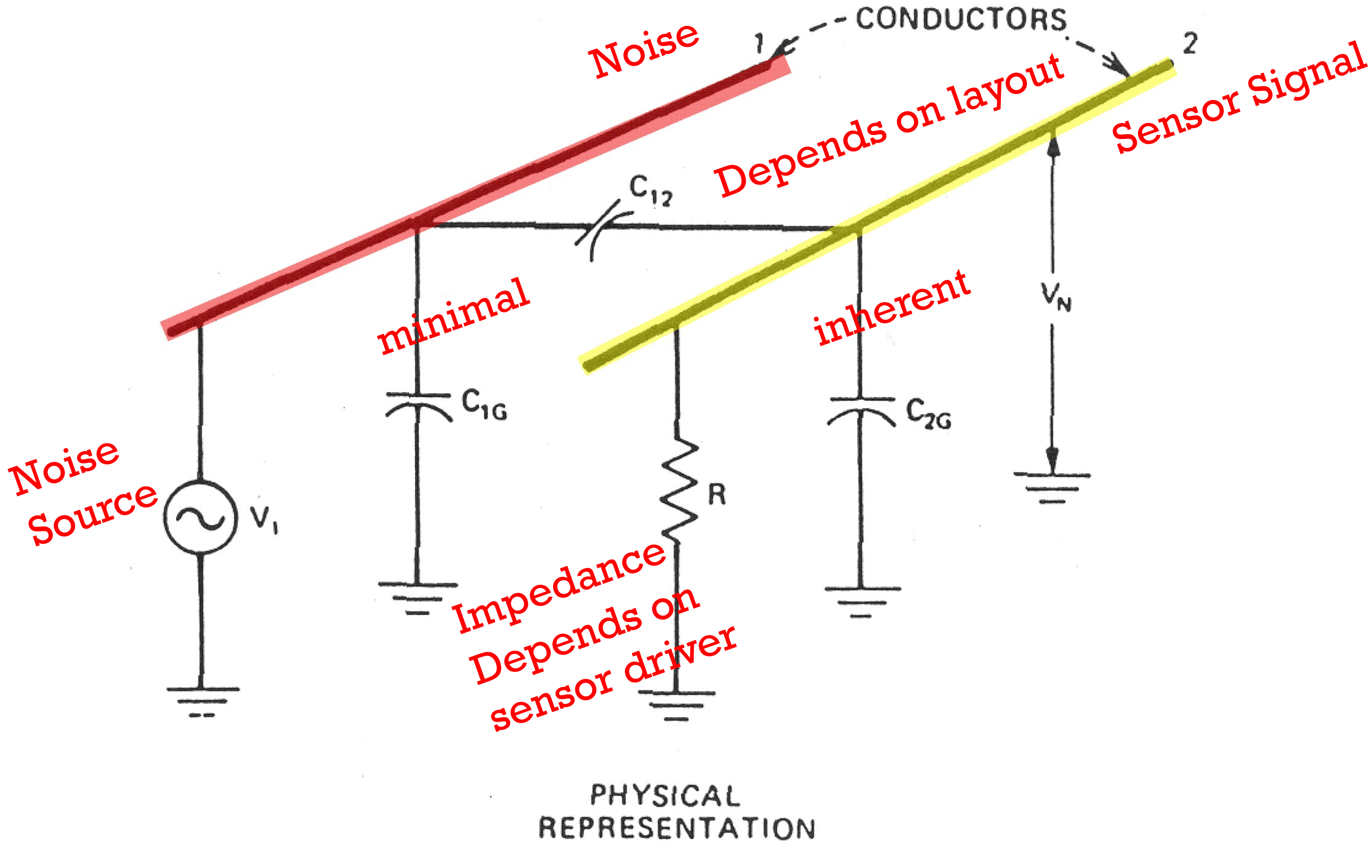
Coupling capacitance

Shared area



Typical example is leads next to each other, or long parallel cables

Physical Representation of Capacitively Coupled Noise



Equivalent circuit for capacitively coupled noise

We want noise to be small, V_N -> 0

$$V_N = \frac{j\omega[C_{12}/(C_{12}+C_{2G})]}{j\omega + 1/R(C_{12}+C_{2G})} V_1$$

If $R >> \frac{1}{j\omega(C_{12}+C_{2G})}$

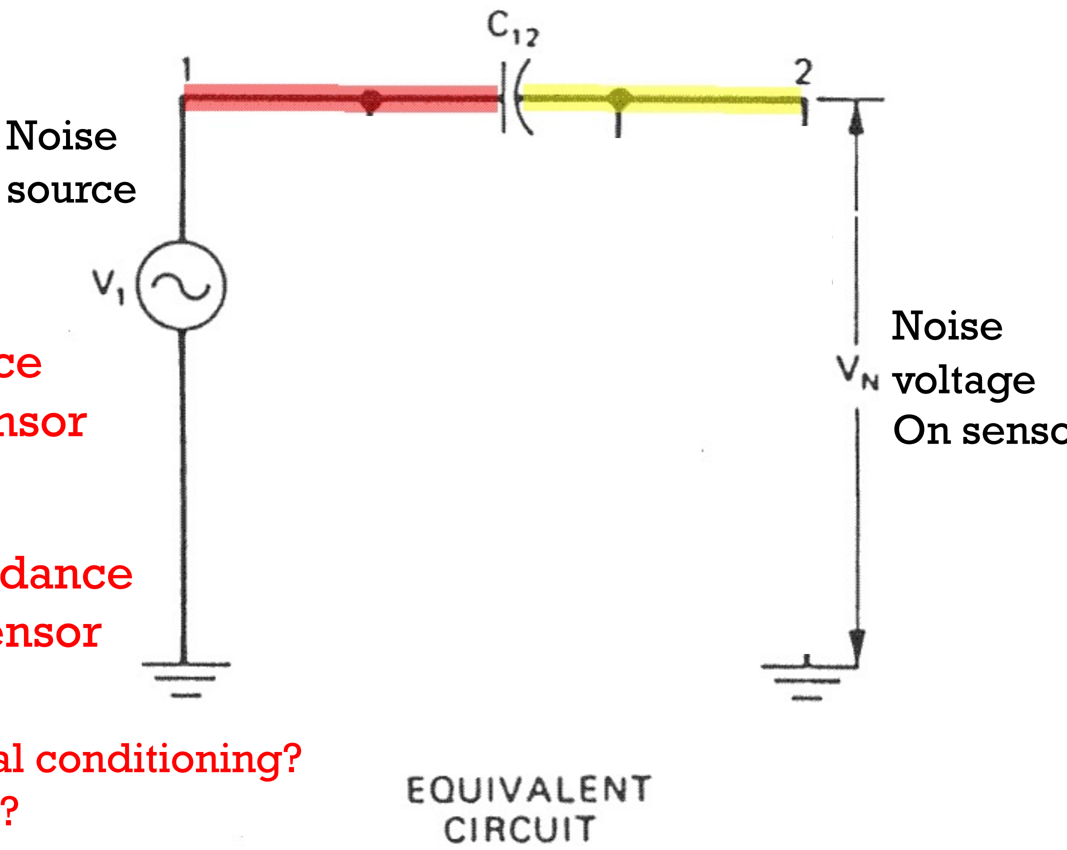
$$V_N = \frac{C_{12}}{(C_{12}+C_{2G})} V_1$$

High impedance (weak sensor driver)

If $R << \frac{1}{j\omega(C_{12}+C_{2G})}$

$$V_N = j\omega R C_{12} V_1$$

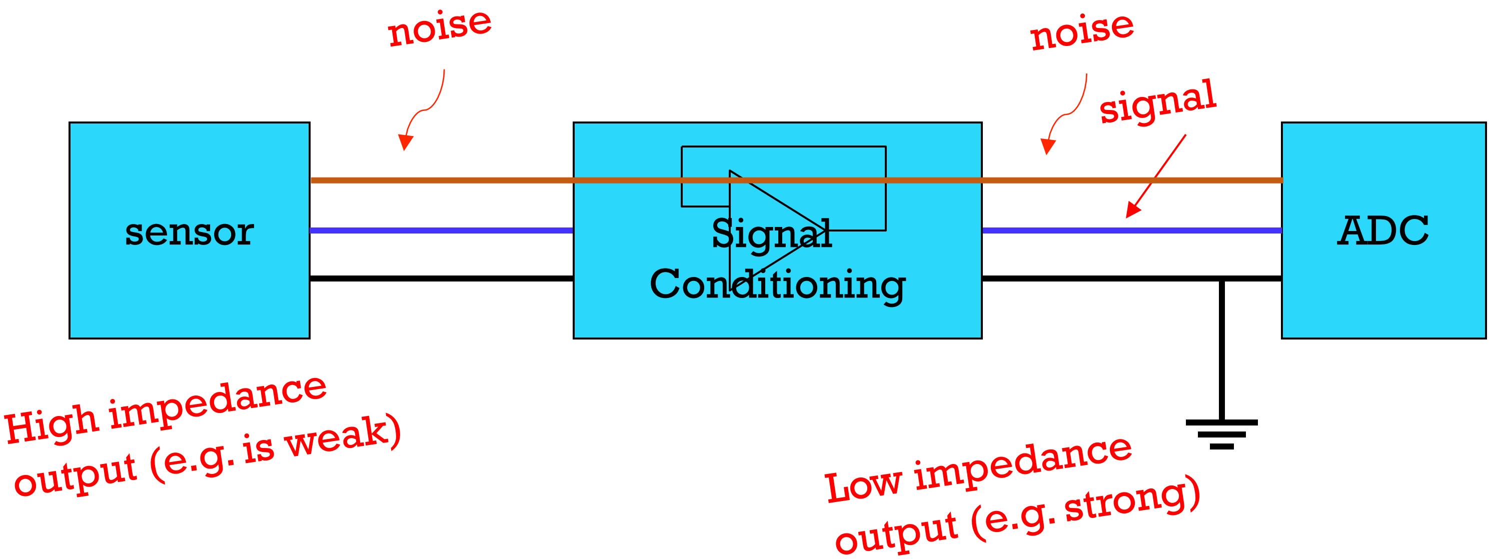
Low impedance (strong sensor driver



How should you hook up your signal conditioning?

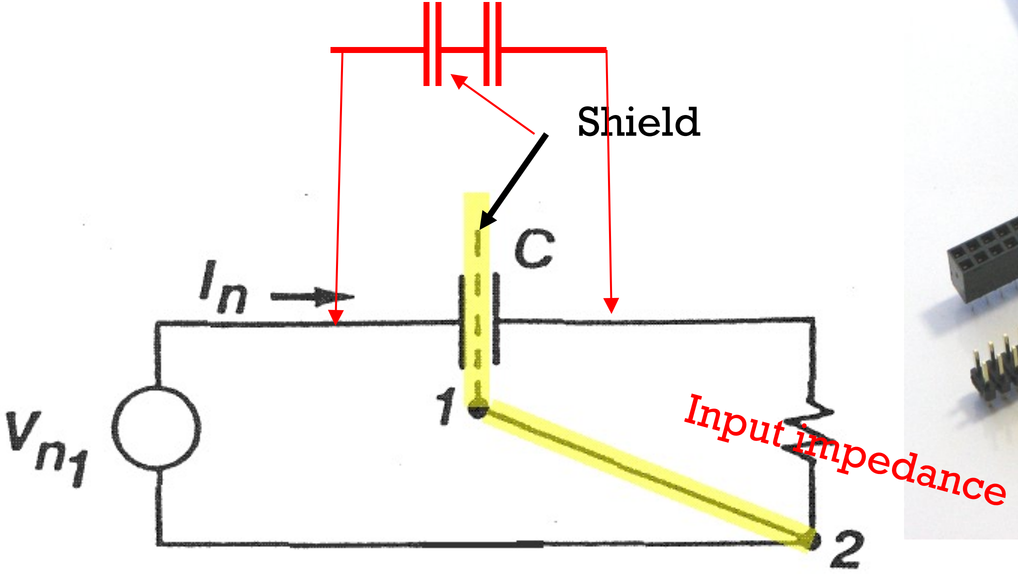
How should you hook up your ADC?

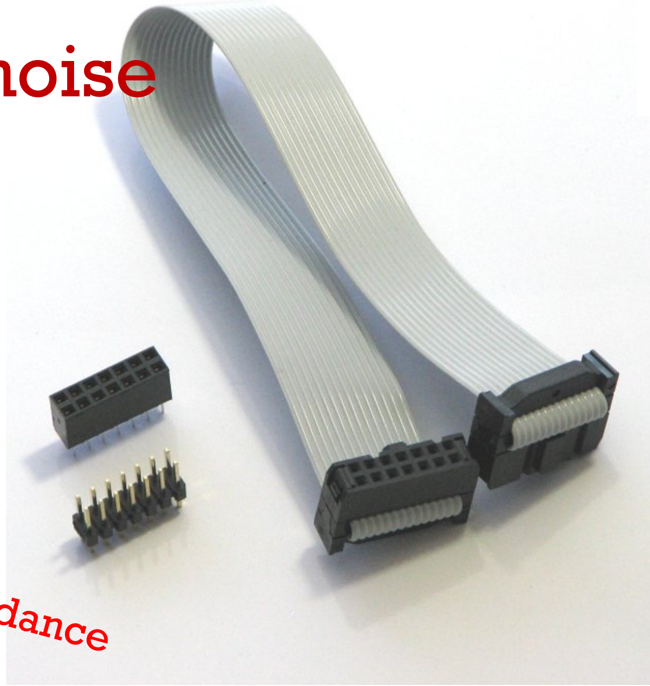
Sensor wiring



Higher impedance is more sensitive to noise.
Reduce high impedance coupling at the cost of increased low impedance coupling.
(e.g. op amp output driving larger current)

Reducing Capacitively coupled noise





Position the shield to intercept the noise current

Connect the shield to return the noise current to the source

Summary of capacitive noise reduction techniques

Reduce capacitive noise coupling by

1. Reducing coupling capacitance
2. Reducing circuit impedance
3. Using shielding

e.g. reduce parallel lines

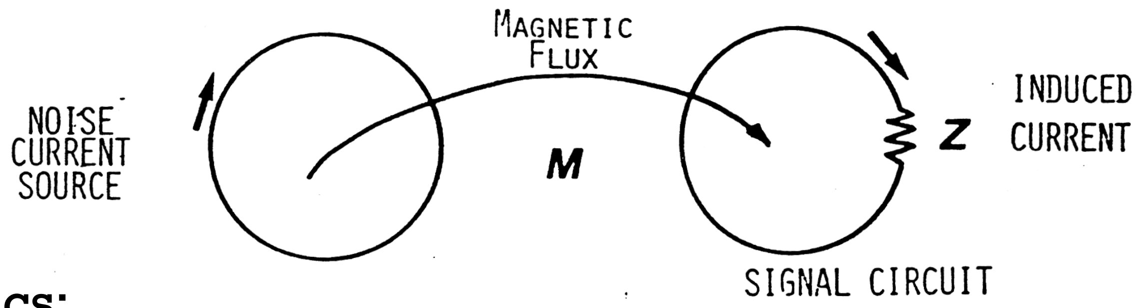
e.g. increase output drive w opamp

Capacitive shielding requires

1. Proper shield location
2. Correct shield connection

A little tricky when we consider inductive noise

Inductive Noise Coupling



Characteristics:

- Large loop areas in wiring
- High noise current or frequency
- Unaffected by non-conducting, non-magnetic materials
- Shield effectiveness not helped by grounding
- Detectable by magnetic (loop) sensor

In our robots where do we have large loops of coils with large currents going through tme?

Reducing Inductive Noise Coupling

Reduce the current in the noise source

Reduce the frequency content in the noise source

Reduce the mutual inductance

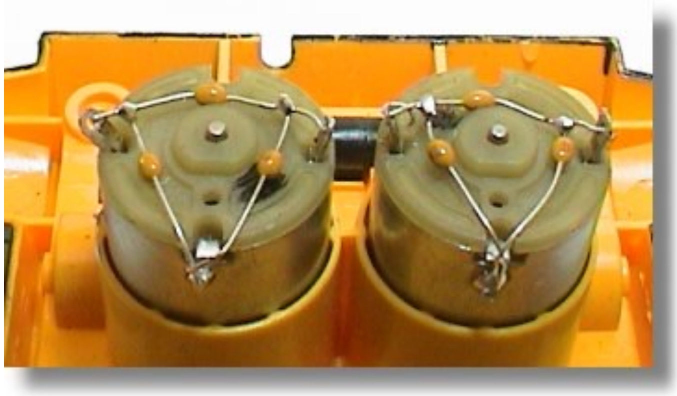
e.g. makes loops smaller, or
Increase distance from source

Use magnetic shielding

e.g. iron-based metal foil

Filter the noise at the signal circuit

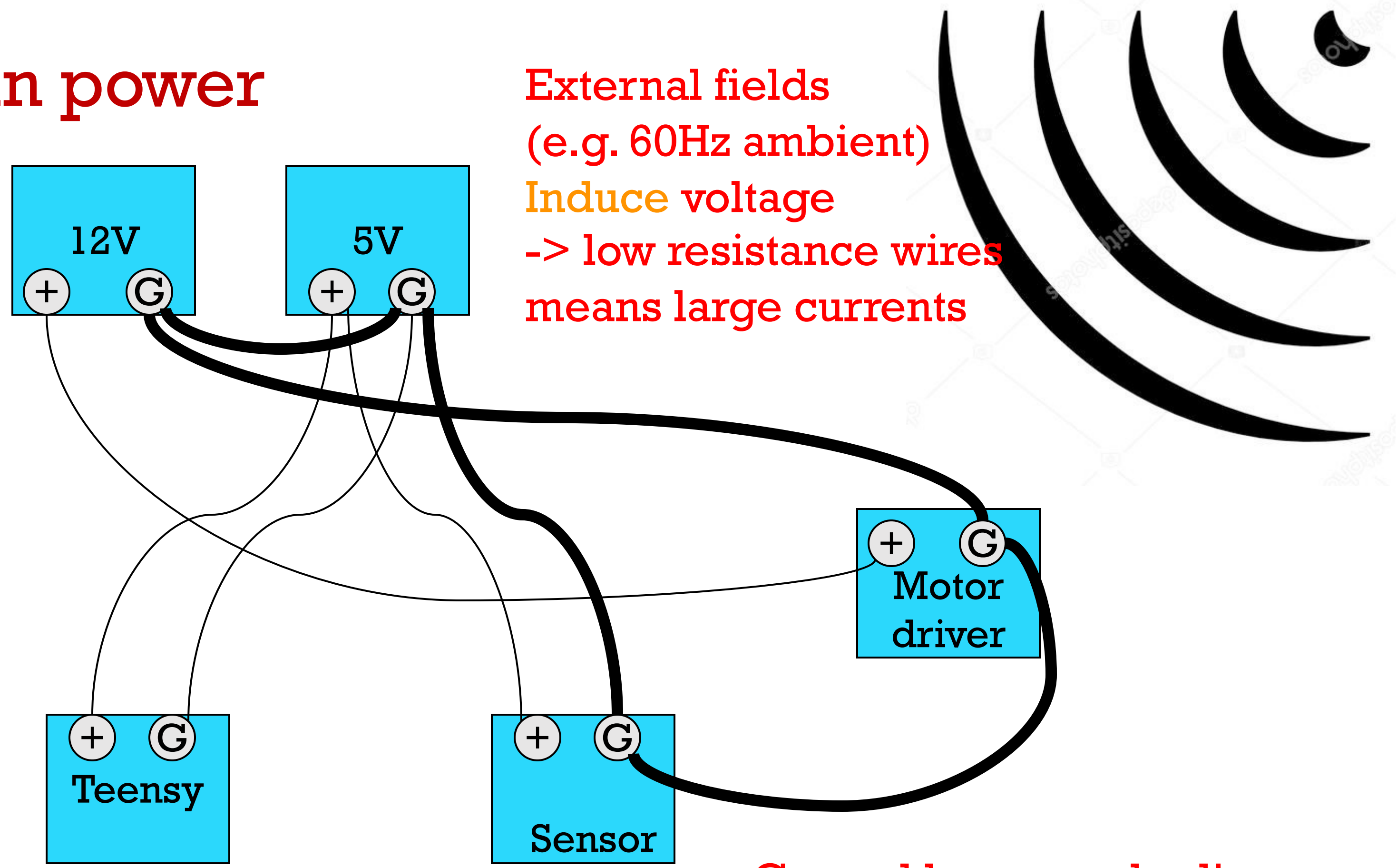
For motors, typically solder capacitor
across leads and to motor casing.



Choose monolithic caps rated
at least 3x's motor voltage

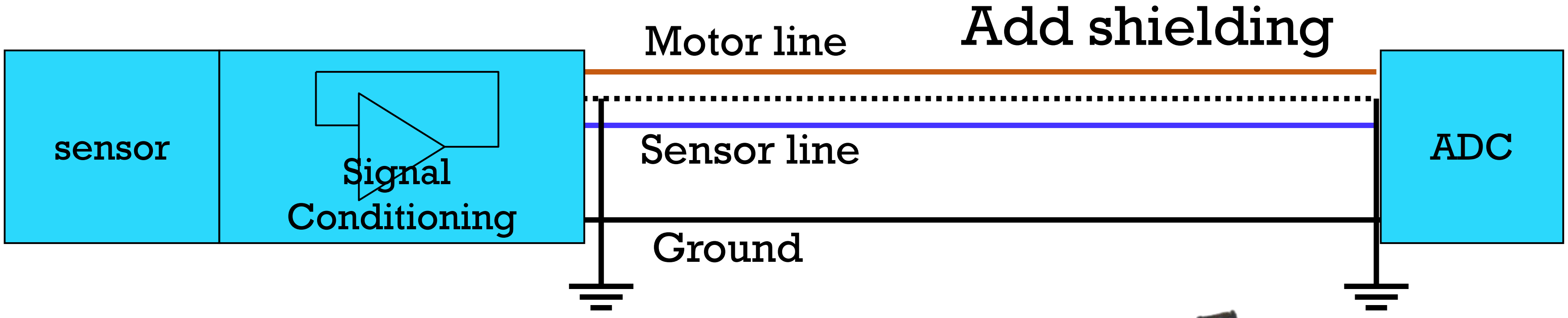
Loops in power

External fields
(e.g. 60Hz ambient)
Induce voltage
-> low resistance wires
means large currents



Ground loops are bad!

Q4: What kind of potential noise could we have and where?



Q5: How about now?

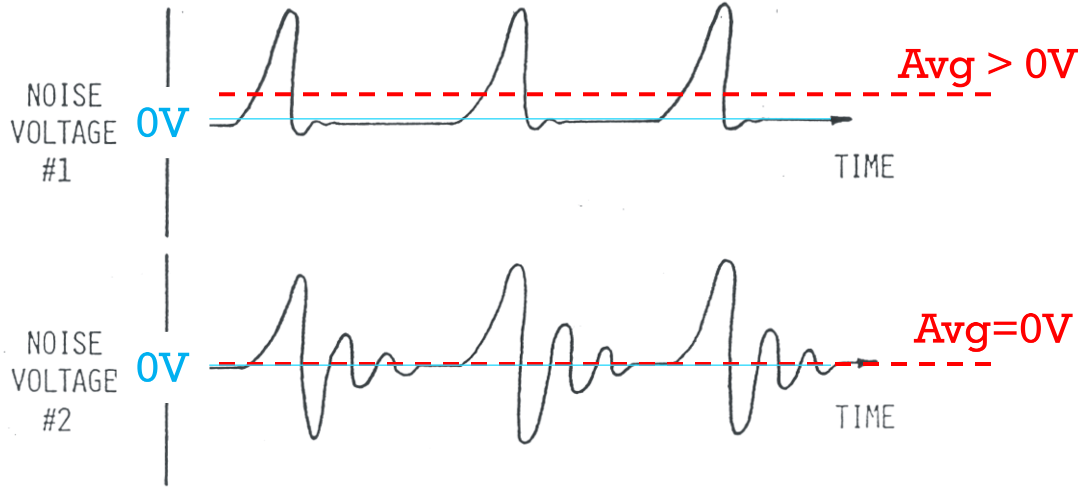
Ground loops occur if you ground both ends of your shield

Q6: Which waveform must have noise from a conductively coupled source?

A

Pure signal should be 0V (straight line)

B

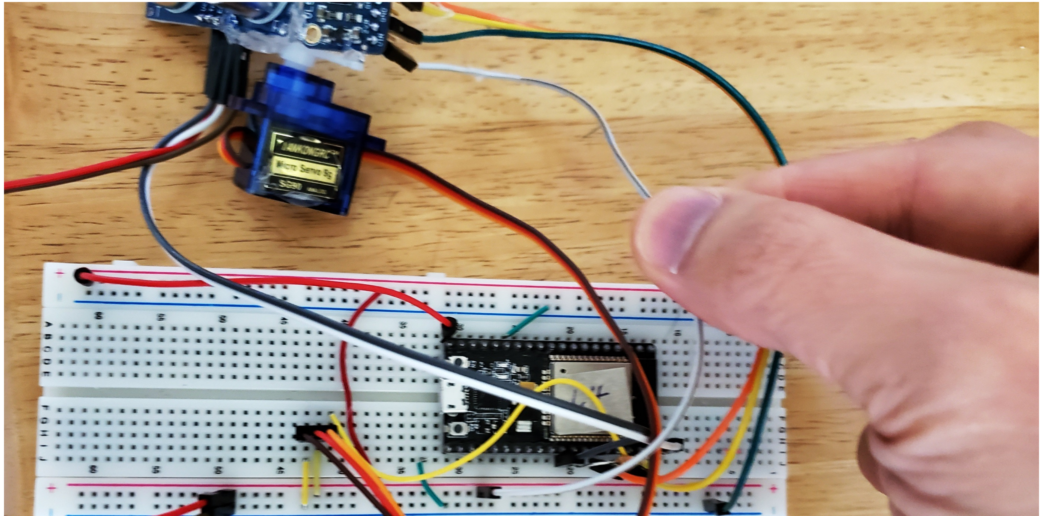


Why?

Identifying which type of coupling

- Non-zero average value for the waveform -> Conductive
- Conductive contact required -> Conductive
- Affected by people or cable movement -> Capacitive or Inductive

Wiggling wires shows some change in noise (not loose contact)



Isolating cross talk for ToF

- LED transmits at 10Mhz
- Summing phase shift gives time of flight for reflection
- Need to isolate photodiode from LED.

4 strategies to reduce crosstalk from LED to photodiode:

1. Separate ground planes surround each device
2. LED and photodiode leads are mounted perpendicular
3. Traces to diodes minimize loop area
4. Vias form vertical "picket fence"

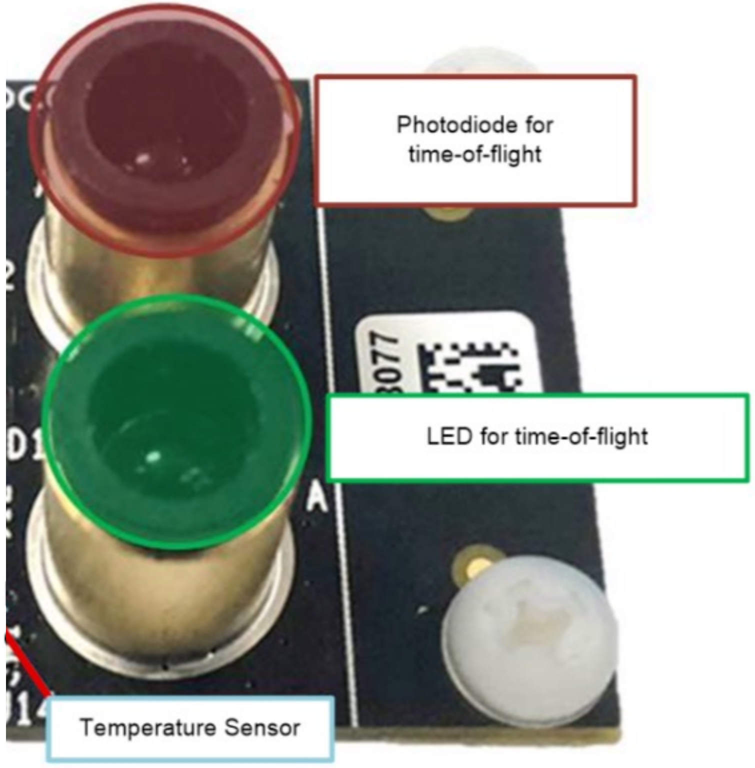
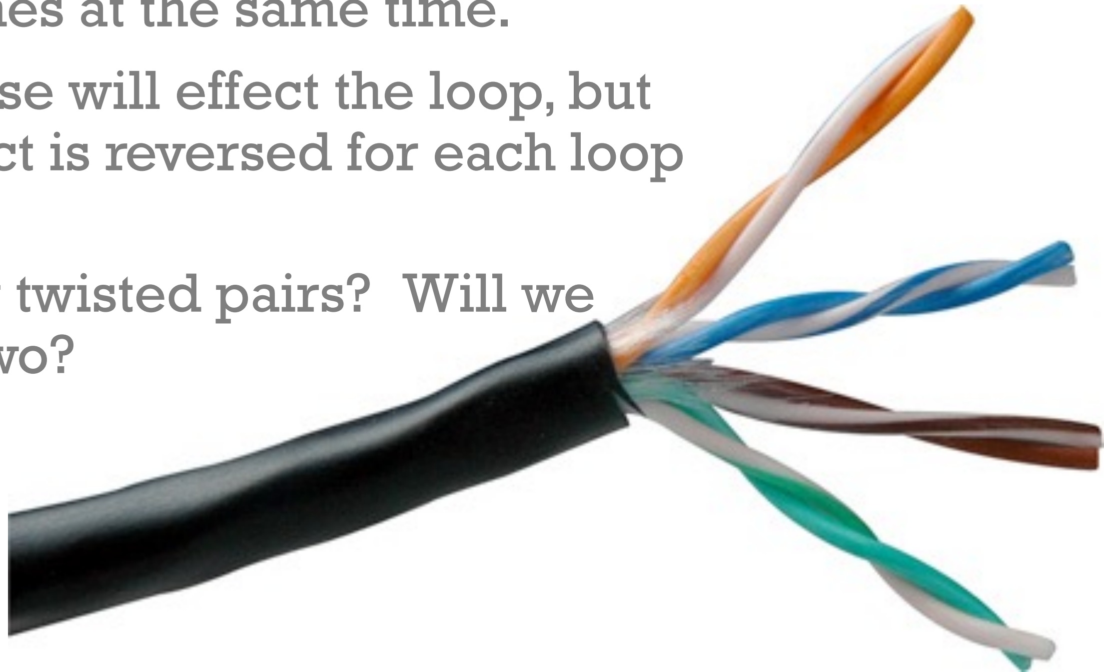


Figure 179. PCB Layout example

Cables often come in twisted pairs – Why? Especially considering capacitively coupled noise

- Twisted pairs are intended to be used differentially. Both lines hold one signal (so coupling is a good thing). Noise effects both lines at the same time.
- External fields inducing noise will effect the loop, but the twisting means that effect is reversed for each loop canceling out each other.
- But what about neighboring twisted pairs? Will we get crosstalk between the two?



Noise Summary

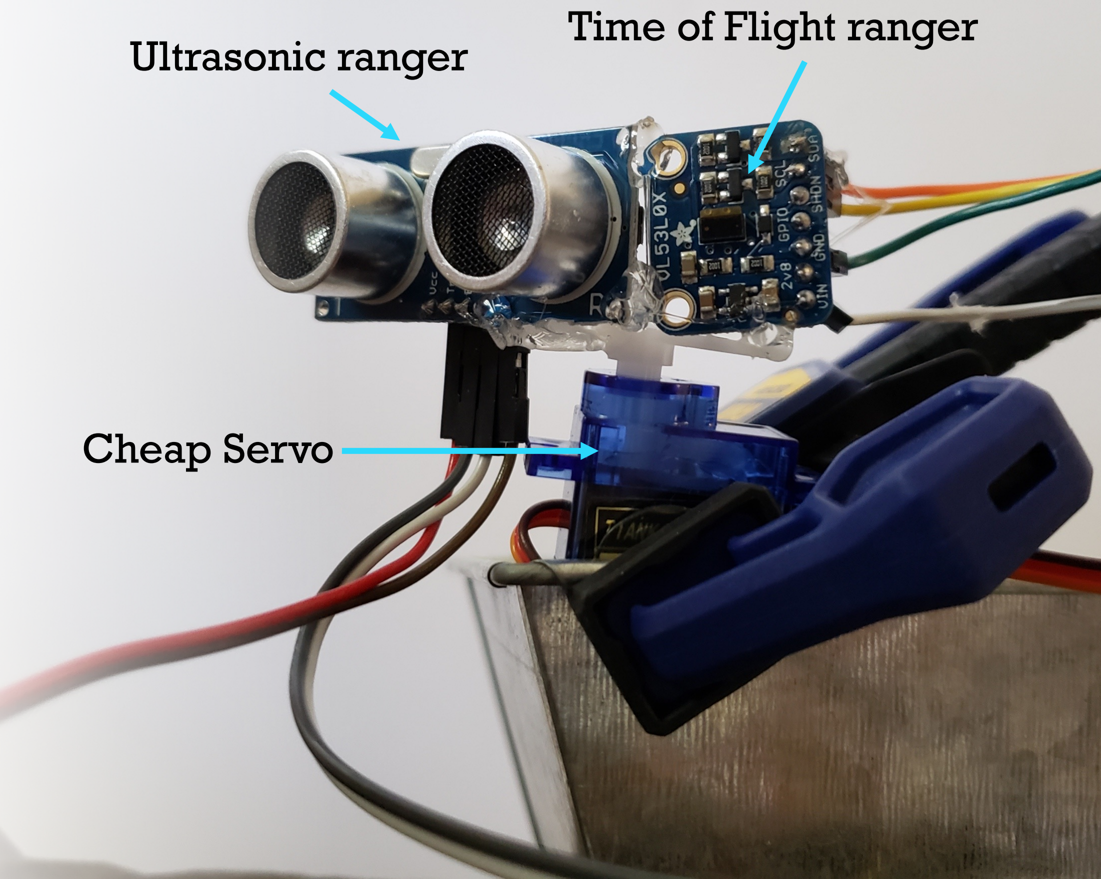
1. If needed, use two power supplies if you have noisy (e.g. motor circuits) and separate controllers
2. Add caps to motor leads
3. Add decoupling caps
   A. each digital chips (.1uF monolithic)
   B. .1uF monolithic and ~10uF electrolytic @ power to board
4. Keep sensitive lines from high freq or high current lines (if must be close, cross at right angles)
5. Shield as necessary, attach only at one end!

02

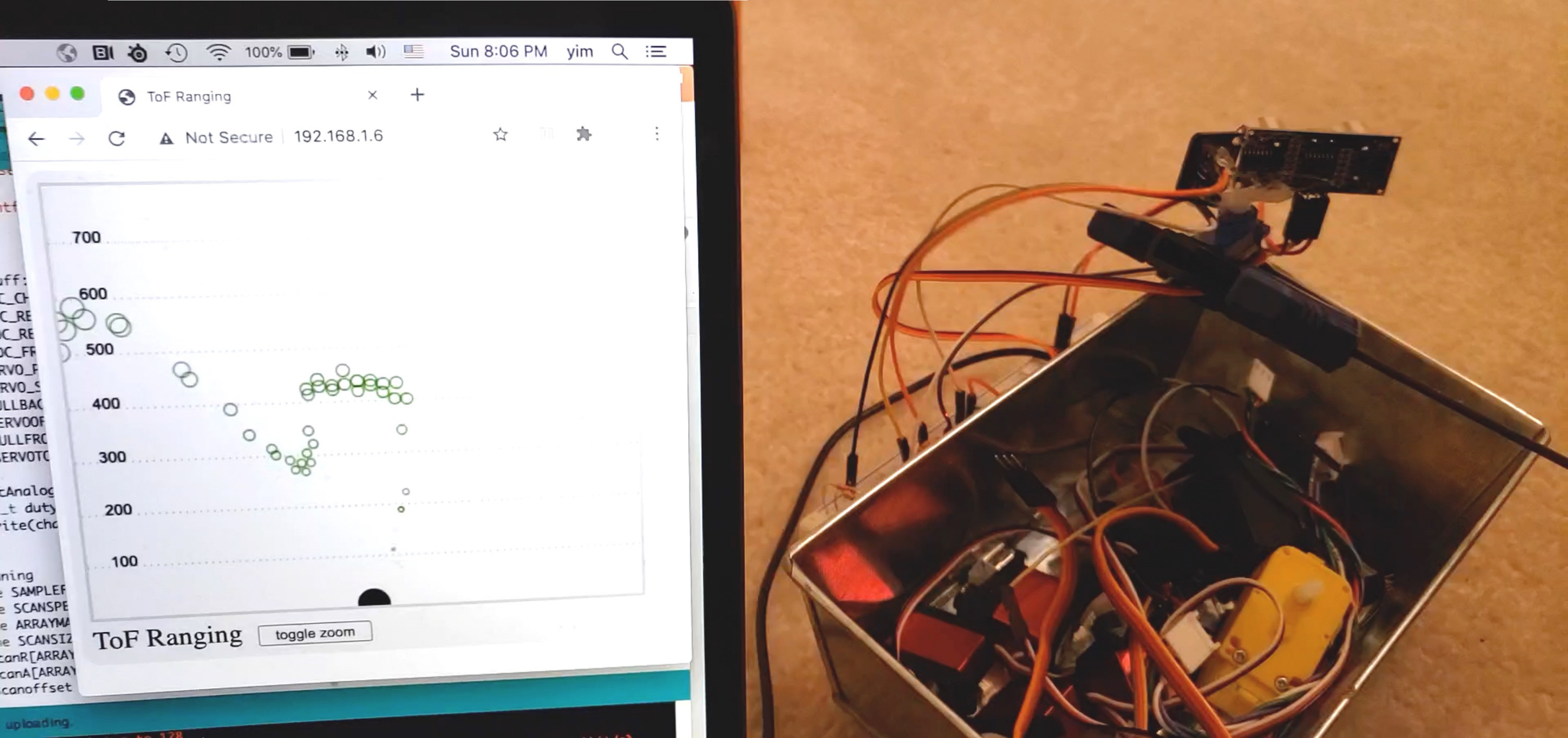
Distance Ranging example

Ranging with ToF and Ultrasonic

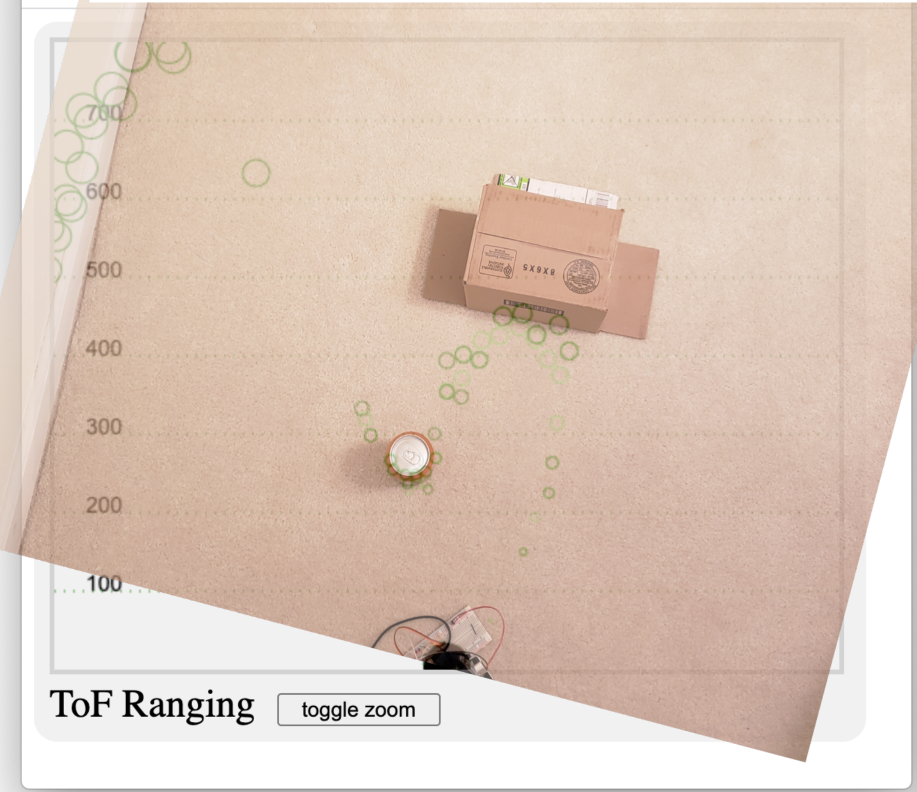
- Rangers hotglued to servo horn
- Servo clamped to box
- ESP32
  - 2 GPIO for ultrasonic
  - I2C for ranger
  - Power and ground for both
- Code runs set up to choose one sensor.

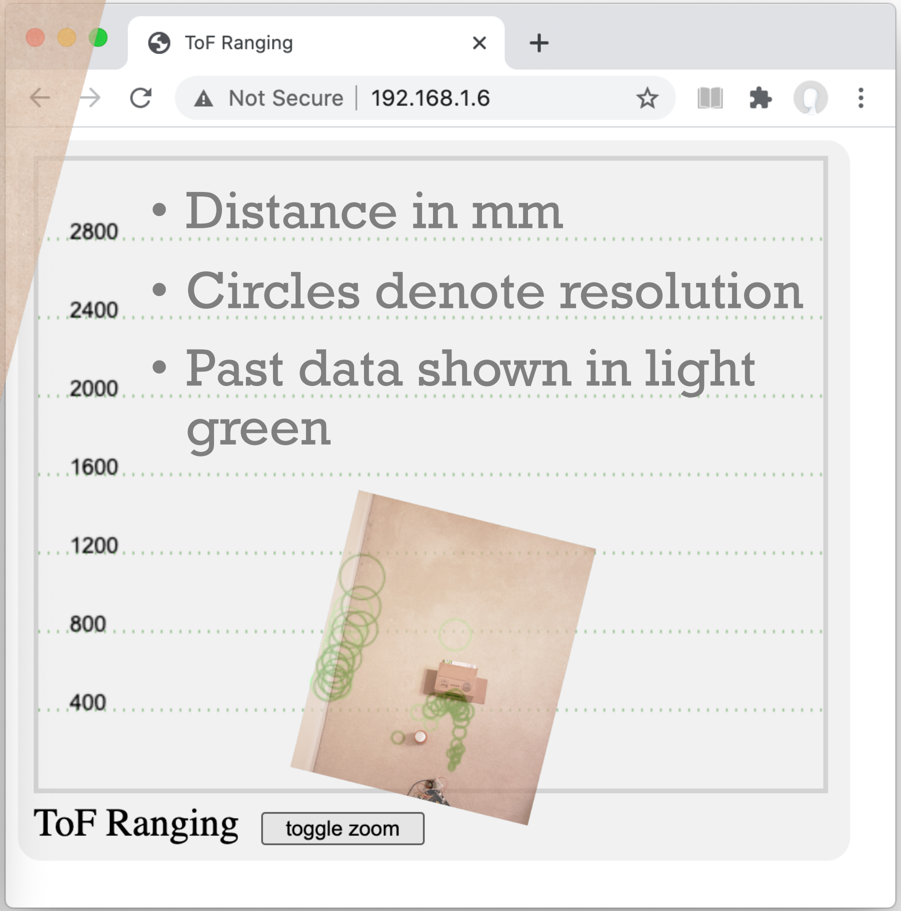


ToF Ranging example



Display explanation





- Distance in mm
- Circles denote resolution
- Past data shown in light green

Ranging Code Has 5 parts

✓ Range Sensor Reading

✓ Servo Commands

- Scanning and storing

✓ Web communication

- Javascript Display

Code Structure

```
void setup() {
  ...// normal setup for serial, web, sensors, servo
  h.attachHandler("/up",handleUpdate);
  h.attachHandler("/ ",handleRoot);
}


void loop() {
  static uint32_t lastServo = micros(), lastmicros = micros(), us = micros();

  us = micros();
  if (us-lastmicros > 5000){     // check for webpage request @ 200Hz
    lastmicros = us;
    h.serve();
  }
  if (us-lastServo > 1000000/SAMPLEFREQ) { // update servo pos at SAMPLEFREQ
    scanStep(rangeToF());   // can use rangeSonar() for ultrasonic
    lastServo = us;
  }
}
```

Scanning and storing range data

range0	range1	range2	range3	range4	range5	…	range*n*
angle0	angle1	angle2	angle3	angle4	angle5	…	angle*n*
				scanoffset	**scanoffset**		**scanoffset**

```
int scanR[ARRAYMAX];

int scanA[ARRAYMAX];

int scanoffset = SCANSIZE; // current array position

void scanStep(int range) { // pass range value to be stored
  static int angle;
  static int dir=SCANSPEED;

  if (angle+SERVOOFF > FULLFRONT)  dir = -SCANSPEED;
  if (angle+SERVOOFF < FULLBACK)   dir = SCANSPEED;

  scanR[scanoffset % ARRAYMAX] = range;
  scanA[scanoffset % ARRAYMAX] = -SERVOTODEG(angle);
  scanoffset++;
  angle += dir;
  ledcAnalogWrite(LEDC_CHANNEL, SERVOOFF+angle, LEDC_RESOLUTION); // move servo
}
```

Why store values in array?
Alternative: send each point as you get it (no storage)

- Website is client so ESP32 can't push data until client asks
- Also overhead is too large. Each HTTP request results in ~1000 extra characters.
- GTA 2021 rules: 10k byte/sec transmission limit
  - -> ten packets max per second.

```
GET /L HTTP/1.1
Host: 777f4a18c8ef.ngrok.io
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Referer: http://777f4a18c8ef.ngrok.io/H
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 172.58.207.29
X-Forwarded-Proto: http
```

```
GET /favicon.ico HTTP/1.1
Host: 777f4a18c8ef.ngrok.io
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Referer: http://777f4a18c8ef.ngrok.io/L
X-Forwarded-For: 172.58.207.29
X-Forwarded-Proto: http
```

Ring Buffer (AKA Circular Buffer)

```
scanR[scanoffset % ARRAYMAX] = range;
```

STORING INTO BUFFER

data0	data1	data2	data3	data4	data5	...	data*n*
Offset	**Offset**			**Offset**	**Offset**		**Offset**

Note: reading and storing operations are asynchronous

```
s = s+","+ scanR[(scanoffset-i) % ARRAYMAX];
```

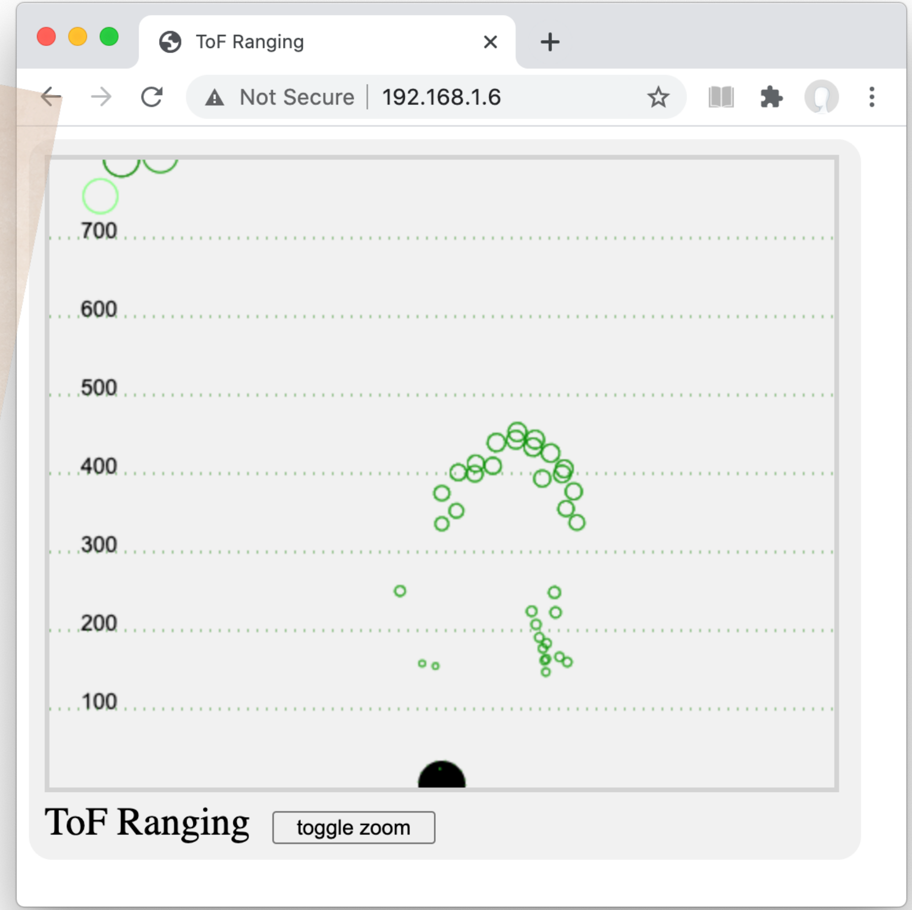
READING OUT OF BUFFER

data0	data1	data2	data3	data4	data5	...	data*n*
Off -1	**Offset**					**Off - *i***	**Off -2**

Want most recent group of data (start at current offset)

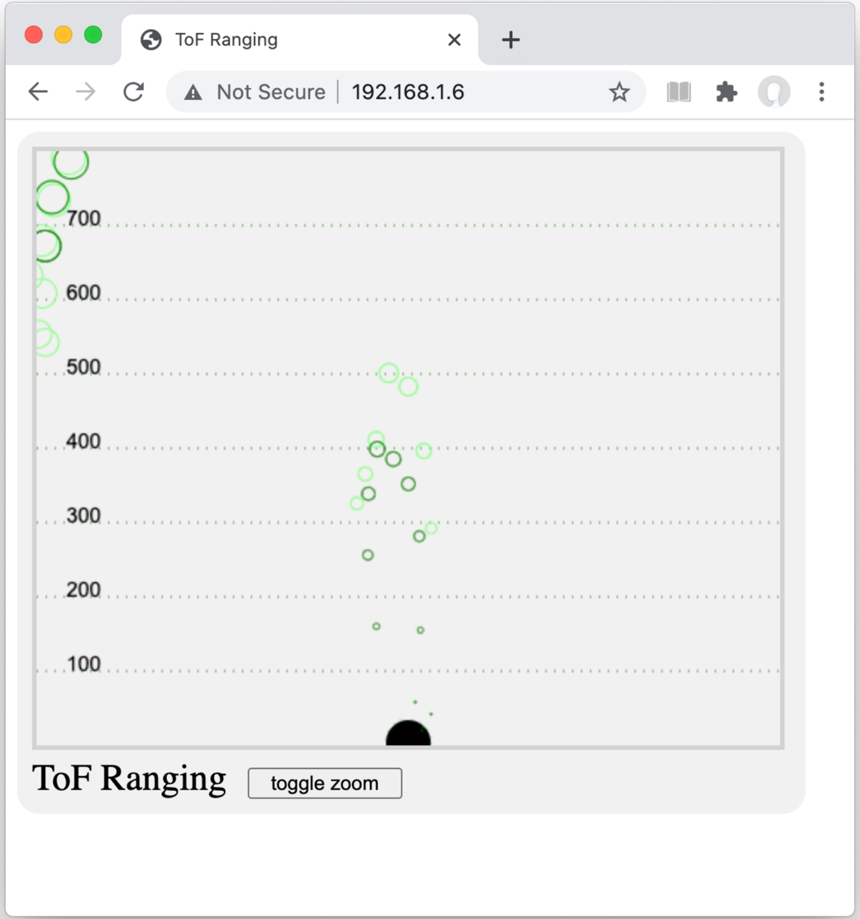
ToF Sensor

- ~1.2m range
- Sees can, box, wall fairly accurately, with some noise



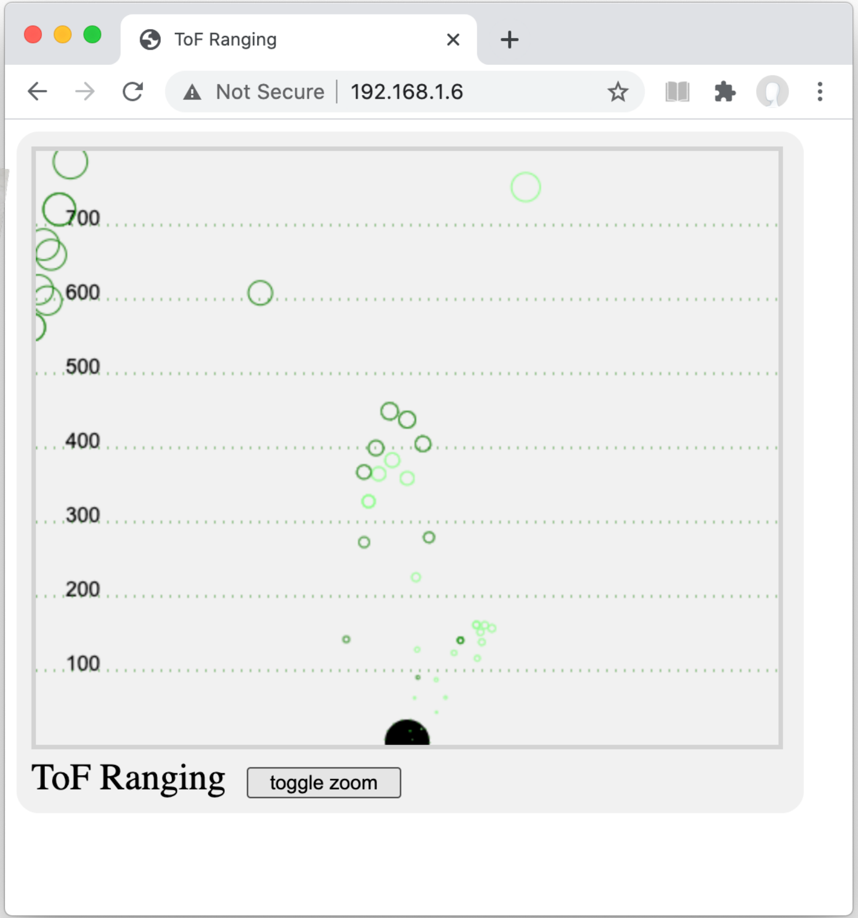
ToF

- Is it seeing the box?



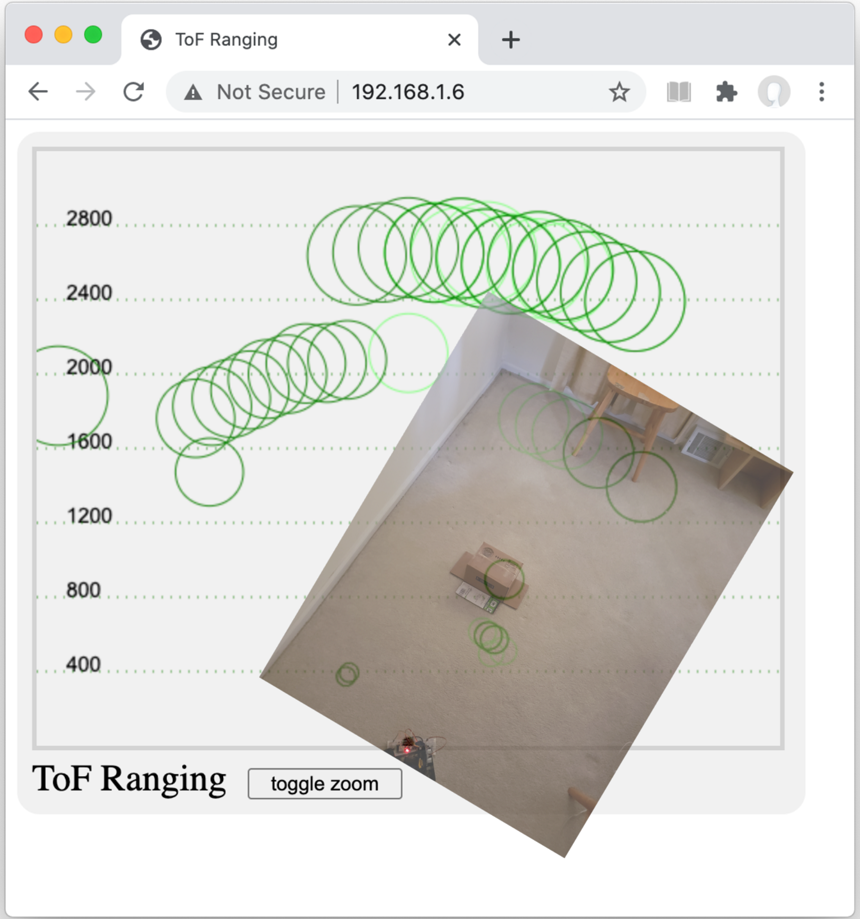
ToF

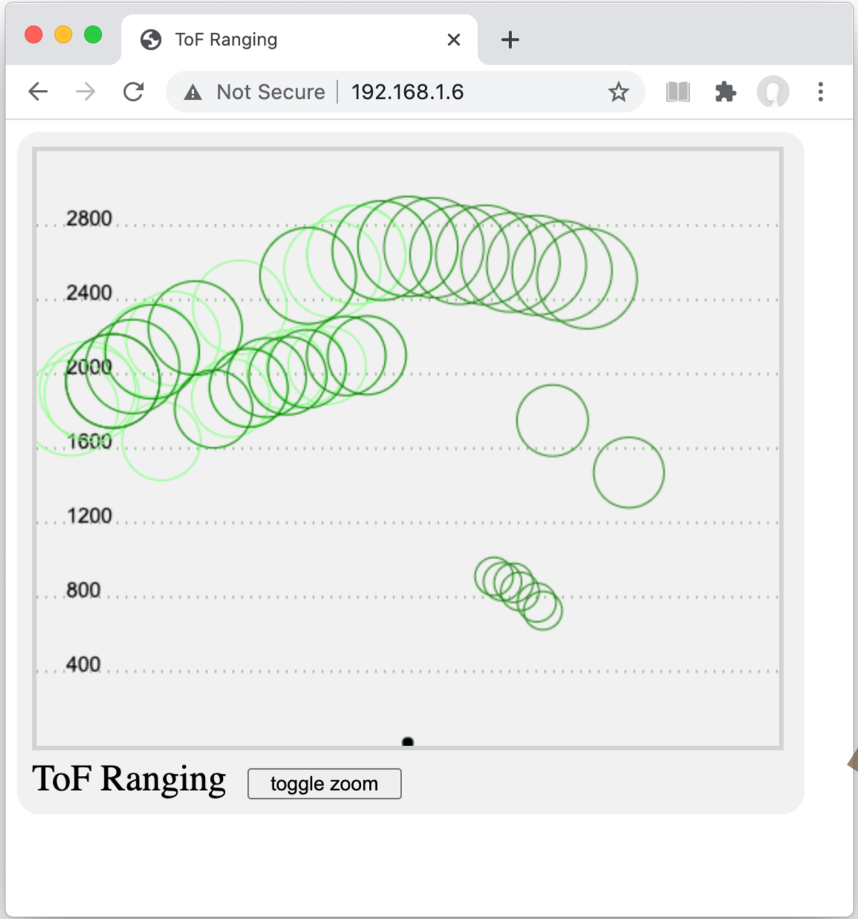
- Phantom data
- Seeing the floor
- Need to angle up slightly



Ultrasonic

- Note longer range.
- Random phantom points
- Wall ranges are distorted
- Specular reflection helps (doesn't see floor)

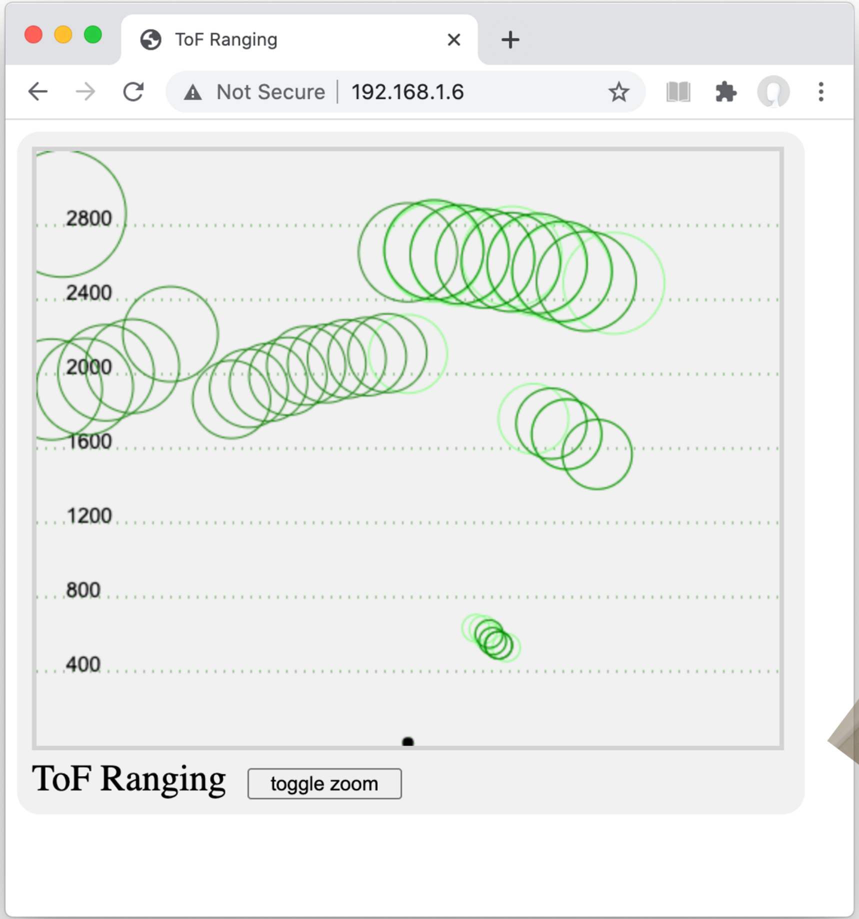




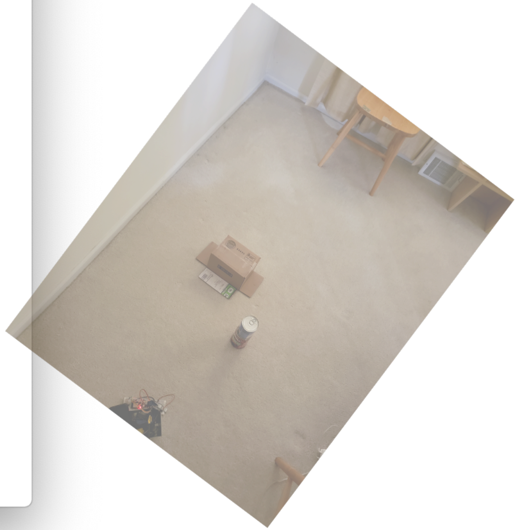
Ultrasonic

- Sees box clearly when perpendicular to sensor.
- Specularity could be a problem
  - E.g. when flat surfaces are at an angle > ~10deg dending on distance

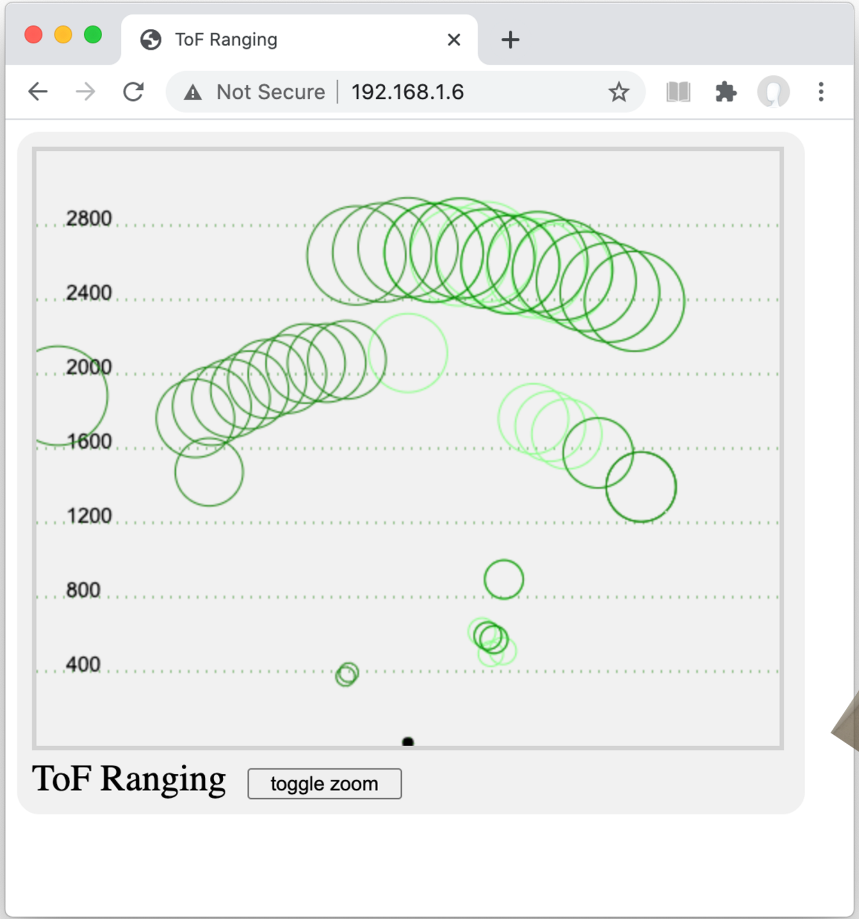
Ultrasonic



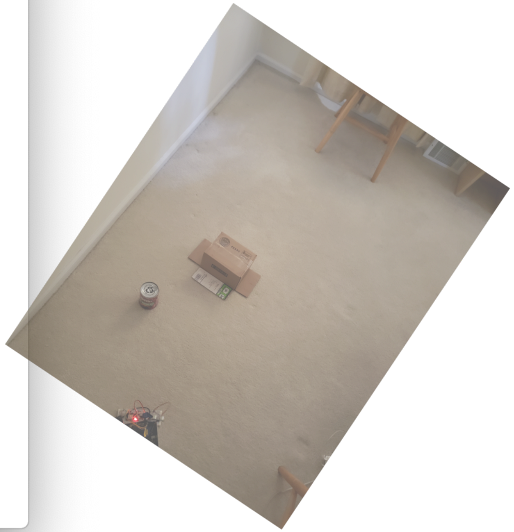
- Can't see the can



Ultrasonic



- Maybe sees can, maybe noise…



Quiz on scanning

For each question, answer:

TOF, Ultrasonic, Both or Neither.

Q7: Which would you use to see a small object on the floor?

Q8: Which has longer range?

Q9: Which cannot see soda cans reliably

Q10: Which is not likely to see the 2x4 field walls?

03 extra

Master-slave recap

I2C Master – Slave communications

WRITE SEQUENCE

M: Start / Address (write)

S: Ack (0)

M: Write Data

S: Ack (0)

READ SEQUENCE

M: Start Address (read)

S: Ack (0)

S: Read Data

M: Nak (1)

I2C Master – Slave communications

WRITE SEQUENCE

Only master can initiate transfer.

Me: Joe!	• Address (write)
Joe: Yes?	• Ack (0)
Me: Think about resistor color for 2.	• Write Data
Joe: Ok	• Ack (0)

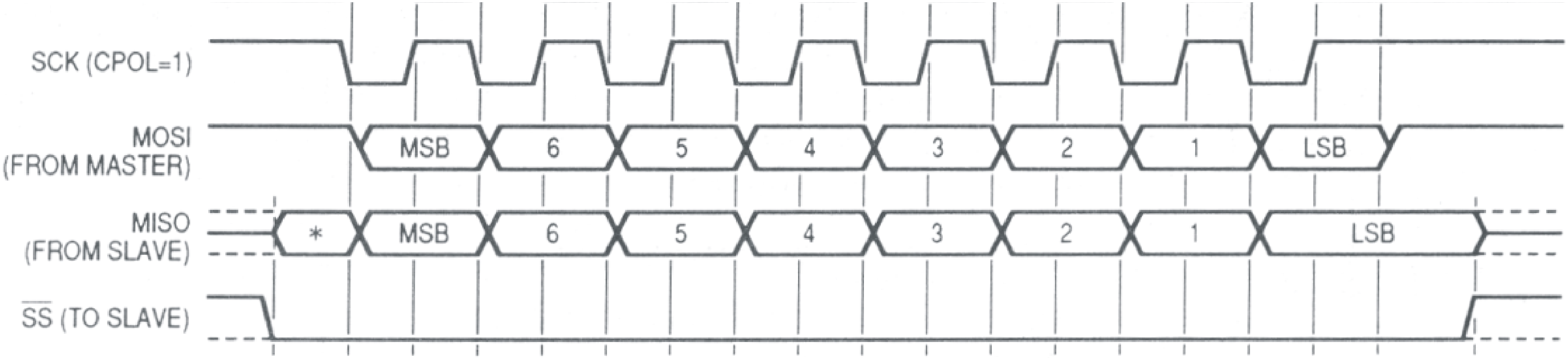
READ SEQUENCE

Me: Joe?	• Start / Address (read)
Joe: Yes?	• Ack (0)
Joe: Red	• Read Data
Me: Thanks	• Nak (1)

SPI Master – Slave communications

Only master can initiate transfer.

Me: Joe	• SS line activated	
Me: What is resistor color for 2.	• Write Data	
Me: Joe	• SS line activated	
Me: What is resistor color for 3	• Write and Read Data	Joe: Red
Me: Joe	• SS line activated	
Me: What is resistor color for 4	• Write and Read Data	Joe: Orange



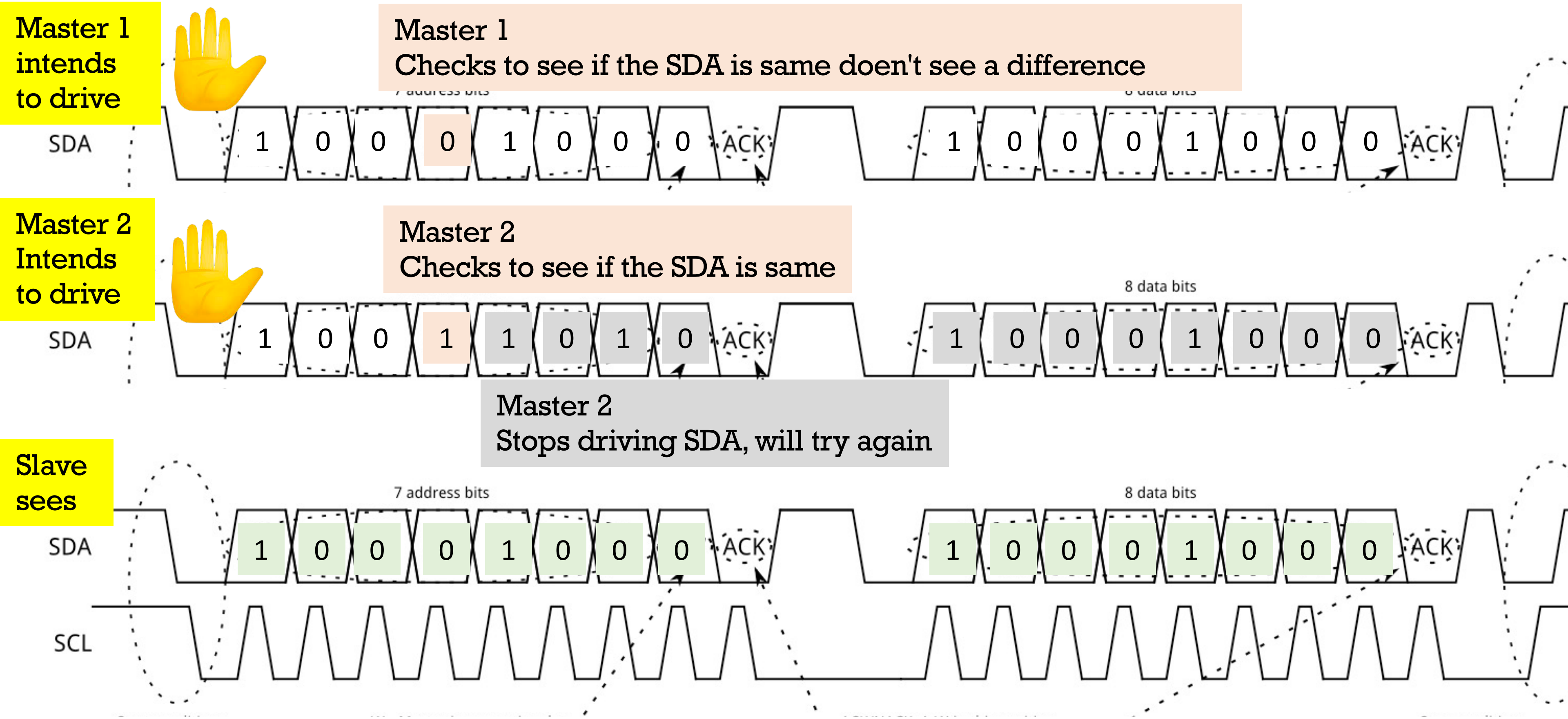
Multi-master (peer) is like Zoom

- Use zoom raise hand feature
- Answer questions on resistor values

- Students are the multiple-masters. Anyone can talk, but need arbitration to determine who talks first.

- Use zoom chat feature to private chat questions.
- Students are clients. Teacher is server.

I2C multi-master arbitration – relies on OC output "wired-AND"



Other arbitration (collision detection)

- SPI – cannot happen, single master only
- Ethernet – random backoff time.
  - Randomly choose one of 1024 time slots
  - If collision occurs back off again (upto 16 times)
  - Impacts usage for real-time (though large variable frame size is the main reason).
- WiFi – RF frequencies avoid transmitting at same time.
  - 802.11 has CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) and RTS/CTS (Request To Send/Clear To Send).
  - Random backoff time if collision expected
  - ESP32 WiFi has limited concurrent clients (maybe 4?)

Answer in CHAT

Answer how you feel about each topic below with:

1. I don't understand this topic at all
2. I don't know now, but know what to do to get by
3. I understand some, but expect to get the rest later
4. I understand completely already

A. Conductive noise and how to deal with it
B. Capacitive noise and how to deal with it
C. Inductive noise and how to deal with it
D. Ring buffers and scanning data