Report

Ultrasound Thickness (UT) using the Vine Robot - CHARM Lab Sustainability Accelerator

Ege Turan

One of the most common and traditional methods for non-destructive measurement (gauging) to get structural integrity of a surface material is ultrasonic thickness (UT) gauge or ultrasonic wall thickness gauge. It measures the local thickness of a solid element (typically made of metal, if using ultrasound testing for industrial purposes) based on the time taken by the ultrasound wave to return to the surface.

$$l_m = ct/2$$

where

l_m is the thickness of the sample

c is the velocity of sound in the given sample

t is the traverse time

The formula features division by two because usually the instrumentation emits and records the ultrasound wave on the same side of the sample using the fact that it is reflected on the boundary of the element. Thus, the time corresponds to traversing the sample twice.

The wave is usually emitted by a piezoelectric cell or EMAT (electromagnetic acoustic transducer) sensor that is built into the measurement sensor head and the same sensor is used to record the reflected wave. The sound wave has a spherical pattern of propagation and will undergo different phenomena like multipath reflection or diffraction. The measurement does not need to be affected by these since the first recorded return will normally be the head of the emitted wave traveling at the shortest distance which is equivalent to the thickness of the sample. All other returns can be discarded or might be processed using more complicated strategies.

Advantages	Disadvantages
• Non-destructive technique • Does not require access to both sides of the sample • Can be engineered to cope with coatings, linings, etc. • Good accuracy (0.1 mm and less) can be achieved using standard timing techniques • Can be easily deployed, does not require laboratory conditions • Relatively cheap equipment • EMAT does not require the use of couplant. • EMAT can conduct thickness measurements through corrosion and other surface coatings on metals • No need to remove the coating of the metal.	• Usually requires calibration for each material • Requires good contact with the material • Cannot take measurement over rust (Does not apply to EMAT) • Requires coupling material between the measured surface and the probe. (Does not apply to EMAT) • Interpretation needs experience

Rough Timeline (7/3 - 9/7) [Plan date: June 19, 2025]

Week 1	Literature & Technical Review Product & Market Survey
Week 2	System Components & Electronics Overview 1.) Identify subsystems 2.) Study the handling of signal generation, amplification, filtering, and digitization
Week 3	Commercial UT Probe Dissection
Week 4	Commercial UT Probe Dissection
Week 5	Breadboard Pulse Generation and Reception
Week 6 Week 7	Breadboard Signal Conditioning and Processing 1.) Design and test analog signal conditioning chain 2.) Raw data reception, storage, and processing
Week 8	Advanced Signal Processing and Thickness Measurement Visualization
Week 9	Advanced Signal Processing and Thickness Measurement Visualization
Week 10	Ultrasonic Sensor Integration at Scale

Contents

Comparison of common nondestructive testing (NDT) methods

Method	Best for	Pros	Cons
Ultrasonic (UT)	Wall thickness, internal corrosion, cracks	Deep penetration, accurate thickness, detects internal flaws	Needs couplant, limited on rough/dirty surfaces
Eddy Current (ECT)	Surface/subsurface cracks in conductive pipes	No couplant, fast, sensitive to small cracks	Shallow depth (~6 mm), only for conductive pipes
Radiography (RT)	Weld quality, internal flaws	Permanent record, good for hidden defects	Radiation safety, costly, slow
Magnetic Particle (MT)	Surface cracks in ferromagnetic pipes	Cheap, quick, easy to apply	Only ferromagnetic, surface/subsurface only
Dye Penetrant (PT)	Surface-breaking cracks	Very low cost, simple	Surface only, prep required

Application

- UTM is frequently used to monitor metal thickness or weld quality in industrial settings such as mining. NDE technicians equipped with portable UTM probes reach steel plating in sides, tanks, decks and the superstructure. They can read its thickness by simply touching the steel with the measurement head (transducer). Contact is usually assured by first removing visible corrosion scale and then applying petroleum jelly or another couplant before pressing the probe against metal. However, when UTM is used with an electromagnetic acoustic transducer the use of couplant is not required. These testing methods are used to inspect metal to determine quality and safety without destroying or compromising its integrity. It is a requirement of many classification societies (https://en.wikipedia.org/wiki/Nondestructive_testing).
- We want to integrate this technology to the in-pipe robot (with its cameras as well as ultrasound) to create a system capable of navigating and mapping (in terms of infrastructure thickness, corrosion…) confined and hard-to-reach pipe networks. The goal would be a high-resolution reconstruction of pipelines based on camera and UT values on the generated map.
- Presentations:
    - Weekly: Weekly Updates - Ege
    - Final: EgeTuran_Summer2025_UT

Existing experimental setup

Vine Robot Base Station, Pneumatic setup, and Electronics hardware

Considerations:

- We have a lab that has a 2-inch robot and will test a 6-inch robot, so we need to find out our limit for how small this UT system can be, and find an alternative (such as eddy current sensor to calculate the thickness of pipe) for the smaller robot width if necessary.
- We already use ESP32s in our lab. We need to find out whether it will be powerful enough or we will need a side chip to do some computation.

Introduction

For the proposed UT system, we hope to deliver physical mechanisms, circuit system components, industry components (+ outreach to companies, maybe robotic solutions).

Examples

DroneMAT

- Advanced Ultrasonic Thickness Gauge for Industrial UAVs
    - No surface preparation or couplant
    - Robust thickness measurements
    - Lightweight Compact & Modular
    - No force required from UAV to take measurements

0 Contact Sheet

https://docs.google.com/spreadsheets/d/14D2hpGbUpyjWfjxsvG3LziXw14hG6tVVnGVS426y1Zs/edit?usp=sharing

1.1 Literature & Technical Review

Sensors

1. Mixed
    1. https://ms.copernicus.org/articles/12/479/2021/ms-12-479-2021.html#&gid=1&pid=1
        - Uses UT
            i. Gel-based impedance matching
        - Eddy current sensor to ensure the coupling gap between the ultrasonic scanning thickness measurement and the part.

- **No details on UT sensor**

2. https://www.mdpi.com/2076-0825/14/6/299
    - Uses UT
        i. Gel-based impedance matching
    - To ensure stable contact and prevent overshooting or probe rebound, a pressure sensor (FSR400 from Interlink Electronics, Fremont, CA, USA) was attached between the probe and its holder to provide feedback and prevent measurement errors
    - The linear actuator (LA-T8W-6-30-100/155-32 from Leioutejidian, Shenzhen, China) has a stroke range of 0 mm to 100 mm with a movement speed of 30 mm/s. We confirmed that the gel pump (LFP101ADB from Hangzhoulifu, Hangzhou, China) can dispense a sufficient amount of coupling gel for one measurement in 15 s. The maximum injection volume was 2 mL/min when using water. The rest of the parts were made by 3D printers. The linear actuator, gel pump, and servo motor (MG90S from Tower Pro, Singapore) were controlled by an Arduino UNO-compatible board.The probe was connected to the **ultrasonic thickness sensor (SW-6510S from SNDWAY, Dongguan, China)**.

2. Ultrasound (UT) https://en.wikipedia.org/wiki/Ultrasonic_thickness_measurement (UTM)
    a. Impedance matching:
        i. Gel-based: mechanical syringe for ultrasound gel near the probe
            1. https://ieeexplore.ieee.org/abstract/document/10637169
                - Uses UT
                - Gel-based impedance matching
                - **SW-6510S by SNDWAY**
        ii. Pad-based: Flexible pad attached at the probe
3. Eddy Current (ECT) https://en.wikipedia.org/wiki/Eddy-current_testing

Technical Principles:

Ultrasonic Testing:

1. https://www.youtube.com/watch?v=UM6XKvXWVFA
    a. Diagrams and experimental setup, working principles

2. https://www.youtube.com/watch?v=DgeQTw7q_xQ
    a. Current way of operating
3. https://www.youtube.com/watch?v=jliCdxQy0RU
    a. Coated (echo echo [ignore pulse 1, check 2 and 3 - also for very thin]), noncoated; dual-, single- element; delay lines

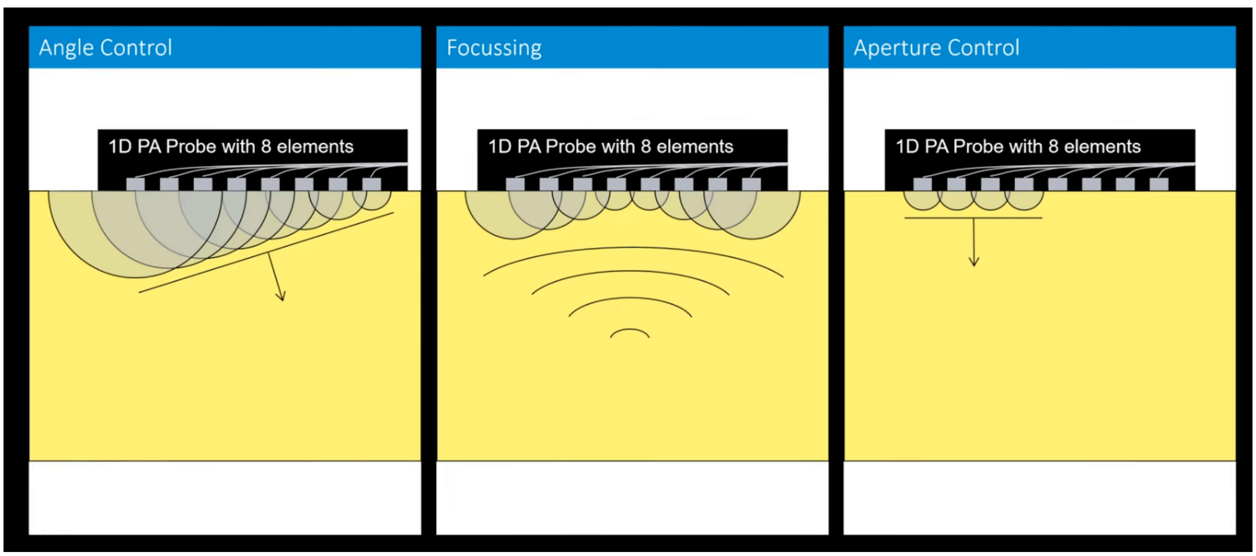
Eddy current:

1. https://www.youtube.com/watch?v=oriFJByl6Hs
    b. Diagrams and experimental setup, working principles

Phased Array Ultrasound:

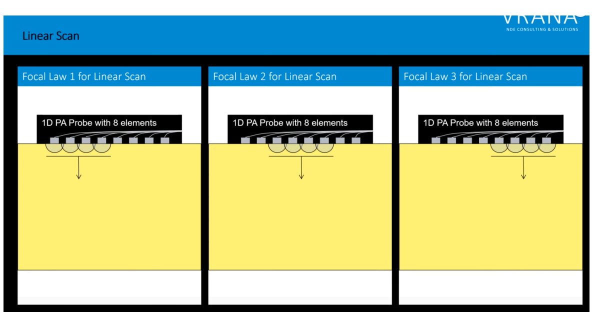
https://www.youtube.com/playlist?list=PLn8PRpmsu08q9U0y7_63Dfz5cawEnicxi

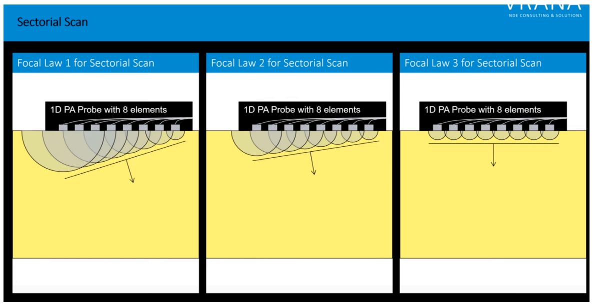
https://www.youtube.com/watch?v=g4C-zh51FM0

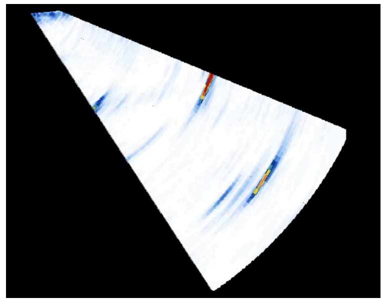


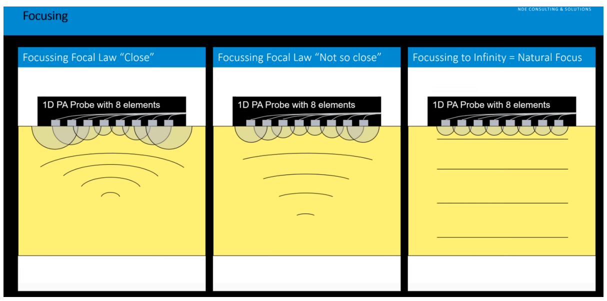
These applications can be mixed as well.

Also, OPERATION MODES:





 shear-wave inspection



More cables, more interfaces…

More expensive

Also, power distribution problem

1D → 2D phased array

1. 2 scan directions
2. Angle control in both directions
3. Better focusing

<u>Alternative solution</u>:

1. Have a long probe cable and have only the probe be at the tip of the robot
2. Have the signal processing happen at the base of the robot where we have a UT gauge that brings the raw data to whatever computer there

This website mentions a probe that is 30 meters long:

https://novotest.biz/ut-probes/#Specifications

Probes for underwater thickness measurement

5.0 MHz DE/E-E, underwater	5 MHz	2.0 – 300 mm (3.5 – 30 mm [1] through the coating)	0.078 – 11.81 in. (0.137 – 1.18 in. for Echo-Echo)	10	32.8	Ø30	114	Ø1.18	4.48	10/2	0.39/2
				20	**65.8**						
				30	**98.4**						

Requirements:

Target resolution and accuracy

Materials to be tested: priority on metal (stainless steel, cast iron + coated)

Thickness range (1 – 250 mm) [for steel pipes]

Thickness resolution (start with subcentimeter)

Spatial resolution (start with subcentimeter)

Easy interface

Small (need to be able to 90deg turn in 2in body)

Pipes according to interviews:

Concrete, steel, iron…

6-8 inch pipes

1.2 Product & Market Survey

TI dev board for Time-to-digital converter for time-of-flight (ToF) applications for LIDAR and ultrasonic

- https://www.ti.com/product/TDC7200

Small Plug-and-play Sensors

145 x 73 x 37 mm

131.3 x 63.5 x 31.5 mm

120 x 60 x 30 mm

130 x 30 x 20 mm

? x ? x ? mm

wrist-worn

- https://www.abqindustrial.net/store/dakota-ultrasonics-cx2-ultrasonic-material-thickness-gauge-z-310-0001-p-2183-op-2571_13830.html?srsltid=AfmBOor5y1E9K0z4UpPlcHYSlXAt50hRHXHIQ5XSwjkCJgF3jODojK-AFGU&gQT=1
    - 145 x 73 x 37 mm (5.7 x 2.84 x 1.46") without transducer

- https://www.abqindustrial.net/store/ultrasonic-testing-equipment-c-100/ultrasonic-thickness-gauges-c-100_2/dakota-ultrasonics-zx-2-ultrasonic-wall-thickness-gauge-z-301-0001-z-301-0003-p-1754.html?srsltid=AfmBOoql1gW9gRTPzCfR9hYcmmV8sCCz-KqNiLq2yaeDp8JeclG4W-ZA
    - 131.3 x 63.5 x 31.5 mm (5.17" x 2.5" x 1.25")
- http://dwyeromega.com/en-us/ultrasonic-thickness-gage-0-05-7-9-in-1-2-200-mm-pipe-wall-hvac-auto/Model-UTG/p/UTG?utm_source=google&utm_medium=cpc&utm_campaign=Omega-PLA-US-GGL-Connector-CatchAll&utm_content=undefined&utm_term=go_cmp-19186378522_adg-141571748582_ad-640145425429_pla-294682000766_dev-c_ext-_prd-UTG_mca-97897_sig-CjwKCAjw4K3DBhBqEiwAYtG_9BDsgSAqZJMYqw4pv_9A4CftUl3StxkS-UeJQo2x7poALT0jqt1M9BoCFo4QAvD_BwE&gad_source=1&gad_campaignid=19186378522&gbraid=0AAAAAD_f1FvEH9J1mEzODg2Mjq5zsS8Ys&gclid=CjwKCAjw4K3DBhBqEiwAYtG_9BDsgSAqZJMYqw4pv_9A4CftUl3StxkS-UeJQo2x7poALT0jqt1M9BoCFo4QAvD_BwE
    - 120 x 60 x 30 mm
- https://acs-international.com/product/pengauge/
    - 130 x 30 x 20 mm
- https://us.cygnus-instruments.com/product/cygnus-dive-underwater-gauge/?gad_source=1&gad_campaignid=20231494776&gbraid=0AAAAADJM8fReZ0KLMmzoRgGWZ7izNtZVg&gclid=CjwKCAjw4K3DBhBqEiwAYtG_9JMkwQUTbnQ6k64-i6MG-1nTjWDMK9mK3uo3F1m3dImnV8WC8bV8TxoCVkgQAvD_BwE
    - Small and unknown, rectangular prism

Projects

- https://robotik.dfki-bremen.de/en/research/projects/minoas
- Underwater ROVs: https://www.blueyerobotics.com/blog/ultrasonic-thickness-measurements-using-underwater-ROV

s

- https://us.cygnus-instruments.com/product/cygnus-mini-rov-mountable/
- Sensor developed by Cygnus, available for purchase

- Drones
  https://www.flyability.com/elios-3-ut-payload

- https://us.cygnus-instruments.com/flyability-announce-ut-probe-payload/
- Sensor developed by Cygnus, NOT available for purchase; cannot even see on this image

1.3 Project Direction Assessment

Simultaneously pursue the following 3 threads:

1. Talk to a reliable, well-known UT solution provider such as Cygnus or Olympus. Collaborate with them to engineer a tailor-made solution.
    a. Companies
        i. Cygnus https://us.cygnus-instruments.com/flyability-announce-ut-probe-payload/
        ii. Olympus https://ims.evidentscientific.com/en/flaw-detectors?creative=761181590102&keyword=olympus%20ndt&matchtype=e&network=g&device=c&campaignid=22733436176&adgroupid=184775027947&gad_source=1&gad_campaignid=22733436176&gbraid=0AAAABALRtDIoqIeaOrULm53HVygzpe57O&gclid=CjwKCAjw7MLDBhAuEiwAIeXGIYb4KMLeLFWOe_1NFzdosh6GAwn6MzpcWvYj7ZYoojvZJAMISgQf1BoCNjkQAvD_BwE
        iii. Russell Tech https://www.russelltech.com/Products/Ultrasonic-Flaw-Detectors
        iv. Voliro
            1. https://www.unmannedsystemstechnology.com/feature/a-guide-to-ultrasonic-testing-with-voliro-t-drones/
            2. https://voliro.com/voliro-solution/?utm_source=unmannedsystemstechnology.com&utm_medium=referral
        v. Eddyfi https://www.eddyfi.com/en/appnote/robotically-deployed-visual-and-ultrasonic-testing-for-extended-vertical-pipeline-assessments
        vi. XARION https://xarion.com/en/
        vii. Flying Vision https://www.flying-vision.com/en/ndt-products/ndt-ultrasonic-testing/ruc/
        viii. ASNT https://www.asnt.org/what-is-nondestructive-testing/methods/ultrasonic-testing
        ix. Searching more
    b. Outreach email template:

Hello, My name is [NAME], and I'm a [TITLE]. We're developing a next-generation in-pipe robotic platform with exceptional capabilities for navigating confined and tortuous environments. As we move toward spinning out our technology, we are actively seeking to integrate advanced ultrasonic testing (UT) sensors into our system. We've learned that [COMPANY] is a global leader in UT sensing and [has experience with integration

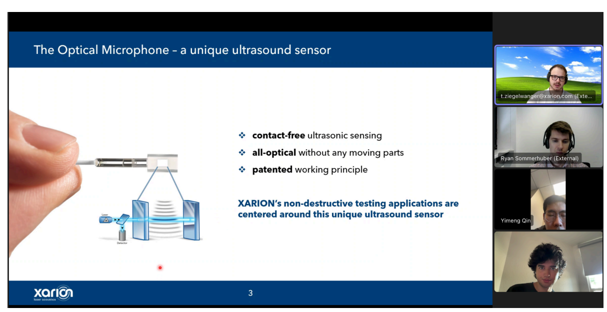
> into drone platforms].
>
> We're very interested in exploring potential collaboration and would appreciate the opportunity to learn more about your technology and capabilities.
>
> Would it be possible to set up a meeting in the coming weeks?
>
> Best regards,
> [NAME]

2. Find an off-the-shelf solution (maybe modify it to fit size and other constraints).
    a. Amazon purchase to be dissected: https://www.amazon.com/Benetech-GM100-Ultrasonic-Thickness-Measuring/dp/B0C32WYY58/ref=sr_1_4?crid=2L4P95GY9PND9&dib=eyJ2IjoiMSJ9.D79pXOxjEuVwydWBzZJz3dGbt7Z97ctiSohdR5uT-VXn3ylyDLOUAicKsC27LOKmE0lrvs2zTLmSVKesv5sIY4_c0a-xO_zBAWmDG1K5p02DCX6gq1r7dPgK1tDYFBBP-RBT9KO6uyBioSPOCHPCqkrgb9O_2BwUTW-9URMiTEvGI_vNszI8qqQr_XJS1dl8G8r5f3YieHgGN1mRVxjQGfNC_6wlQ4Wty0H-abEG3mY.NAESYJMotIgPI6Eo1IRbv-mrmXPGUk3CDwzztYViX1U&dib_tag=se&keywords=Ultrasonic+Thickness+Gauge+metal&qid=1752085858&sprefix=ultrasonic+thickness+gauge+metal%2Caps%2C177&sr=8-4

3. Develop from the ground up our own full solution.

The rest of this document will focus mostly on this 3rd thread.

1.4 Interview Notes

Evident/Olympus	**Xarian** (for NDT)
Steven Labreck Preston Gutelius Kristofer Flugan Edwin Avila Elisa Jose 3x4 in is the smallest Smallest probes are usually not recommended… 2 echo timing, when water in-between (immersion offset) = no airgap	

Coupling: gel or glycerine, some urethane (elastimere…)

If fluid is water or smt, maybe can couple
Oil or petroleum may attenuate

How much force to apply on transducer? Depends on coupling, well coupled – no pressure should be needed
Just remove the air (with liquid couplant, easy)

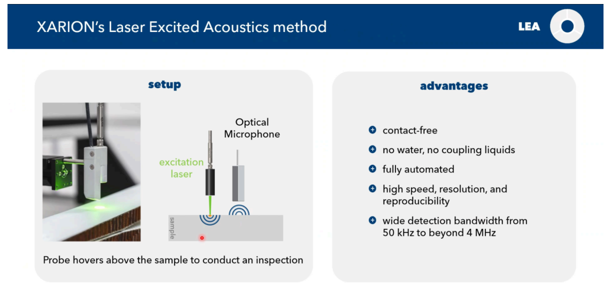
Soft urethane elastomere (gel pad) == more attenuation
More pressure needed (squeeze out any air in between), good contact, surfaces need to be clean

Within couplant bubbles / echoes

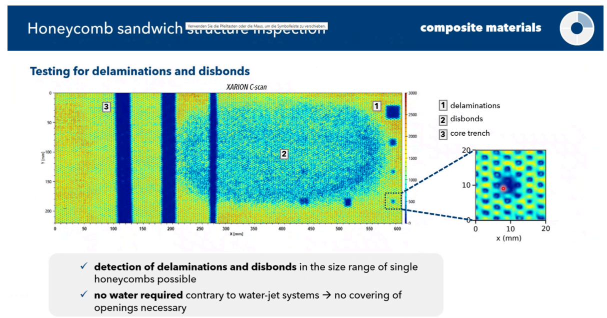
A lot of solution: flood pipe, through the pipe and comes back out, “IRIS SYSTEM”

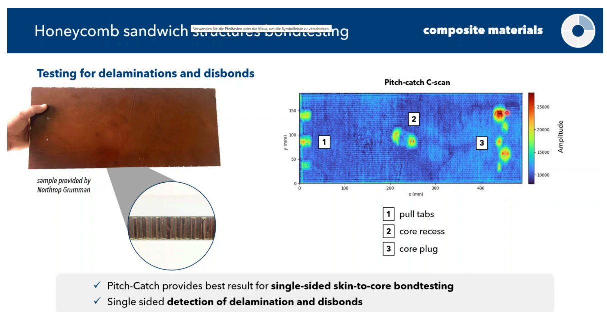
Any collaborations with robotics projects?

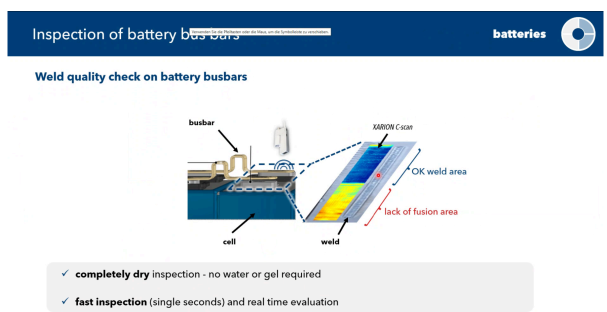
Smallest portable assembly they have: https://ims.evidentscientific.com/en/products/thickness-gauges/45mg
Maybe better for building circuits around it: https://ims.evidentscientific.com/en/products/thickness-gauges/39dl-plus

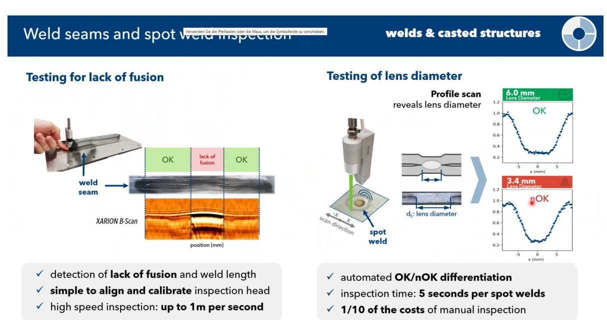


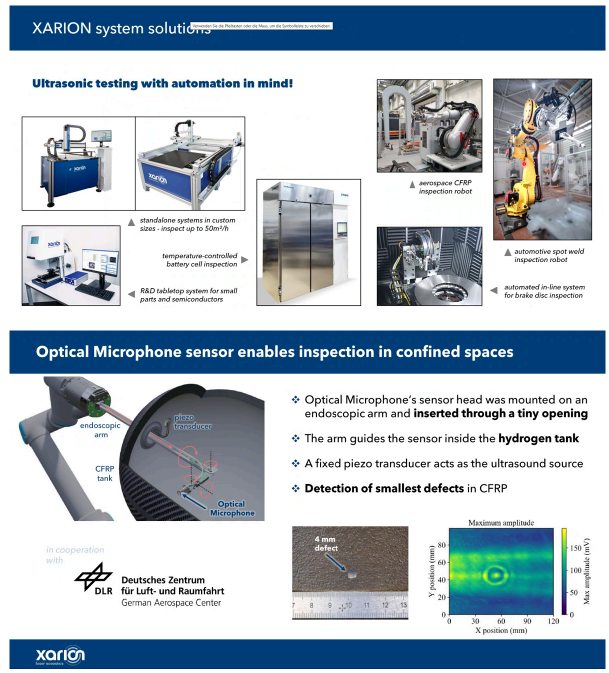
Specific parameters chosen for materials…











Bending radius of optical fiber is 3cm

3-4cm through transmission for steel pipe

Corrosion: oxide state still may reflected?

75mm bending radius for excitation fiber

Every couple of meters, excitation fiber in the pipe…

Outside pipe, listen from the inside… → defeats the purpose + underground

Our robot: Nylon coated with TPU

Can maybe do through

Materials: has to be opaque to the excitation laser

Also depends on dampening (high attenuation polymer foam may be trouble)

If metallic surface is very reflective / highly polished, that may be trouble

Our case: Steel, cast iron, PVC/plastic should be fine

	Robotics companies: is support provided → work with integrator Homemade software that does the checking… A scan, C scan Can add extra features… 24hr return policy on bugs 19 inch length 8 inch depth 2 inch height 100-200m length cables possible because of optical SAM have 150m! Excitation cable can be 15m or so!

Evident/Olympus, demo and interview with Kristofer Flugan

Notes also at Sustainability Accelerator Market Interviews and 8/21/2025: Evident Technologies
 	
We got to demo and experience two **ultrasonic thickness (UT)** devices by Evident/Olympus:
1. 45MG https://ims.evidentscientific.com/en/products/thickness-gauges/45mg (USD 2,500)
2. 39DL https://ims.evidentscientific.com/en/products/thickness-gauges/39dl-plus (USD 5,000)
We also got to see briefly a **phase-array transducer and nondestructive testing (NDT)** system capable of beam-forming, angular scans, and other multi-transducer features, namely:
- **Full Matrix Capture (FMC)**: fire one transducer, listen with every other transducer
    - **Total Focusing Method (TFM)**: Amplitude Analysis
    - **Phase Coherence Imaging (PCI)**: Normalize all the signals, discontinuity organizes differently
 	
Industry standard is 39DL according to Evident/Olympus, so we will focus on that from now on.
They also sell a lot of the D790SM transducer probe, which is 5 MHz (middle-of-the-range, general purpose)
 	
Zero-send Calibration: Because the transducer end can get deteriorated over time, the UT device does a reflection amplitude test with no gel and surface.

Calibration: They use a 1018 steel (carbon steel) block of 0.1, 0.2, 0.3, 0.4, 0.5 mm to tune the device to measure 1018 steel. They recommend to use thicknesses right below and above what you want to measure while doing this calibration.

- Idea: What if we do a best fit curve (rather than a linear interpolation)?
- Idea: What if we try to characterize a more complex fit?
- Idea: Transfer function (and subtract the baseline)

Signal-to-noise Ratio (SNR): They are used to 3:1, we saw on the screen possibly 10-20:1.

<u>Transducer probe trade-offs</u>:

<u>Single vs dual element</u>

- Single (Rx and Tx together or through-transmission) is easier to excite, possibly smaller. Better for some applications.
- Dual (Rx and Tx separate) is better SNR and dynamic range in general.
  - Ege chose dual and Evident/Olympus also demos dual.

<u>Crystal size</u>

- The bigger it is
  - More power to excite
  - More transmission (can travel deeper)
  - Worse spatial resolution for defects

<u>Frequency</u>

- The higher the speed of sound in material, the higher the frequency you need.
- The more attenuation, the lower frequency to maintain more power.
- 1-10MHz used in industry.
  - Ege chose 5 MHz and Evident/Olympus also demos 5 MHz.

<u>Impedance ratio</u>

- Z value - acoustic impedance
- Impedance ratio
- Z22-Z12)/Z12 0.75 energy transmitted, 0.25 reflected

<u>Impedance matching (acoustic)</u>

- Displacing the air between materials is the goal. Ultrasound gel is the most common method.
- We tested a gel pad. The device isn't programmed for it, but signal peaks were there, so plausible with homebrew software.

<u>Impedance electrical (electrical)</u>

- Usually 50 ohm probes.

<u>Voltage at transmit</u>:

- 24V input to the device
- The demo was done with 200V spike at the transmit
- 60-325V is the allowed range on a 39DL
  - Ege had chosen independently (±10 V input and 100 V spike at the transmit)

Probes with high temp resistance and such exist

For pipes (and other irregular surfaces) circumferentially/axially fitted probes or assemblies are possible.

Scan types:
ABC scans (mapping B can be vertical or horizontal)

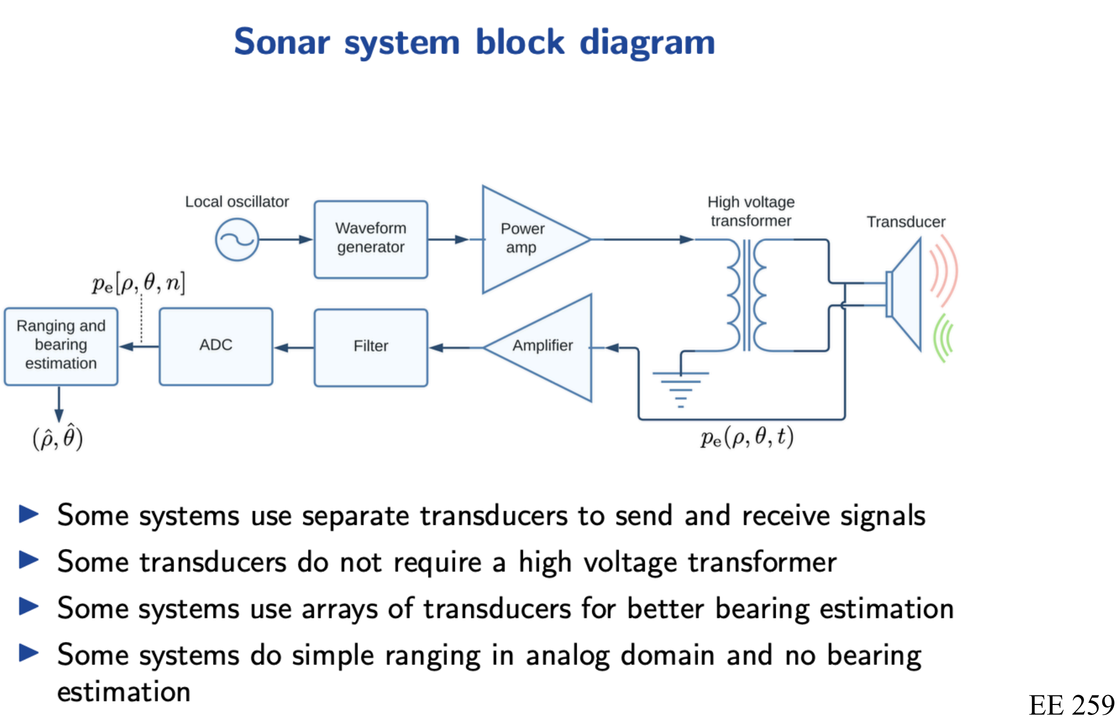
Testing types:
1. Thickness - time difference is important
2. Flaw detection - phase and amplitude information over time and space
    a. Porosity - amplitude inspection + time… how much does it attenuate
    b. Transfer functions of specific defects (comparative inspection) - standards built up

Thickness - usually ultrasonic preferred
Eddy current - for surface defects
Lead pipe - can be done based on velocity? Maybe some conductiveness measure as well

2.1 System Components & HW/SW Overview

<u>Goals</u>:

- Identify subsystems
- Study how signal generation, amplification, filtering, and digitization are handled



Target: Build a system that measures material thickness by sending an ultrasonic pulse and measuring the time it takes for the echo to return.
Core technologies: Pulser circuit, ultrasonic transducer, signal conditioning, high-speed ADC, microcontroller, display/UI, optional wireless logging.

General ideas shared across designs:

- Find a high frequency transducer (piezo) which couples well to your material in order to have good axial resolution
- Most use gel-based coupling to eliminate the air gap and impedance mismatch
- Need a transmitter and receiver circuit (pulser-receiver) and a switch to protect the receiver from the transmitter
  - For a transmitter, start with an impulse no longer than half the period of the transducer resonant frequency. The voltage should be as high as you can manage staying within the piezo's spec. The impedance of the piezo may be quite low so you may need a high current driver.
  - For the receiver, you want a low voltage noise amplifier. Probably a non-inverting opamp. Then you need to look at the signal with an oscilloscope and decide whether you can use a simple edge detection circuit or if you need to digitize the raw signal in order to do signal processing to extract the thickness.
- http://www.ti.com/solution/ultrasound_system
- TI's TDC7200 and TDC7201 do the pulse → digital time delay conversion

Most helpful projects:

1. https://www.instructables.com/Body-ultrasound-Sonography-With-Arduino/
2. FPGA-based: https://uwspace.uwaterloo.ca/items/fa07278a-a773-4219-8159-ec1c15c8ad75
3. Hacking ultrasound with a DIY dev kit
   a. https://kelu124.gitbooks.io/echomods/content/Chapter2/basicdevkit.html
   b. https://kelu124.gitbooks.io/echomods/content/devkit11.html
   c. https://kelu124.gitbooks.io/echomods/content/Chapter1/QuickStart.html
      i. https://hackaday.io/project/9281-murgen-open-source-ultrasound-imaging

*Full Mechatronics System

Essentials:

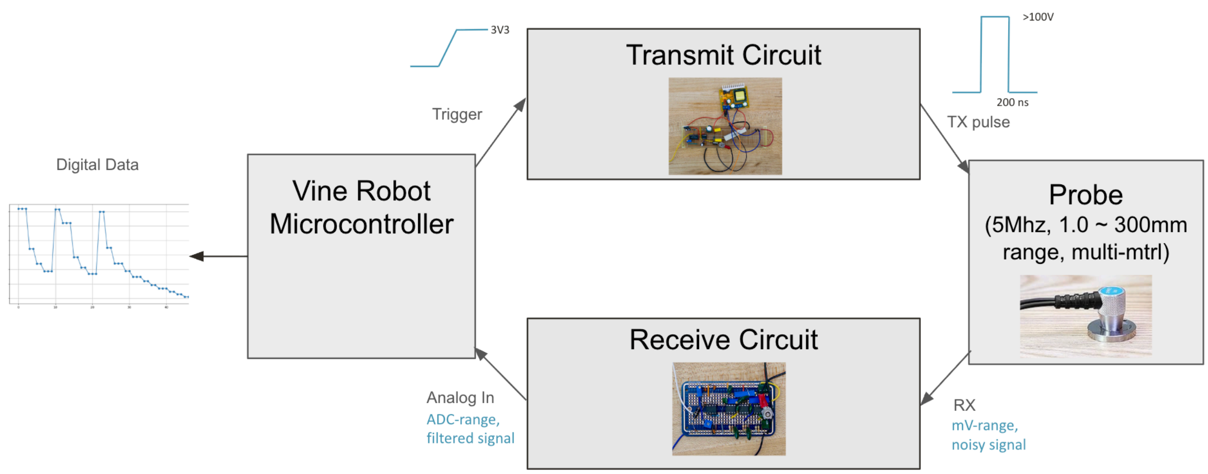
1. **UT Probe**

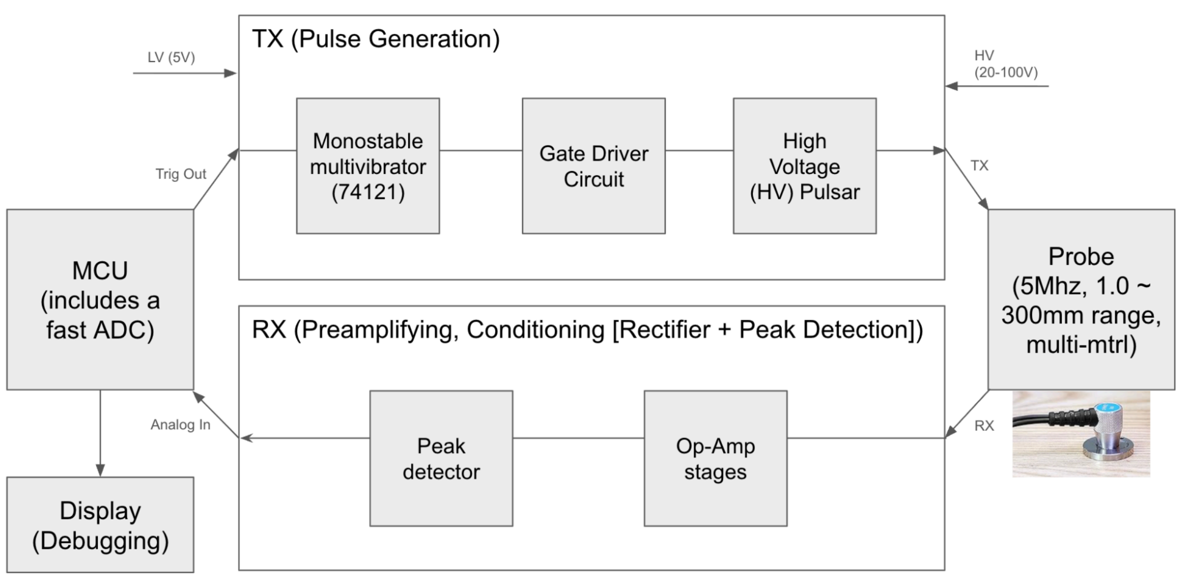
2. **Transmit circuit**
3. **Receive circuit**
4. **Power management**
5. Gel-based/pad-based mechanical coupling solution

Good to have:

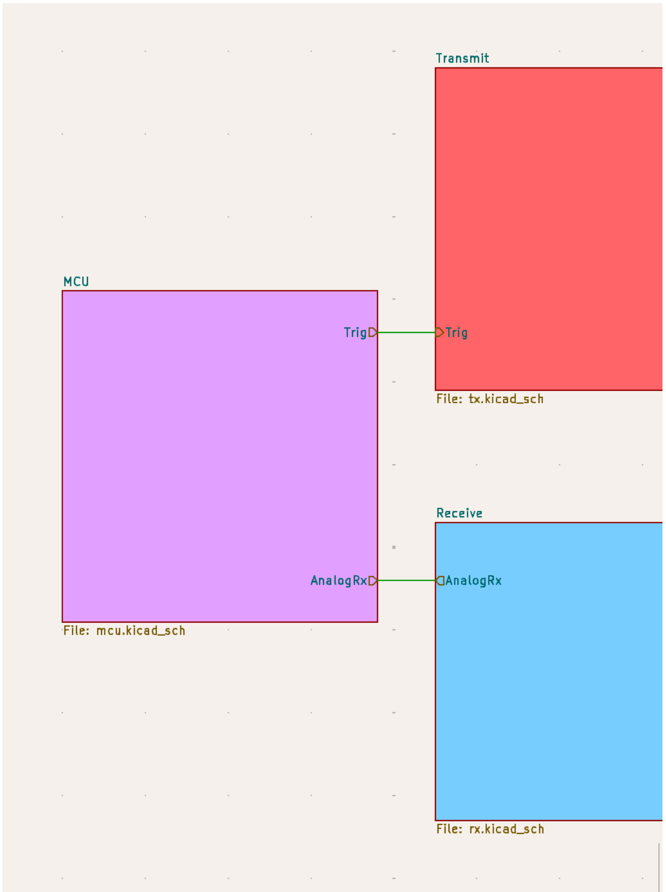
1. Surface cleaner (for probe to get a good reading)

2.2 Electrical System Block Diagram (*Essentials 1-4)





Circuit reference (top is receiver, bottom is transmitter):



Operational notes:

TX

- The GM100-transmitter works with a frequency of 5 MHz, which is flexible and middle-of-the-range (1-10 MHz). Therefore, we have to create very short pulses with a length of 100-200 nanoseconds (we achieved about 300 ns).
- The 7412-monoflop is able to create such short pulses. These short pulses go to the ICL7667-mosfet-driver, which drives the gate of an IRL620 (here, we need a high-voltage (up to 200V), high-switching-speed mosfet). If the gate is switched on, the 100V-100nF-capacitor discharges and a negative pulse of --100V is applied to the transmitter-piezo.

RX

- The ultrasonic-echoes, received from the probe, are going to a 3-stage amplifier with the fast op amp AD820.
- After the third step we need a precision-rectifier. For this purpose we use an LM7171 operational amplifier.

2.3 Testing Software Description

The microcontroller (must be fast) will store and display the reflected pulses.

https://github.com/ege-turan/ultrasound-thickness-gauge

We have two different types of fast analog-read-codes (Arduino_Due_fastanalogread_01 and Arduino_Due_fastanalogread_02). One is faster (about 0.4 µs per conversion) but I got 2-3 times the same value when reading in the analog input. The other one is a bit slower (1 µs per conversion), but hasn't the disadvantage of the repeated-values. We choose the first one since duplicates are of no concern in our application.

There are two switches on the receiver-board for our control:

1. stop the measurement
2. choose two different time-bases.

   One for measure-times between 0 and 120 µs and the other between 0 and 240 µs. We read out 300 values or 600 values. For 600 values it takes twice the time, but then I take just every second analog-in-value.

The incoming echoes are being read with one of the analog-input-ports of the arduino. The zener-diode should protect the port for too high voltages because the Arduino Due can only read voltages up to 3.3V. I chose the Arduino Due because it has a higher/faster read rate for analog signals. A single analog read process takes around 0.4 µs. A total of 300 or 600 voltages are read per scan, which takes 120 or 240 µs, respectively.

Each analog-input-value is then transformed into a value between 0 and 255. With this value a further grey-coloured-rectancle will be drawn on the display. White means high signal/echo, dark-grey or black means low signal/echo.

Display Code:

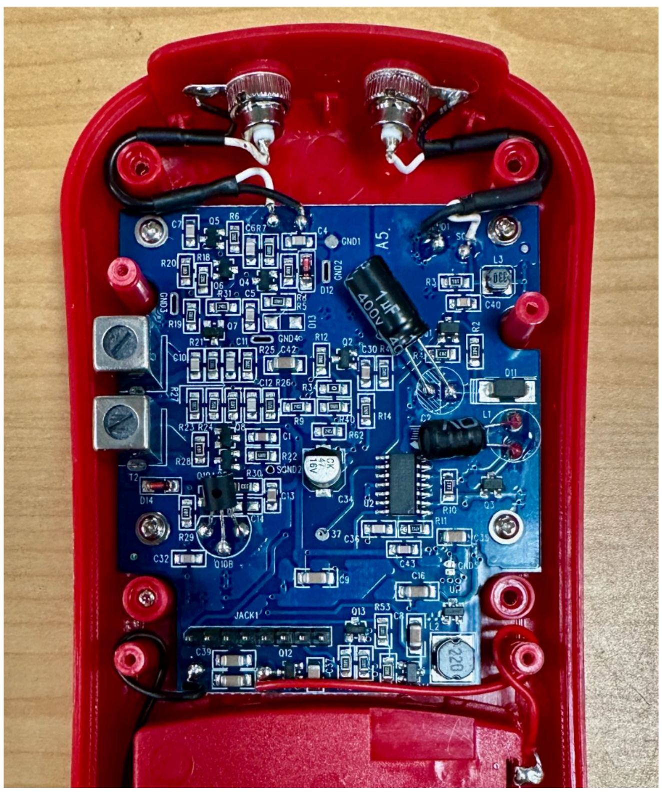
Here are the lines in the code for drawing the rectangles with 24 pixel width and 1 pixel height. This is a column-by-column loop…

```
C/C++
for(i = 0; i < 300; i++)
{

values[i] = map(values[i], 0, 4095, 0, 255);

myGLCD.setColor(values[i], values[i], values[i]);

myGLCD.fillRect(j * 24, 15 + i, j * 24 + 23, 15 + i);

}
```

3 Commercial UT Probe Dissection

Medical: https://www.youtube.com/watch?v=lmy8J8n9wPU

Ordered: GM 100



4 Breadboard Pulse Generation and Reception

Bill of Materials:

https://docs.google.com/spreadsheets/d/1GvlKbNne9WWIcBN9UraZQLvywWVBUGEUhrwNzLanVb4/edit?usp=sharing

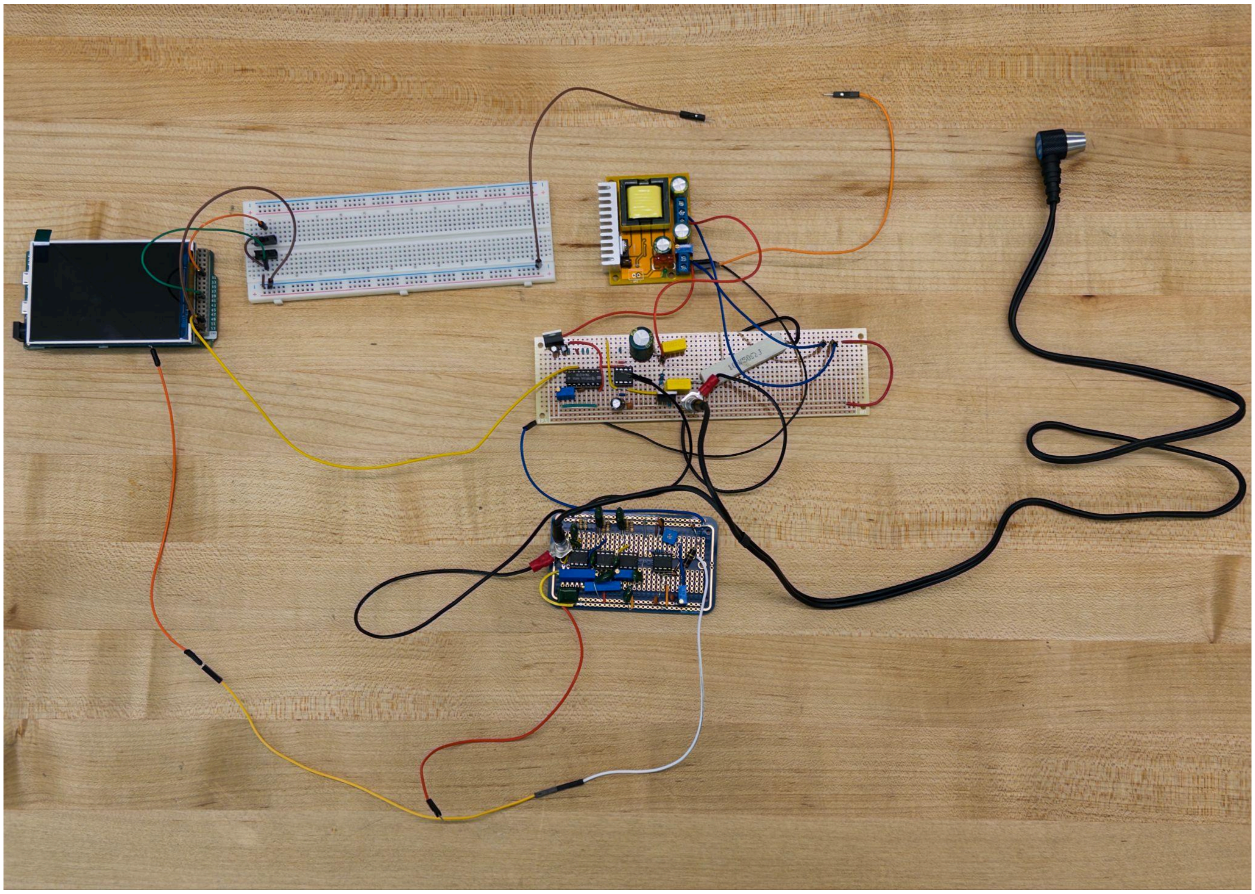
Lab Equipment:

- Keysight E36311A DC Power Supply Triple-Output

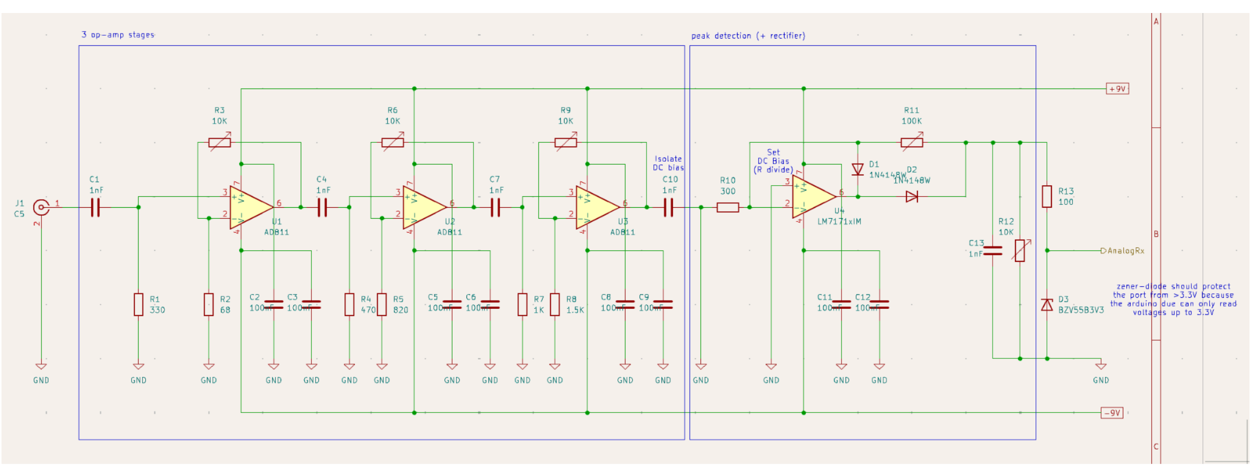
- Keysight DSOX 3024G - Digital Oscilloscope (200 MHz)

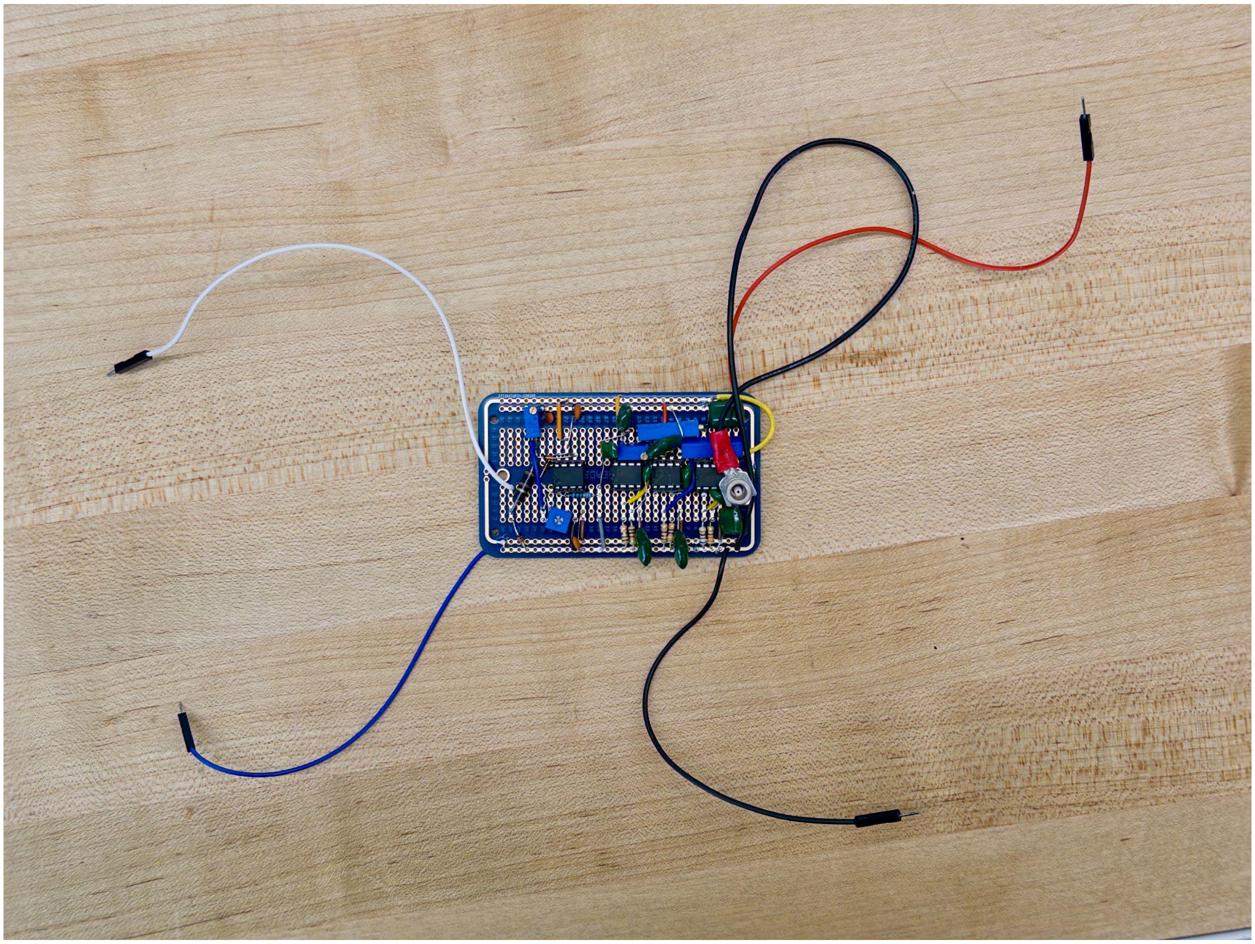
5.1 Design of Signal Conditioning and Processing Circuits

Full System

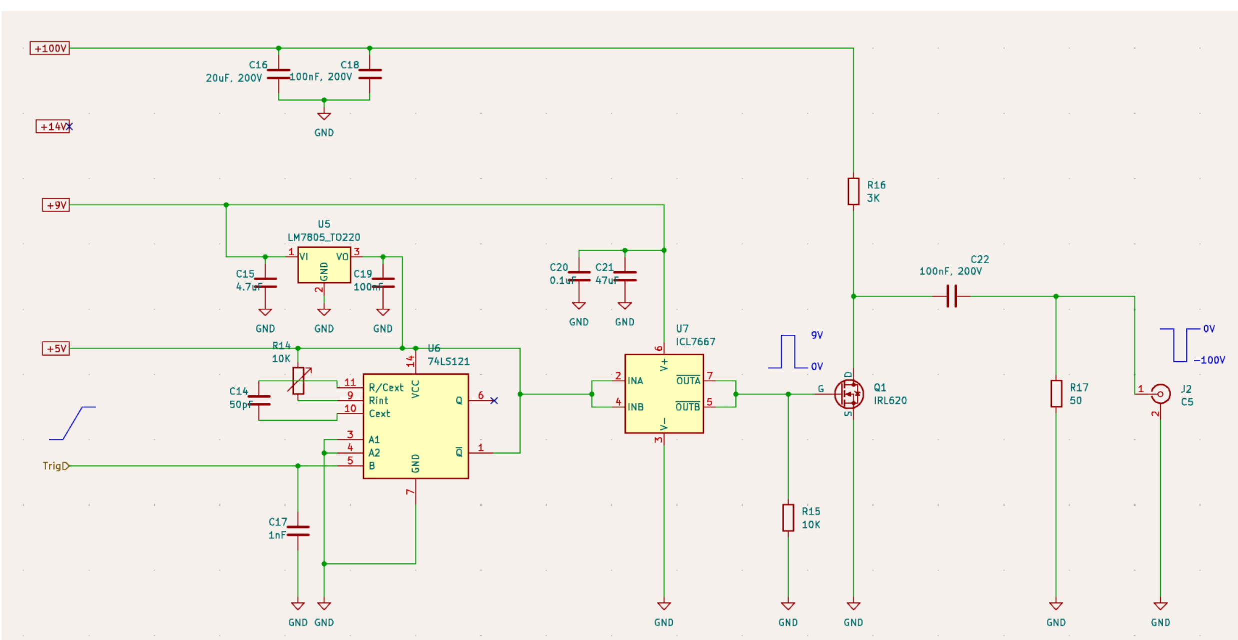


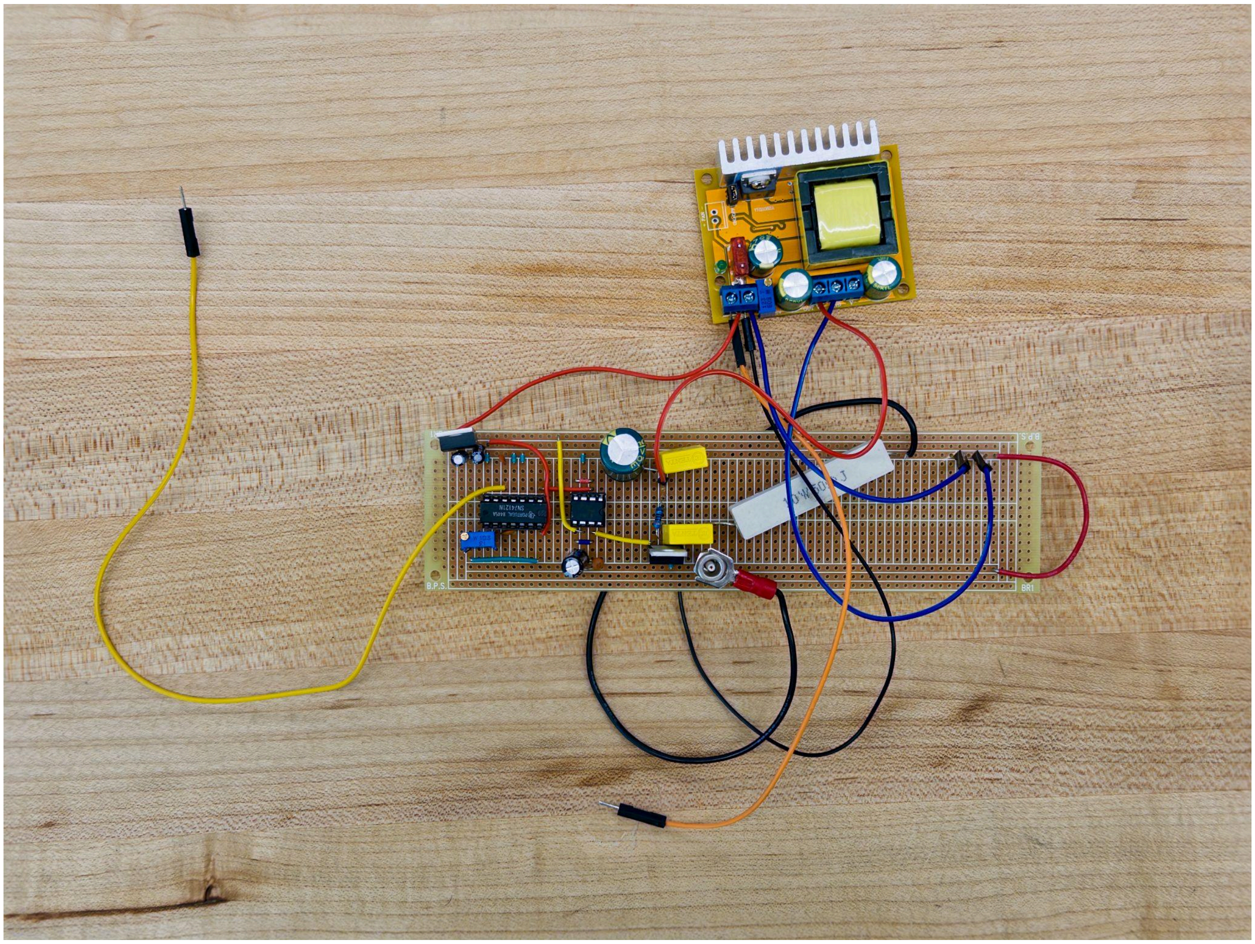
Receive Circuit



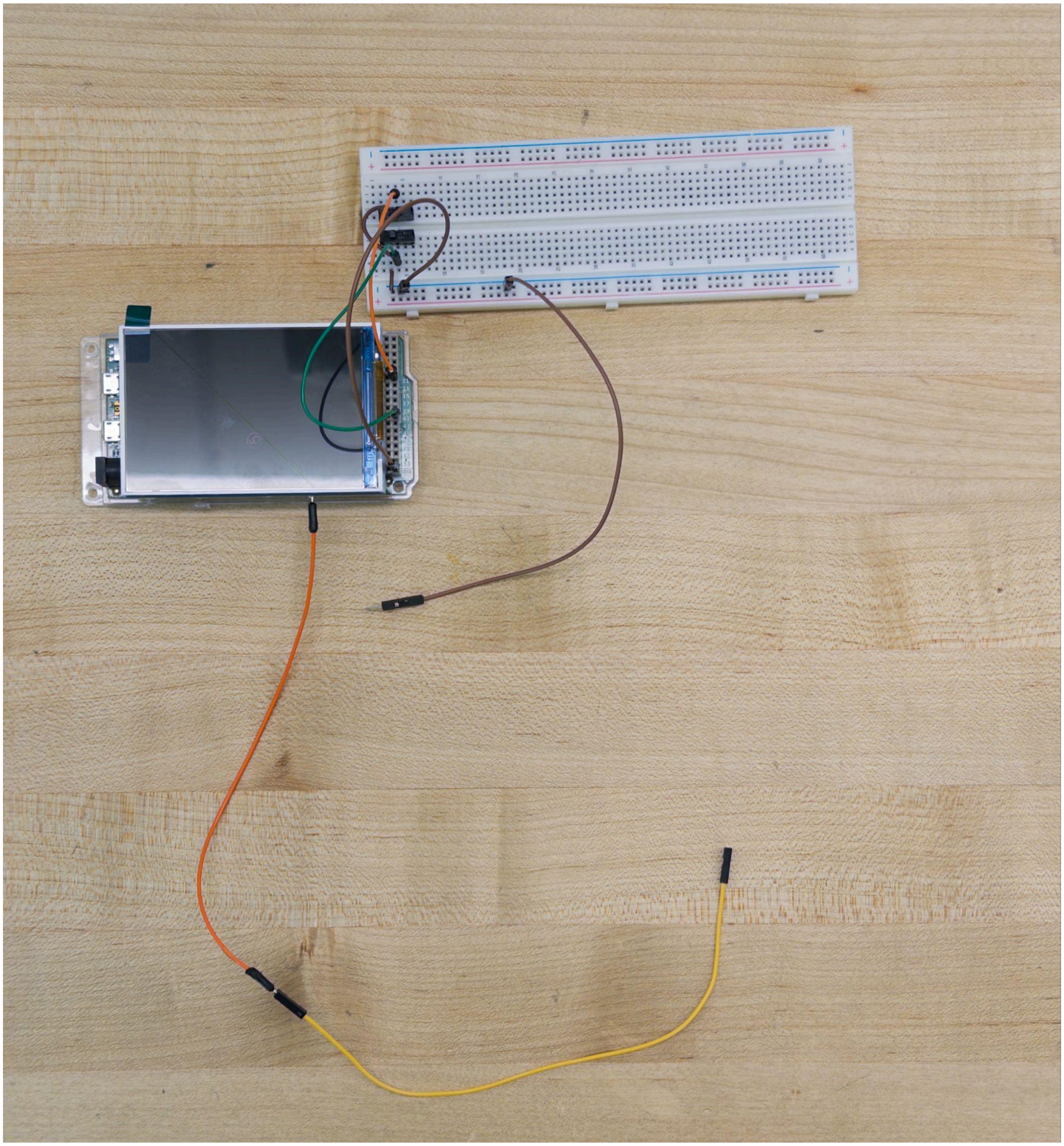


Transmit Circuit





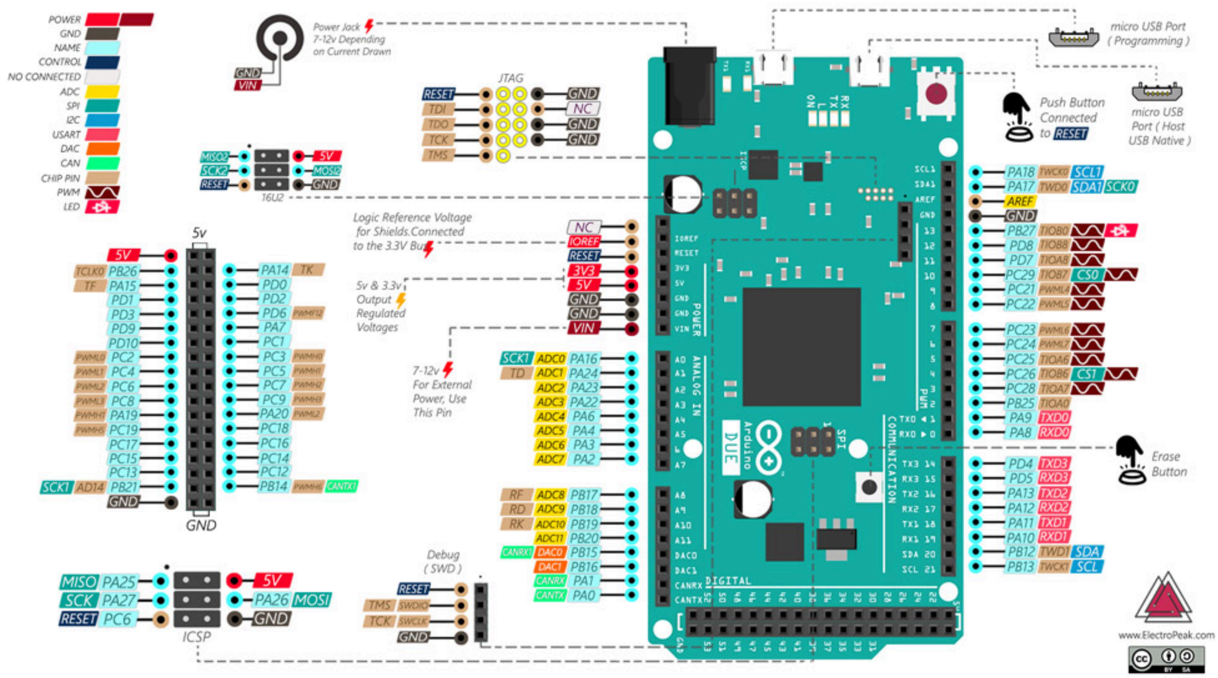
Arduino Due and User Interface (2 switches and an ILI9486 TFT display)



We use pins 31 and 41 for the switches.

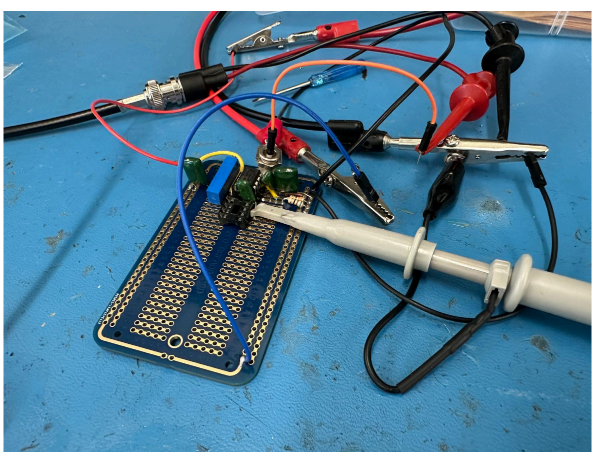
We receive the echo signal at A7 DAC and transmit the trigger at digital pin 51.

Arduino Due Pinout

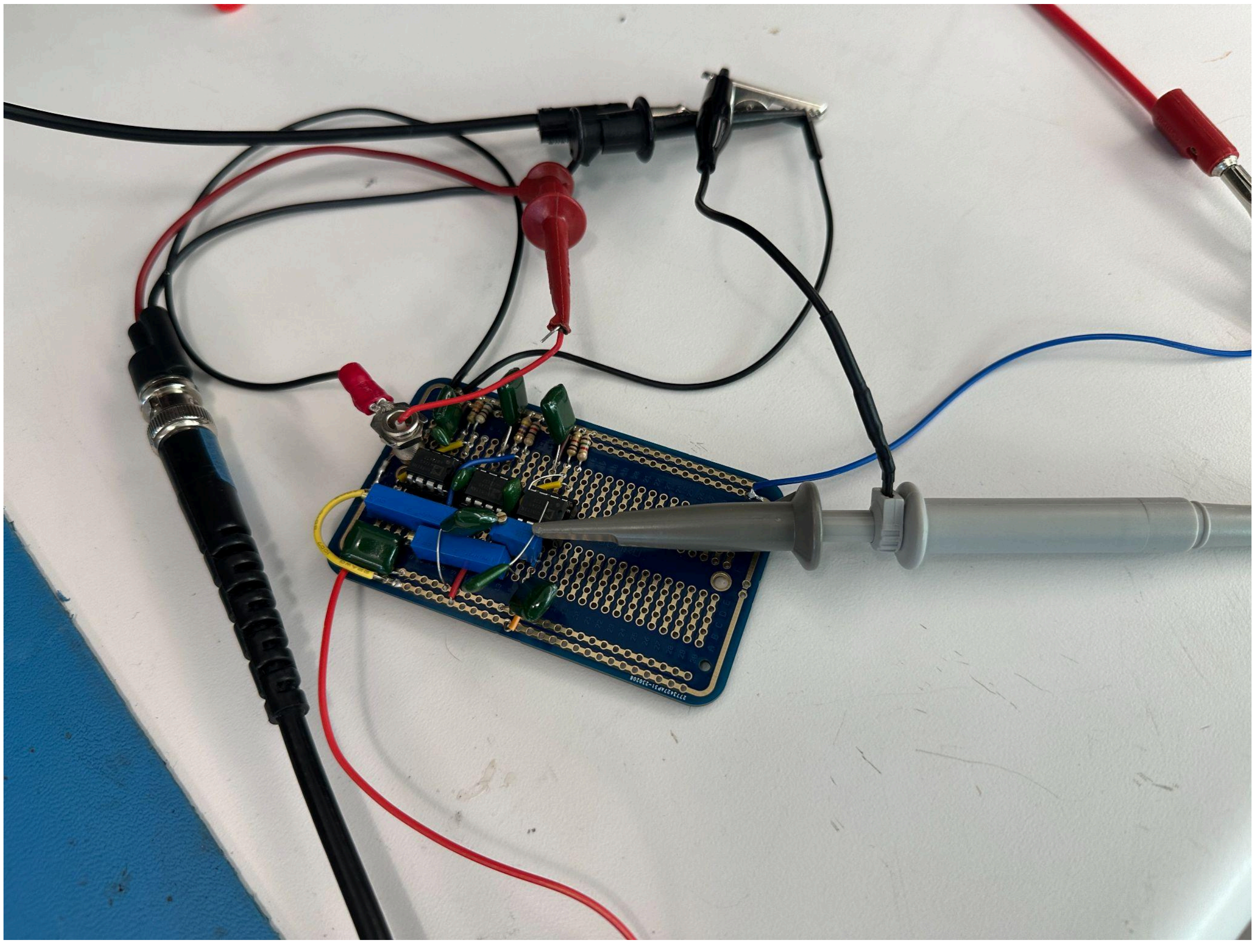


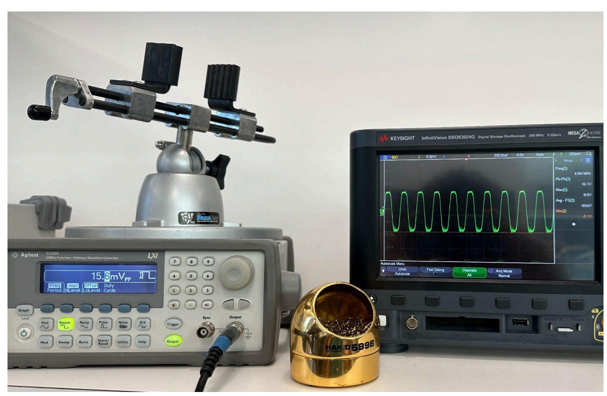
5.2 Verification of Signal Conditioning and Processing Circuits

Receive - one stage of high-pass filter and low-noise amplifier (tuned):

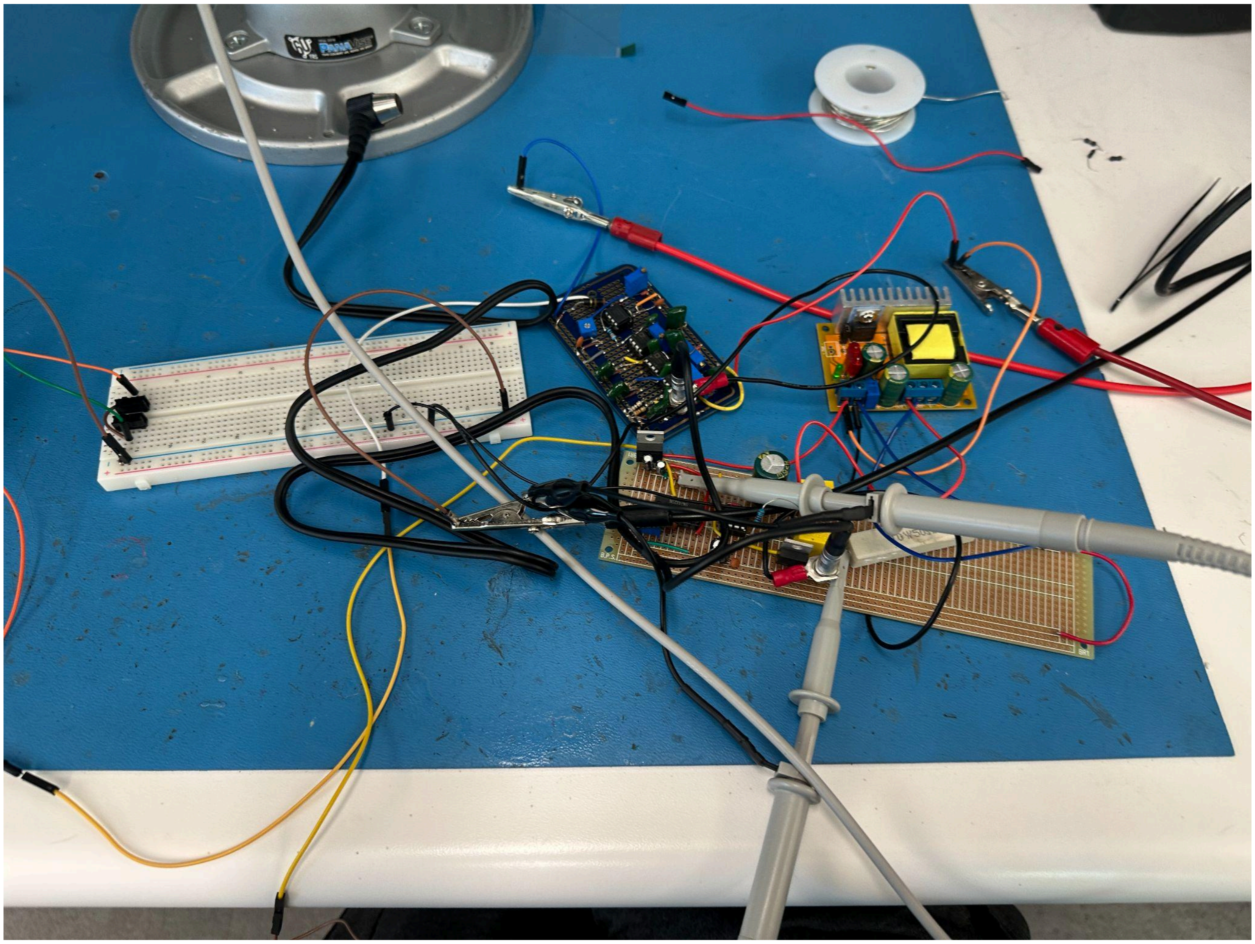
Circuit	Output for a 88.5 mVpp 5 MHz square wave input
	

Receive - three stages of high-pass filter and low-noise amplifier (tuned):



<table>
<tr><td>Input: 15.6 mVpp 5 MHz square wave</td><td>Output: 15.6 mVpp 5 MHz square wave</td></tr>
<tr><td>
</td><td></td></tr>
</table>

Transmit - We verified with the following setup that a trigger signal at the beginning of our transmit circuit leads to a 0.300us pulse at about 100V at our transmit circuit output. We validated with the following setup that a trigger signal at the beginning of our transmit:



~~We had a transmit pulse width of about 300 ns, so theoretically, we had the following uncertainty as to where our peak is: peak uncertainty/width = (½) * transmit pulse width * (6420 m/s) = 0.009648 s = 9.648 ms.~~

Power Management Circuit Verification

We are using a 9.8V and -9.8V power supply at the ±9V inputs of our circuit. (later changed to be ±10V)

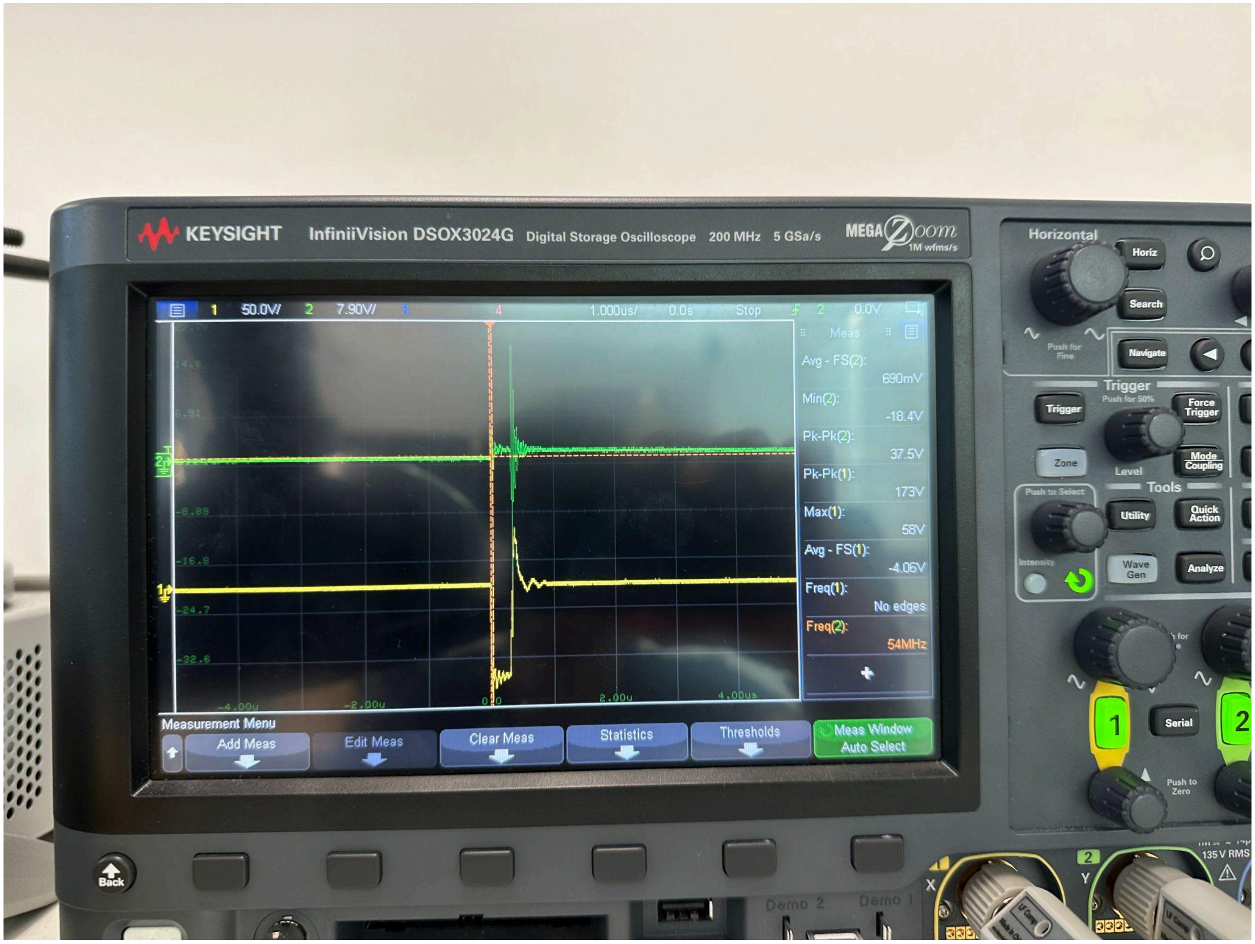
We see consistent 4.8-5.1V at the output of the 5V regulator.

We see consistent 98-101V at the output of the 100V regulator.

5.3.1 Validation of Signal Conditioning and Processing Circuits

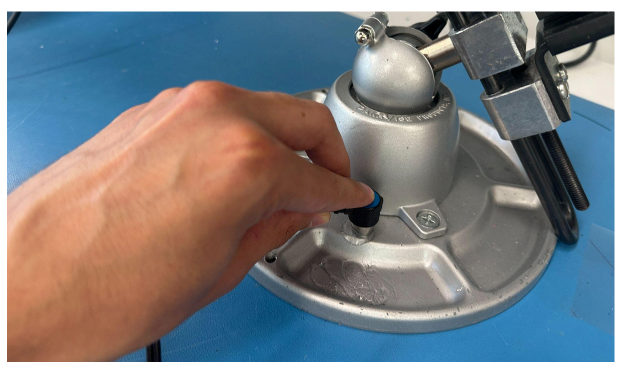
Using an oscilloscope, we got the following readings at the transmitter output (green) and the arduino input (yellow). We should see echoes a few microseconds out from the transmit the yellow waveform.

We got a baseline reading with no surface (no echo expected):



We see peaking and some ringing at the transmission and no echoes.

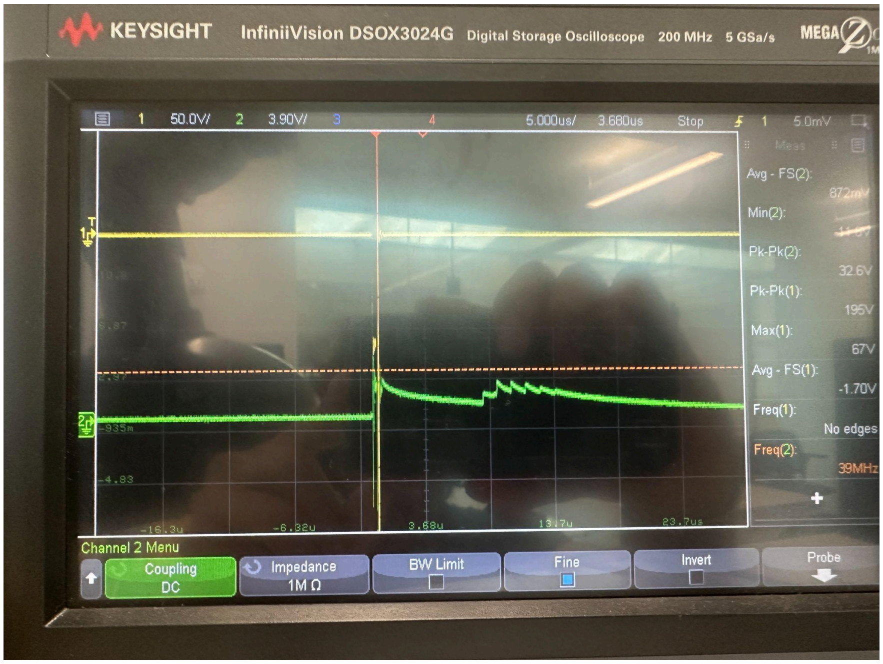
We then started our initial echo testing with a piece of steel equipment in the lab:

Detected echo at 3us

Detected echo at 7.5us

Detected echo at 9us



Detected echo at 9us

Detected echo at 10us

Detected echo at 10us (with ringing – maybe due to coupling differences)

5.3.2 Power Management Circuit Validation

The full system draws about 80mA, 100mA on 9V and 50mA on -9V power supplies.

We are using a 9.8V and -9.8V power supply at the ±9V inputs of our circuit.

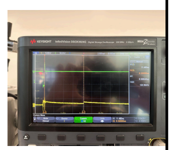
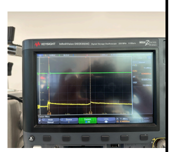
We see consistent 4.8-5.1V at the output of the 5V regulator.

We see consistent 98-101V at the output of the 100V regulator.

5.4 Raw data reception, storage, and processing

Now that we have received data on the oscilloscope, we will record and analyze ADC data to set delays, transmission coefficients, baseline threshold, delta threshold. We will then apply DSP (digital signal processing) to analyze thickness and do other NDT.

We have three pieces of metal:

Thickness (mm)	Material	Speed of sound (m/s)	Oscilloscope waveform capture	Measured constant + echo delay (us)	Expected echo delay (us)	Constant delay (us)
4.0	Aluminum 6061?	6224.4898?		8.32	1.245	7.07
6.6	Aluminum 6061?	6224.4898?		8.94	2.054	6.89
9.8	Aluminum 6061	6224.4898?		9.88	3.050	6.83

Using the above calculations and our current number of samples, we will set a constant delay of 6.93us.

Raw DAC output reception (example echo detection at sample #19):

0 1573

1 1573

2 1441

3 1441

4 1441

5 1322

6 1322

7 1322

8 1212

9 1212

10 1212
11 1123
12 1123
13 1042
14 1042
15 1042
16 960
17 960
18 960
19 1389
20 1389
21 1389
22 1260
23 1260
24 1260
25 1172
26 1172
27 1172
28 1082
29 1082
30 1004
31 1004
32 1004
33 921
34 921
35 921
36 860
…

Here, there is an echo at 19. We detect this with the following code. We also visualize and display on screen results.

Warning: The theoretical constants we calculated did not work too well, so we empirically calculated the sampling period, speed of sound in material, and the constant delay. Empirical constants are used, but theoretical ones are also given in the code.

Code from 8/16/25 https://github.com/ege-turan/ultrasound-thickness-gauge/tree/main/UT_SN_V2

```
C/C++
#include <Adafruit_GFX.h>
#include <MCUFRIEND_kbv.h>

MCUFRIEND_kbv tft;

// ----------------- Colors -----------------
#define BLACK   0x0000
#define WHITE   0xFFFF
#define YELLOW  0xFFE0

// ----------------- Pins -----------------
#define pin_output 51
#define pin_input_magnification 41
#define pin_input_sleep 31

// -------------- Empirical Constants for Thickness Calculation ---------------
#define SPEED_OF_SOUND_MM_PER_US 7.684   // 6061 Aluminum speed of sound in
mm/us
#define SAMPLE_PERIOD_US 0.23             // ADC sample period in microseconds
#define CONSTANT_DELAY_US 2.7           // Additional constant delay to
subtract
#define MAX_SAMPLES 100                  // Limit to first 100 samples per scan
#define RUNNING_AVG_LENGTH 5             // Number of scans to average
#define MIN_ECHO_SAMPLE 5

// -------------- Theoretical Constants for Thickness Calculation
---------------
// #define SPEED_OF_SOUND_MM_PER_US 6.224   // 6061 Aluminum speed of sound in
mm/us
```

```
// #define SAMPLE_PERIOD_US 0.2             // ADC sample period in
microseconds
// #define CONSTANT_DELAY_US 6.93           // Additional constant delay to
subtract


// ----------------- Globals -----------------
unsigned int values[MAX_SAMPLES];
int Trigger_time;
int i, j;

// Running average buffer
float thicknessBuffer[RUNNING_AVG_LENGTH] = {0};
int bufferIndex = 0;
int numFilled = 0; // number of valid entries in buffer

void setup() {
  Serial.begin(115200);

  // Configure ADC for Arduino Due in free-run mode
  REG_ADC_MR = 0x10380080;  // freerun, prescaler
  ADC->ADC_CHER = 0x03;     // enable ADC channels 0 and 1 (adjust if needed)

  // Pin setup
  pinMode(pin_output, OUTPUT);
  pinMode(pin_input_magnification, INPUT);
  pinMode(pin_input_sleep, INPUT);
  digitalWrite(pin_output, LOW);
  Trigger_time = 1;

  // TFT setup
  uint16_t ID = tft.readID();
  tft.begin(ID);
  tft.setRotation(1);
```

```
  tft.fillScreen(BLACK);
  tft.setTextColor(BLACK, YELLOW);
  tft.fillRect(0, 0, 432, 13, YELLOW);
  tft.setCursor(120, 1);
  tft.setTextSize(1);
  tft.print("UT - DEBUG");

  Serial.println("System Ready.");
}

void loop() {
  for (j = 0; j < 18; j++) {
    // ----------------- Trigger ultrasonic pulse -----------------
    digitalWrite(pin_output, HIGH);
    digitalWrite(pin_output, LOW);

    // ----------------- Acquire ADC samples -----------------
    for (i = 0; i < MAX_SAMPLES; i++) {
      while ((ADC->ADC_ISR & 0x03) == 0);   // Wait for ADC ready
      values[i] = ADC->ADC_CDR[0];          // Read ADC value
    }

    // ----------------- Output for Serial Plotter -----------------
    // Serial.print("Scan ");
    // Serial.print(j);
    // Serial.println(" ADC values:");
    // for (i = 0; i < MAX_SAMPLES; i++) {
    //   Serial.print(i);        // Sample number
    //   Serial.print(" ");
    //   Serial.println(values[i]); // ADC value
    // }
    // Serial.println("-----");

    // ----------------- Echo detection (second upward spike) -----------------
```

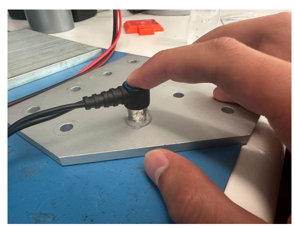
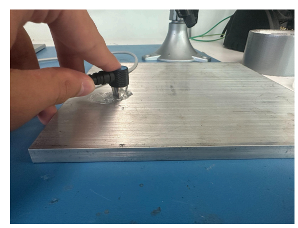
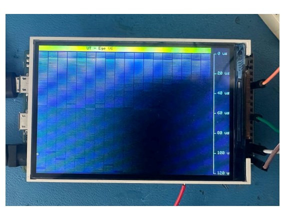
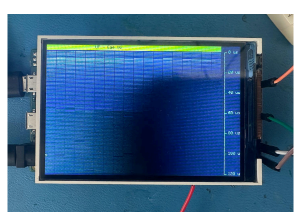
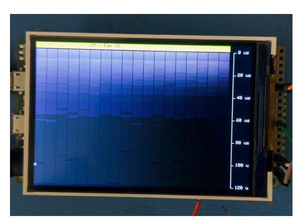
```
    int echoIndex = -1;
    int spikeCount = 0;

    for (i = MIN_ECHO_SAMPLE; i < MAX_SAMPLES; i++) {
      int delta = values[i] - values[i - 1];

      if (delta > 0) { // only upward spikes
        spikeCount++;
        if (spikeCount == 1) { // spike num
          echoIndex = i;
          break;
        }
      }
    }

    // ----------------- Thickness calculation -----------------
    float thickness_mm = 0;
    float correctedTime = 0;
    if (echoIndex >= 0) {
      float echoTime = echoIndex * SAMPLE_PERIOD_US;         // Raw time (us)
      correctedTime = echoTime - CONSTANT_DELAY_US;          // Apply delay
correction
      if (correctedTime < 0) correctedTime = 0;             // Avoid negative
time

      thickness_mm = (correctedTime * SPEED_OF_SOUND_MM_PER_US) / 2.0;

      Serial.print("Scan ");
      Serial.print(j);
      Serial.print(": Echo at sample ");
      Serial.print(echoIndex);
      Serial.print(", time = ");
      Serial.print(correctedTime);
      Serial.print(" us, thickness = ");
```

```
      Serial.print(thickness_mm, 2);
      Serial.println(" mm");

    } else {
      Serial.print("Scan ");
      Serial.print(j);
      Serial.println(": No echo detected.");
    }

    // ----------------- Update running average -----------------
    thicknessBuffer[bufferIndex] = thickness_mm;
    bufferIndex = (bufferIndex + 1) % RUNNING_AVG_LENGTH;
    if (numFilled < RUNNING_AVG_LENGTH) numFilled++;

    float runningAvg = 0;
    for (i = 0; i < numFilled; i++) {
      runningAvg += thicknessBuffer[i];
    }
    runningAvg /= numFilled;

    // ----------------- Display thickness, echo sample/time, and running
average on TFT -----------------
    tft.setTextColor(WHITE, BLACK);
    tft.setTextSize(2);
    tft.setCursor(50, 200);
    tft.print("Echo S: ");
    tft.print(echoIndex);
    tft.setCursor(50, 225);
    tft.print("Time: ");
    tft.print(correctedTime, 2);
    tft.print(" us");

    tft.setCursor(50, 250);
    tft.print("T: ");
```

```
tft.print(thickness_mm, 2);
tft.print(" mm");

tft.setCursor(200, 250);
tft.print("Avg: ");
tft.print(runningAvg, 2);
tft.print(" mm");

// ----------------- Draw timing scale (right side) -----------------
tft.setTextColor(WHITE, BLACK);
tft.setTextSize(1);
tft.drawLine(440, 15, 440, 15 + MAX_SAMPLES, WHITE);
for (int y = 15, t = 0; y <= 15 + MAX_SAMPLES; y += 20, t += 4) {
  tft.drawLine(440, y, 445, y, WHITE);
  tft.setCursor(450, y - 5);
  tft.print(t);
  tft.print(" us");
}

// ----------------- Draw grayscale scan column -----------------
for (i = 0; i < MAX_SAMPLES; i++) {
  uint8_t gray = map(values[i], 0, 4095, 0, 255);
  uint16_t color = tft.color565(gray, gray, gray);
  tft.fillRect(j * 24, 15 + i, 23, 1, color);
}

// ----------------- Sleep mode check -----------------
if (digitalRead(pin_input_sleep) == HIGH) {
  while (digitalRead(pin_input_sleep) == HIGH) {
    // Pause until released
  }
}

delay(1000); // Wait before next scan
```

```
    }
  }
```

Validation tests with various thicknesses of Aluminum 6061:

Caliper-measured thickness	4.0 mm	6.6 mm	9.8 mm
Oscilloscope-measured system echo time	8.31 us	8.97 us	9.87 us
Arduino-measured system echo samples	19	21	23
Arduino-calculated system echo time	2.14 us	2.36 us	2.59 us
Arduino-calculated thickness	3.95 mm	6.85 mm	9.75 mm
Error	-0.05 mm	+0.25 mm	-0.05 mm
Measurement			
Oscilloscope			
Arduino visualizer			

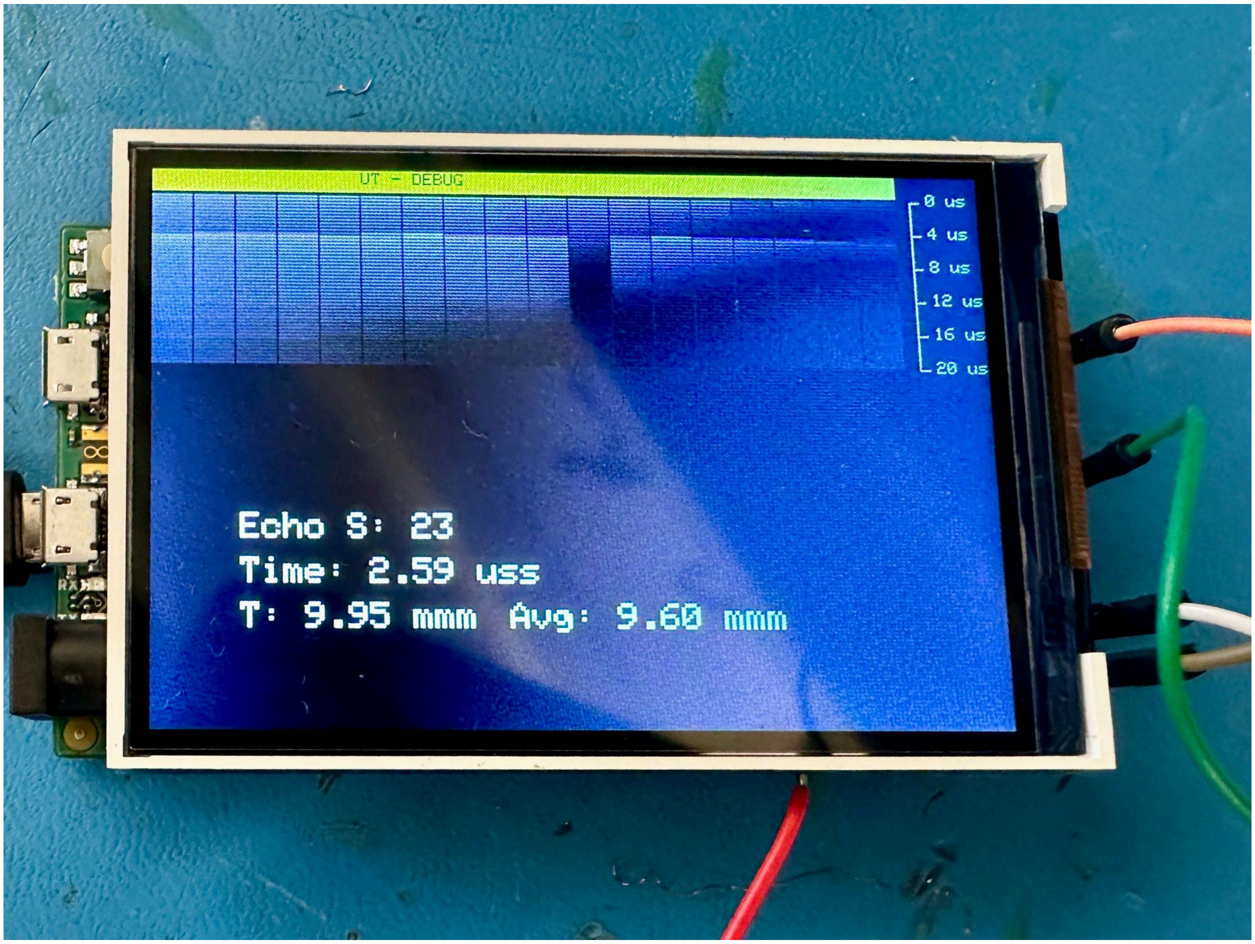
Simplest equation (arbitrating away the math for performance and simplicity):

thickness = 1.45*sample_count + -23.6

From here, we can at least measure something that is 1.05 mm thick, peaking at 17th sample.

6 Advanced Signal Processing and Thickness Measurement Visualization

Measuring an Aluminum 6061 block of thickness 9.8mm



The code was updated after further testing on 8/18/25 to follow the empirical simplified formula:

C/C++
```
#include <Adafruit_GFX.h>
#include <MCUFRIEND_kbv.h>

MCUFRIEND_kbv tft;
```

```
// ----------------- Colors -----------------
#define BLACK   0x0000
#define WHITE   0xFFFF
#define YELLOW  0xFFE0

// ----------------- Pins -----------------
#define pin_output 51
#define pin_input_magnification 41
#define pin_input_sleep 31

// ----------------- Constants -----------------
#define MAX_SAMPLES       100     // Samples per scan
#define RUNNING_AVG_LEN   5       // Number of scans for averaging
#define MIN_ECHO_SAMPLE   16       // Ignore first few samples (ringing)

// ----------------- Globals -----------------
unsigned int values[MAX_SAMPLES];
float thicknessBuffer[RUNNING_AVG_LEN] = {0};
int bufferIndex = 0;
int numFilled = 0;

void setup() {
  Serial.begin(115200);

  // Configure ADC (Arduino Due freerun mode)
  REG_ADC_MR = 0x10380080;
  ADC->ADC_CHER = 0x03;

  // Pin setup
  pinMode(pin_output, OUTPUT);
  pinMode(pin_input_magnification, INPUT);
  pinMode(pin_input_sleep, INPUT);
  digitalWrite(pin_output, LOW);
```

```
  // TFT setup
  uint16_t ID = tft.readID();
  tft.begin(ID);
  tft.setRotation(1);
  tft.fillScreen(BLACK);

  tft.setTextColor(BLACK, YELLOW);
  tft.fillRect(0, 0, 432, 13, YELLOW);
  tft.setCursor(120, 1);
  tft.setTextSize(1);
  tft.print("UT - DEBUG");

  Serial.println("System Ready.");
}

void loop() {
  for (int j = 0; j < 18; j++) {

    // -------- Trigger pulse --------
    digitalWrite(pin_output, HIGH);
    digitalWrite(pin_output, LOW);

    // -------- Acquire samples --------
    for (int i = 0; i < MAX_SAMPLES; i++) {
      while ((ADC->ADC_ISR & 0x03) == 0); // Wait ADC ready
      values[i] = ADC->ADC_CDR[0];
    }

    // -------- Echo detection (first upward spike) --------
    int echoIndex = -1;
    for (int i = MIN_ECHO_SAMPLE; i < MAX_SAMPLES; i++) {
      int delta = values[i] - values[i - 1];
      if (delta > 0) { // upward slope
        echoIndex = i;
```

```
        break;
      }
    }

    // -------- Thickness calculation --------
    float thickness_mm = 0;
    if (echoIndex >= 0) {
      thickness_mm = 1.45 * echoIndex - 23.6;   // Empirical formula relating
thickness and sample count
      if (thickness_mm < 0) thickness_mm = 0;

      Serial.print("Scan ");
      Serial.print(j);
      Serial.print(": Echo at sample ");
      Serial.print(echoIndex);
      Serial.print(", thickness = ");
      Serial.print(thickness_mm, 2);
      Serial.println(" mm");
    } else {
      Serial.print("Scan ");
      Serial.print(j);
      Serial.println(": No echo detected.");
    }

    // -------- Running average --------
    thicknessBuffer[bufferIndex] = thickness_mm;
    bufferIndex = (bufferIndex + 1) % RUNNING_AVG_LEN;
    if (numFilled < RUNNING_AVG_LEN) numFilled++;

    float runningAvg = 0;
    for (int i = 0; i < numFilled; i++) runningAvg += thicknessBuffer[i];
    runningAvg /= numFilled;

    // -------- Display results --------
```

```
tft.setTextColor(WHITE, BLACK);
tft.setTextSize(2);

tft.setCursor(50, 200);
tft.print("Echo S: ");
tft.print(echoIndex);

tft.setCursor(50, 225);
tft.print("T: ");
tft.print(thickness_mm, 2);
tft.print(" mm");

tft.setCursor(200, 250);
tft.print("Avg: ");
tft.print(runningAvg, 2);
tft.print(" mm");

// -------- Timing scale (right side) --------
tft.setTextSize(1);
tft.drawLine(440, 15, 440, 15 + MAX_SAMPLES, WHITE);
for (int y = 15, t = 0; y <= 15 + MAX_SAMPLES; y += 20, t += 4) {
  tft.drawLine(440, y, 445, y, WHITE);
  tft.setCursor(450, y - 5);
  tft.print(t);
  tft.print(" us");
}

// -------- Grayscale scan column --------
for (int i = 0; i < MAX_SAMPLES; i++) {
  uint8_t gray = map(values[i], 0, 4095, 0, 255);
  uint16_t color = tft.color565(gray, gray, gray);
  tft.fillRect(j * 24, 15 + i, 23, 1, color);
}
```

```
    // -------- Sleep mode --------
    if (digitalRead(pin_input_sleep) == HIGH) {
      while (digitalRead(pin_input_sleep) == HIGH) { }
    }

    delay(1000); // Wait before next scan
  }
}
```

Base Functionality: UT_Empirical

- Generates a short trigger pulse on pin_output.
- Captures MAX_SAMPLES ADC readings from the transducer.
- Looks for the first upward slope after MIN_ECHO_SAMPLE to detect the echo.
- Converts echo sample index → thickness with your empirical calibration formula.
- Keeps a running average of the last RUNNING_AVG_LEN measurements.
- Displays raw echo index, thickness, and running average on the TFT.
- Shows a grayscale vertical strip of the waveform (one column per scan) plus a time axis on the right.
- Sleeps if pin_input_sleep is held HIGH.

Further development and testing for more robust and accurate signal processing algorithms:

Observation: We see the signal due to the mechanical link between the transmitter and the receiver. The received signal includes the initial excitation from the transmitter as approximately decaying exponential.
Idea: Baseline subtraction – If we collect the decaying exponential due to the transmit with no object or echo, we can subtract that out from our raw data to get higher SNR and detect even weaker echoes.
Implementation: UT_Empirical_v2

1. Run with no object, press button on pin 41 → reference transmit waveform captured.
2. Place object, run scans → transmit part is subtracted out → echoes stand out more clearly.

Observation: The empirical formula works better, but is harder to adjust than the speed-of-sound based calculation.

Idea: Parameterize the empirical formula based on material to be measured.

Implementation: UT_Empirical_v3

Observation: Difficulties in diagnosing and debugging problems.

Idea: We need more debug features such as reporting state changes, reference recording

Implementation: UT_Empirical_v4

Observation: Debug and nondebug features are together in the code.

Idea: Debug features such as a global variable that keeps track of whether we are debugging, and if so, keeps Serial communication up.

Implementation: UT_Empirical_v5

Observation: We want to record data to better understand and postprocess it.

Idea: Use Arduino serial communication with the computer + Python scripts (pyserial library)

Implementation: UT_Empirical_v5_Record

~~**Observation**: The code has gotten messy with too many new additions.~~

~~**Idea**: Cleanup with a newer version.~~

~~**Implementation**: UT_Empirical_v6~~

~~**Observation**: We want to experiment with new methods to detect peaks.~~

~~**Idea**: New methods to detect peak + dynamic reference + thresholding (does not work well)~~

~~**Implementation**: UT_Empirical_v7~~

Observation: After the previous failure, we want a good demo to reinstate trust.

Idea: Go back to v5, clean up code.

Implementation: UT_Empirical_v5_2

~~**Observation**: We want to experiment with new methods to detect peaks.~~

~~**Idea**: New methods to detect peak + averaged reference~~

~~**Implementation**: UT_Empirical_v8~~

Up to now, v5_2 works best, and was used for our demos at UT_demo

Observation: The reference collection includes the initial peak, but it should more be like an exponential decay.
Idea: Force an exponential-decay-shaped reference to be subtracted from any raw capture. Do later processing to find peaks (probably ignore the first peak).
Implementation: UT_Empirical_v5_3

This new version is also promising.

7 Future – Ultrasonic Sensor Integration at Scale

We can integrate the transmit and receive circuits in a 4-layer PCB in an area smaller than 3x3cm. This should allow for movements within pipes.

The most important area of improvement would be to use a much faster ADC to sample more (and more sensitively to the phase of the signal). This would also be needed for the full-system integration for the vine robot.

In general, we need:

ADC sampling rate > 2 * speed in material (for the fastest material to be used) / smallest measurement needed

We tested with rolled aluminum (speed of sound is 6420 m/s) because this seems to be the pipe material with the fastest speed of sound.

so, for the future, we would want:

ADC sampling rate > 2* (6420 m/s / [0.1 mm]) = 128,400,000 Hz = 128.4 MHz

Replace [0.1 mm] as needed with the smallest feature to be recognized.

Similarly, the pulse needs to be thinner for less uncertainty as to where the peak is, and it needs to be higher voltage (and larger crystal) for a deeper scan.

In our case, we had:

Resolution = 2 * (6420 m/s / ADC sampling rate [5 (or even 1) MHz is the theoretical maximum for Arduino Due]) = 0.002568 m = 2.6 mm, which seems accurate to our smallest decipherable features.

Some subsystem requirements for integration into the broader robotic system:

- +10V and –10V power needs to be supplied.
- High voltage (>100V) needs to be supplied.
- The transducer probe needs to be applied perpendicularly with about 2.5N of force

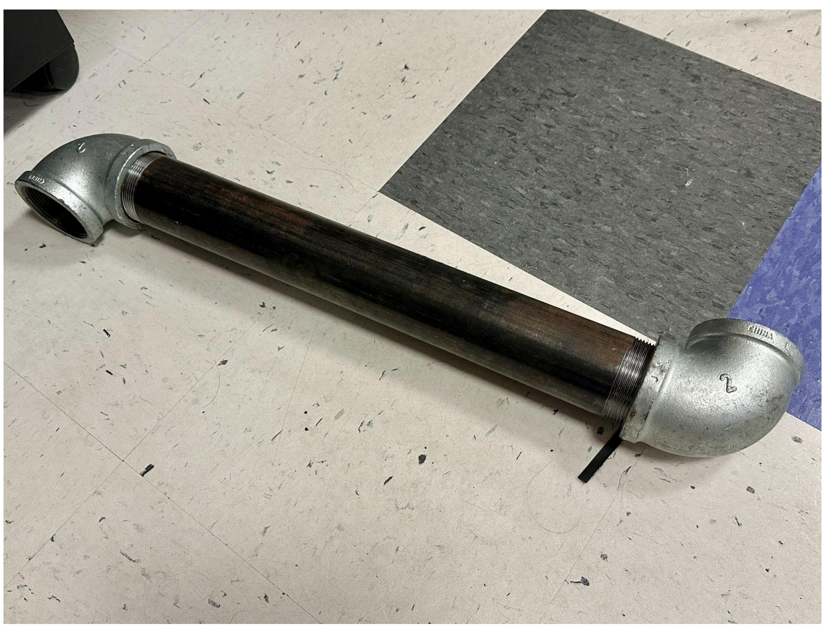
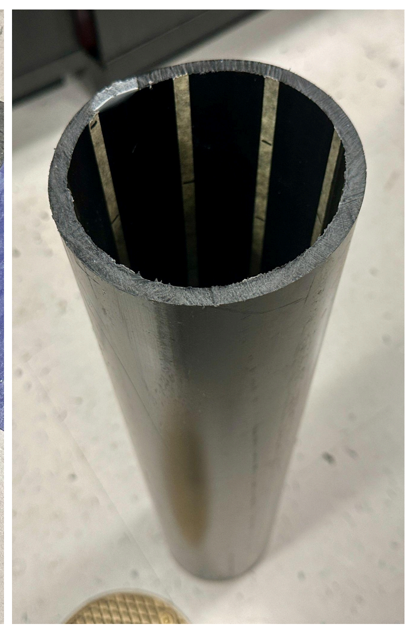
- There needs to be ultrasound gel (or in the future an ultrasound pad) applied to the surface
- No observed thermal requirements. The system was near room temperature (<30 •C) even with power connected for hours.
- The system draws the following power:
    - +10V × 75 mA (the additional 25 mA is due to the 100V regulator and spike generation)
    - –10V × 50 mA
        - Total power need: 1.25W
        - In-pipe Vine Robot currently has a 7.4V power supply,

          so we would draw 1.25W / 7.4V = 169 mA

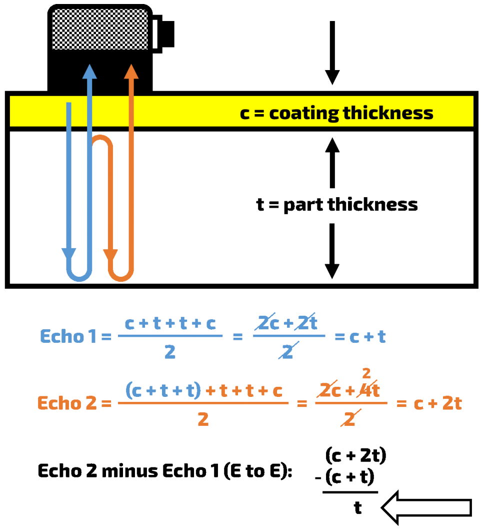
Otherwise, the vine robot NDT team can collaborate with a company such as Cygnus or Evident/Olympus to create a smaller solution with them for this application.

We started the PCB project here, adjacent to all the code:
https://github.com/ege-turan/ultrasound-thickness-gauge

PVC and stainless steel pipes also showed peaks, though further testing is necessary for finding out parameters that consistently estimate thickness from the echo sample index.

We can also implement coating/paint rejection with the following approach (Echo-to-Echo Measurement):



8 Conclusions

Demos of our system can be found here: UT_demo

UT technology has:

- NDT industry's trust
- Trade-offs including:
  - Single vs dual element crystal
  - Crystal size selection
  - Operation frequency
  - Voltage at transmit
  - Multiple avenues for integration into the vine robot
- Considerations including:
  - Impedance matching (acoustic)
  - Impedance matching (electrical)
  - Phased arrays costs and benefits
  - Power consumption (~1.3W)
- **Multiple avenues for integration into the vine robot, and this research project presents knowledge, experience, and a functional circuit that can be miniaturized and used in tests.**