

# **Group 15-22**

## **Ultrasound Imaging System**

### **Project members**

**Advisor: Timothy Bigelow**  
bigelow@iastate.edu

**Aaron Tainter**  
atainter@iastate.edu

**Weikun Han**  
kunhan@iastate.edu

**Haoyu Wang**  
haoyu@iastate.edu

**Jingyu Xie**  
bbsxjy@iastate.edu

**Table of Contents:**

Project Definition-----	3
System requirements-----	3
Functional Decomposition-----	4
System Analysis-----	4
Block Diagram-----	5
I/O Specification-----	5
Interface specifications-----	6
Hardware/software specifications-----	7
Simulations and Modeling-----	7
Implementation Issues and Challenges-----	7
Testing, Procedures and Specifications-----	7
PCB Issues-----	8
Work Done-----	8
Conclusion-----	9

## Project Definition

MRI imaging is very expensive to run and maintain. Our project will provide a cheap alternative to fMRI imaging of the brain. Our client wants us to develop a working prototype that can be used for demonstration purposes.

In this project, our system will consist of software that configures and deploys beams from a beamformer-transmitting device. The software will also acquire and analyze echo signals gathered by a receiving device. This data will then be processed to generate greyscale images that can be used to diagnose brain injuries (as an alternative to fMRI). When we are finished, we will evaluate the accuracy and performance of the software. If the performance of the system is adequate, we will make it function in real-time.

## System requirements

### 1. Transmission System

- Communication with the beamformer
- Beamformer to transducer communication
- Programming wave output via serial connection
- Generates correct pulse waves
  - Over 8 channels (scalable to 512)
  - Ability to adjust pulse width modulation.

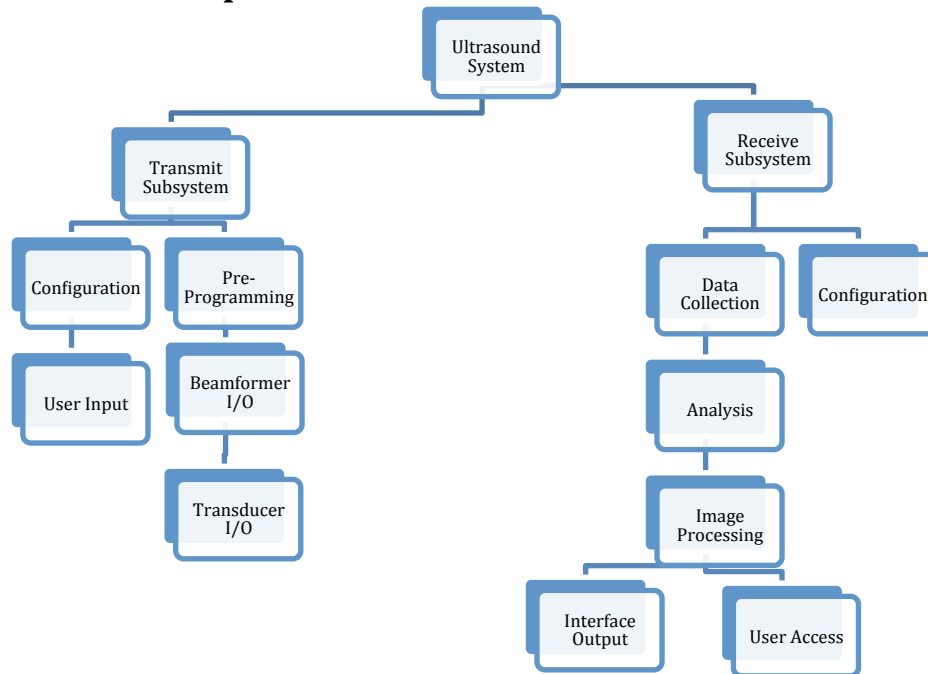
### 2. Receiver

- Receive amplified and filtered signal from the receiver circuit.
- NI 5752 card 8 bit input
  - Contains a variable input ADC

### 3. Analysis

- Stream data continuously on input
- Process 8 bit input
  - Develop an algorithm that generates images based on collected data
  - Process realtime (30fps)

## Functional Decomposition



## System Analysis

### Overview

This system will provide a quick and easy way to produce images of the brain. Due to the fact that this system will not be commercialized, more emphasis is placed on the performance and system accuracy rather than usability.

### Transmitter

The configuration of each element will be controlled separately within the same UI. The beamformer will be controlled by a subsystem within the labVIEW program. It will contain default runtime settings that can be reconfigured for amplitude and pulse spacing by the user.

### Receiver

The beamformer will be programmed via serial data. Once programmed, the beam former will continue emitting pulses until told (via the program) to terminate. Signals from the beamformer will make the transmitter/transducer produce ultrasound pulses. The transducer will receive reflected signals and then amplify them to be sent back to the system. NI 5725 DAQ cards will receive signals from the receiver.

### Analysis

Received data will be analyzed and an algorithm will be used to interpolate data so that we can produce images from the received signals. This subsystem will execute when the NI 5752 DAQ receives data.

The UI will also include elements that allow users to adjust the sensitivity of the receiver as well as the high and low bounds for data during input. This will allow images to have different levels of contrast. The system will also allow user to save data or view it in real time.

## Block Diagram

Below is the block diagram of connected components in our system:

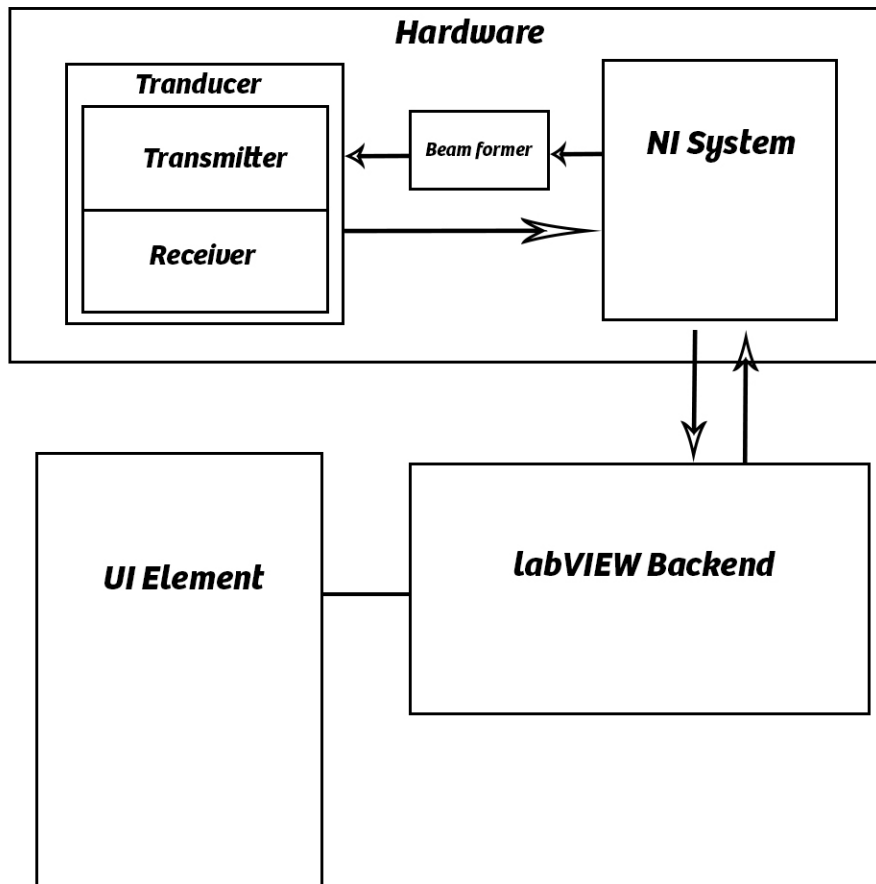


Figure 1

## I/O Specification

### Transmission system

Input

- Beamformer Amplitude signal configuration (-/+ 50v)
- Channels (8 – 512 scalable)
- Variable gain for differing imaging depths (total 70 db)
- Frequency (1-15MHz)

- Switching delay time (32-167ns)
- Pulse shape
- On/Off

#### Output

- Beamformer serial program.
- System initialization
- System termination

### Receiver system

#### Input

- Data sensitivity (Max voltage input 2Vpp)
- Realtime vs Offline Imaging
- Image Contrast adjustments
- Data storage options

#### Output

- Raw signal data
- Interpolation algorithm
- Image file output
- Image realtime output

### Interface specifications

Due to the fact that this project is programmed entirely in labVIEW, there are no true Interface methods. The only interface used will be a GUI element created in labVIEW. The GUI must include controls that conform to the I/O requirements. Those controls are described below.

#### Inputs

**UI Switch:** on/off

**Numeric Controls:** Amplitude, Channels, Variable Gain, Frequency, Delay Time, Data sensitivity

**Dropdown Controls:** Realtime/Offline Imaging, Data Storage, Pulse Shape

**Slider:** Image Contrast

#### Outputs

**Image window:** Image Realtime Output

**File Output:** Image Offline Output, Raw Data

## Serial Data: Programming and Serial Output

### Hardware/Software Specifications

Our system must connect with NI ultrasound hardware. In order to do this correctly, our group must read documentation files in order to initialize and program it correctly. The hardware is interfaced with through NI 5752 cable I/O. In order to use this, the correct drivers must be installed on the labVIEW system.

Additionally, NI hardware operates on specific voltage regulations. If the voltage input or impedance is out of specification, it might destroy the system. We must be careful to use the correct voltages during output. Also, because the parts are so sensitive, we must also use proper safety precautions when handling parts to avoid short circuits or static discharge.

### Simulations and Modeling

Simulations will be created with the function generator. We will test uniform signals to prove the correctness of our interpolation algorithm. We will also test our product on inanimate objects for the same proof.

### Implementation Issues and Challenges

#### Issues

1. Proper settings and control programs
2. Hardware handling methods
3. Correct interpolation algorithm for raw data

#### Challenges

1. Performance
2. Image manipulation
3. Data translation

These issues are important because they are functional requirements for our system. Without these elements performing correctly, our whole system would fail. In order to get them working properly, we will use the following testing procedure. Performance will also be measured through labVIEW controls and timing. We may have to embed some C code in order to get the performance that we need.

### Testing, Procedures and Specifications

We will use a circular development pattern in order to produce the product. This will require us to test, then make improvements and redevelop. The procedure is shown as following:

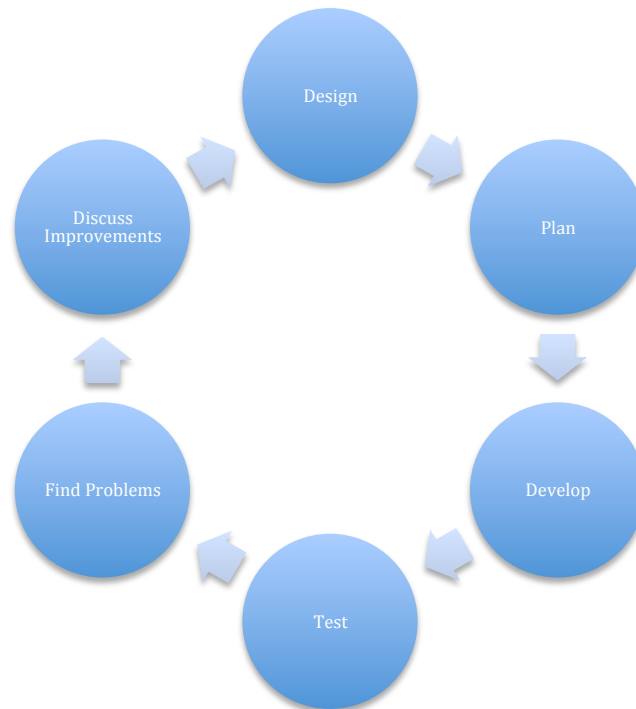


Figure 2

Testing for this project is more difficult because it is not like typical software development with unit testing (labVIEW). We will create a PCB test board that will be used to analyze signals with the oscilloscope and send test input with the function generator. Additionally, we will test the Image interpolation algorithm by using the system on objects of known shape. Additionally, we will be able to save output to files and analyze data collection manually.

## PCB Issues

The PCB used for testing is a very simple adapter component with no extra circuitry. When using the board, we must be careful that system input does not exceed 2Vpp input on the NI DAQ or else we could damage parts.

## Work Done

So far, we have had major issues with configuring the NI system for use with the NI 5752 cards. The system was supposed to be functional when we got it, but it was not. We spent several hours troubleshooting on our own and with engineers at TI. Once we got it fixed we developed some test programs to input and output on the NI cards. Time was spent getting the cards to work in parallel and on multiple channels. Before we got the NI system working, we also developed a labVIEW signal analysis program that can analyze and output input frequencies. This will be integrated into our program later.



We also developed two PCB test boards to work with the NI cards. We submitted those and are waiting for the manufacturer to send our designs back. Once get those, we can test the labVIEW programs.

## **Conclusion**

By the end of this project, our goal is to create the software system that interfaces with custom signal generation hardware provided by National Instruments. Ultrasound Pulse signals will be transmitted and received by the system and controlled by our software. Received data is analyzed and converted into imagery that can be viewed by the user. The system will include UI elements that allow the user to manipulate system settings.