EE/CprE/SE 491
wDAQ System (sddec24-19)
Design Document: Design

April 2, 2024
Client: Manojit Pramanik and Avishek Das
Faculty Advisor: Manojit Pramanik

TEAM MEMBERS

Adam Shoberg [EE] - Circuit Design & Simulation, PCB Design, Team Communications Leader

Henry Chamberlain [EE] - PCB Design & Construction

Lisa Tordai [SE] - Software Development, Wireless Data Sharing

Vaughn Miller [CprE] - Computer Engineering

4.2.1. DESIGN DECISIONS

- **Two-stage, single supply voltage low-noise RF amplifier:** Per the decision specifications, the device needs to amplify an input signal from 1mV to 1 volt. This required two amplifier ICs, MAR-6SM+, to be cascaded together to achieve the gain required. The device also needs to be biased by a single voltage supply to minimize complexity and size.

- **High-speed Microcontroller Unit and separate ADC:** Due to the bandwidth requirement, the sample rate of the device needs to be at least 15 Megasamples per second or higher. An FPGA and ADC would be the best choice if time and money weren't a concern, but a microcontroller-based solution is more familiar and far less expensive, and modern offerings from ST Microelectronics are capable of handling data processing at high speed. The ADC also needs to be as noise free as possible at the frequencies we're working with, so we've chosen to employ a separate ADC IC in series with an STM32 MCU.

- **Wireless connection of device to user software:** We have decided to go with a Wi-Fi connection (instead of Bluetooth) because Wi-Fi chips allow more data to be transmitted. This allowed the team to narrow down the research of a suitable module. The module of choice ended up being the ESP32 due to its ease of use, low cost, performance matching, and compatibility with other technologies being used in the project.

- **EasyEDA for PCB Design:** One of the major components of this project is the design of our own PCBs. In the end, we decided to go with EasyEDA for a multitude of reasons, the first being its large library of parts. This minimizes the need for third-party websites to import footprints and other files. The second reason for choosing EasyEDA was its file sharing capabilities. The program allows for team folders, where members can work remotely on the same file. This maximized team collaboration and saved man hours. The last reason EasyEDA was selected for the PCB design was because it was easy to buy the components and PCB directly from the program.

- **Rechargeable battery power supply:** We will be utilizing a rechargeable battery to power our device alongside a battery management system to protect the battery from overcurrent, designed partially by our student team and partially by our client. The choice of battery comes down to three potential options: Lithium-ion, Nickel Cadmium (NiCd), and Nickel-Metal Hydride (NIMH), all of which are compact in size and can be compatible with integrated USB Type-C charging. We will either be designing the battery management system (BMS) or using a preassembled chip, but a study on the power usage of the circuit and power management optimization, which will be completed after designing a prototype of the device (without the battery), will enable us to determine the requirements of the battery and associated BMS. We will also be implementing a voltage regulator (such as TI's LM7805 Linear Regulator IC) with the battery's power supply that will ensure we supply steady DC voltages to the circuits within our design and avoid accidental overvoltage or overcurrent.

4.2.2. IDEATION

For each of the key design decisions, our team created a Lotus Blossom Diagram to identify potential options and solutions. The wireless communication aspect of the design process was one of the few areas where our team had more leniency and "options" to choose from in the design, rather than needing to use a specific technology or device to achieve a requirement. The section of the Lotus Blossom Diagram where we identified possible options and design considerations relating to wireless communication is shown in *Figure 1* below.

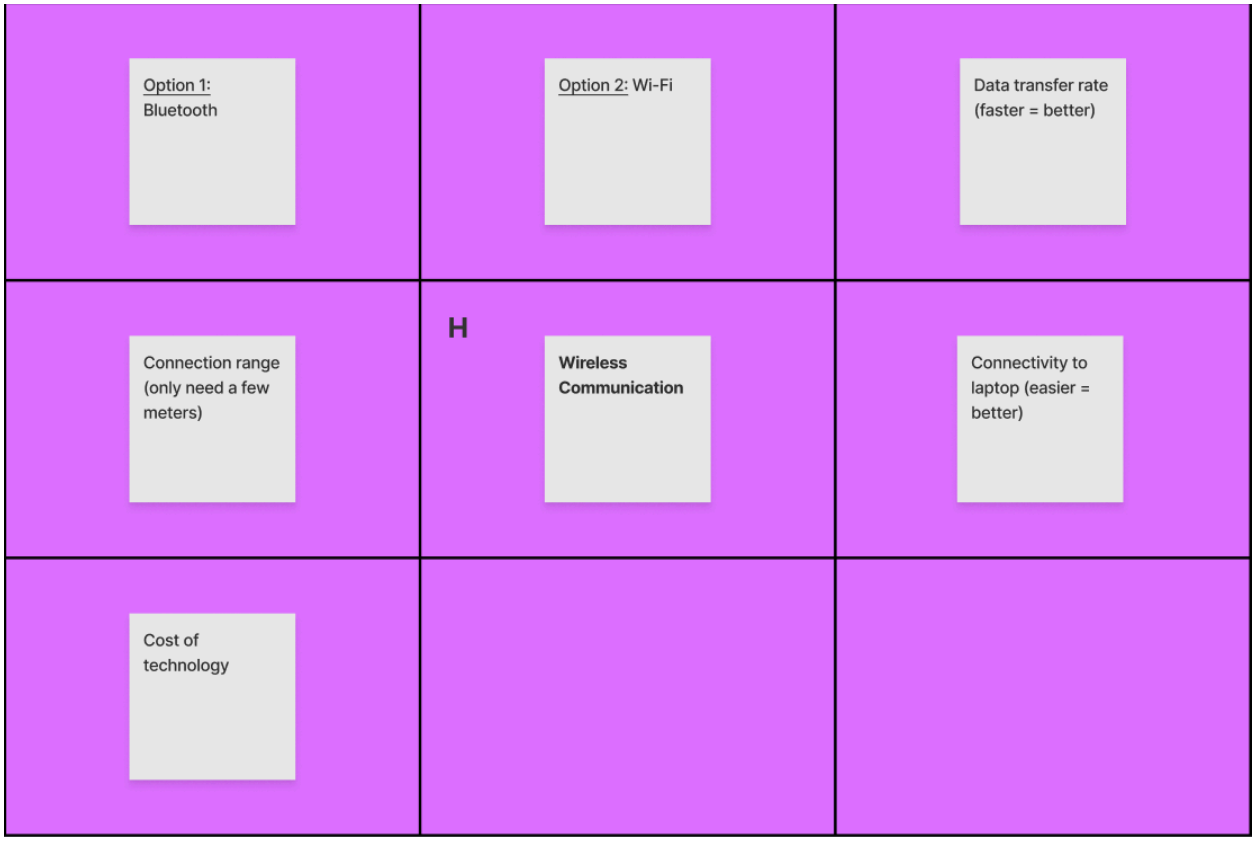

Figure 1: Lotus Blossom for Wireless Communication

The options, constraints, and considerations our team came up with for wireless communication using the Lotus Blossom technique are described in detail below.

- **Bluetooth (Option 1):** Bluetooth communication chips have relatively high latencies for wireless communication (30+ ms) and somewhat slow data transfer rates (typically no more than 30 Mbps (megabits per second)). The connection range is also fairly small (typically 10-20 meters or less). The advantages of Bluetooth include mostly reliable & consistent connectivity and secure data transfer.

- **Wi-Fi (Option 2):** Wi-Fi communication tends to have a lower latency for wireless communication (often less than 10 ms) and a much higher data transfer rate than Bluetooth (over 100 Mbps), which enables high-speed data collection at all times. The connection range of Wi-Fi chips is also significantly higher than BT chips (~100 meters). Wi-Fi connection is also usually consistent and is highly secure.

- **Data Transfer Rate:** The data transfer rate of the device is an important factor in the decision for wireless communication, because a high data transfer rate ensures fast communication with software and compatibility with the sampling rate of the ADC and clock speed of the MCU. Our ADC of choice (ADC12020 from Texas Instruments) has a sampling rate of at least 20 MS/s (megasamples per second), and the MCU (microcontroller) has a clock speed of over 100 MHz, so the technology we use will need a data transfer rate that matches or surpasses these rates to ensure compatibility. This makes Wi-Fi a much more attractive choice already.

- **Connection Range:** The connection range of our wireless communication is not especially important, because the user will most likely be utilizing the LabVIEW software in close proximity to the device (i.e., within a few meters). However, having more flexibility with the distance between the user and the device is always a plus. Wi-Fi chips offer a connection range in excess of 100 meters, while most Bluetooth chips are limited to a connection range of a few (up to 10) meters, which makes Wi-Fi a better option again.

- **Ease of Laptop Connectivity:** Both technologies are relatively easy to connect to once they have been properly configured. Initial tests of both of the technologies have shown some connection difficulties and delays, but they typically resolve quickly. From a user perspective, Wi-Fi networks are generally a bit easier and faster to connect with than Bluetooth devices, and the use of a security key makes them a safer data transfer choice.

- **Cost of Technology:** Most marketplace Bluetooth and Wi-Fi chips are relatively inexpensive (in the range of $10 to $20), so there is not a significant discrepancy between the costs of the two technologies, which makes this a fairly unimportant factor in the decision process.

4.2.3. DECISION-MAKING AND TRADE-OFF

For the decision-making and ideation process, our team used a weighted decision matrix to identify the pros and cons of each of the potential options for wireless communications. We selected this option for ideation because it enables us to easily understand the different advantages, disadvantages, and characteristics of Wi-Fi and Bluetooth technology, as well as make more direct comparisons between the two technologies to make a well-informed decision. The process of a weighted decision matrix involves identifying the relevant considerations and factors associated with our two options, assigning a “weight” (importance factor) from 1 to 5 for each factor, and assigning a performance score for each technology and consideration, also from 1 to 5, with an explanation of how well each technology satisfies each of the criteria.

Table 1: Weighted matrix used to evaluate Wi-Fi and Bluetooth options

Factor	Weight	Wifi	Bluetooth
Data Transfer rate	**5**	**5** - Data transfer rate often exceeds 100 Mbps and chips have low data communication latency	**2** - Data transfer is rarely above 30 Mbps and some latency exists in data communication
Connection Range	**2**	**4** - Connection range is large (in excess of 100 meters)	**2** - Connection range is limited (typ. 10-30 meters)
Ease of Laptop connectivity	**3**	**4** - Upon configuration of IP addresses it is as simple as connecting to your home Wifi	**4** - After configuration, connection is quick & easy
Cost of Technology	**2**	**3** - Cost of chips range from $10-$20	**3** - Cost of chips range from $10-$20
		Total Weight: 51	**Total Weight: 32**

Note: Based on the weighted matrix, *Table 1,* it is evident that the use of a Wi-Fi chip is the way to go for our project. It scored significantly higher in the most heavily weighted factor, data transfer rate, while staying on par with Bluetooth for every other factor.

4.3.1. DESIGN OVERVIEW

A high level view of our design is a low-cost, user-friendly, wireless oscilloscope (tool used for viewing properties of signals). The design starts off with a low-strength, high-frequency signal being captured and sent to an amplification and filtration stage. The amplification stage essentially takes this initial signal and scales it by a factor of close to 1000. The purpose of the filtration stage is to remove unwanted components of the signal. Similar to how a coffee filter works, it removes the unwanted portions of the signal (the “coffee grounds”) and passes the portions we do want (the coffee). Following this, the filtered and strengthened signal travels to the analog-to-digital conversion stage. This portion of the device uses components that essentially follow the original signal, but in the form of steps (like staircases). Reference *Figure 2* below for a visual representation of the input-output relationship for an ADC.

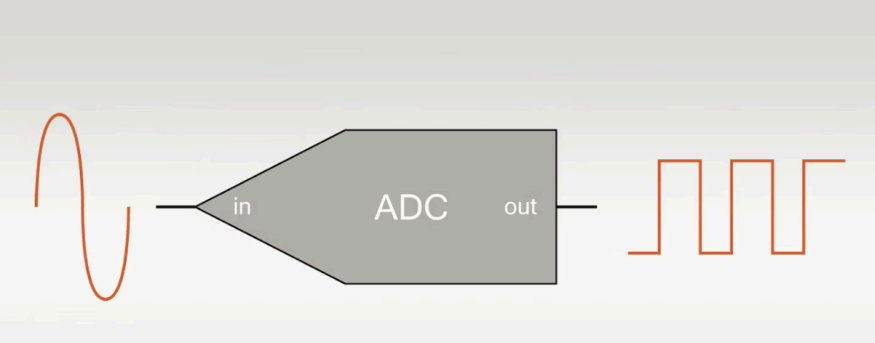

Figure 2: Input/output relationship for an analog to digital converter [1]

Once the signal has been digitized, computers can perform actions with the signal. Following the signal's digitization, it travels to a component called a microcontroller. This is the jack of all trades for electronics components. It has the ability to perform actions based on user-determined instructions using a component called a central processing unit (CPU). It also has storage space to hold onto information pertaining to the digitized signal. Once the microcontroller has executed its user instructions, it spits the information out of the output pins. The last stage of hardware the information travels to is the Wi-Fi module. Similar to how you connect to your Wi-Fi at home, a user is able to connect to the Wi-Fi module on their computer. This is where the information is transferred wirelessly to the computer for the user to see on their computer or mobile device.

4.3.2. DETAILED DESIGN AND VISUAL(S)

The design of our device consists of six different interconnected stages that operate in close succession (the first of which is technically not part of the device). A technical description of each of these stages, followed by a block diagram showing images of the individual components and the order of their connections, are given below.

- **Stage 1: Photoacoustic Transducer:** A Photoacoustic Transducer will provide the input signal to each DAQ system that is implemented within the Photoacoustic Tomography (PAT) system, connected by male and female SMA ports. These transducers take physical signals from the real world (in our application, sound waves and echoes from animals) and transmit them to the DAQ devices as electrical signals that can be operated on and analyzed. The transducers have already been designed and implemented within the PAT system, so our team will only be concerned with connecting them to our devices.

- **Stage 2: Low-Noise RF Amplifier & Filter:** The low-noise amplifier takes a 1 mV (peak-to-peak) analog input signal and amplifies it to an amplitude of approximately 1 V (peak-to-peak). Thus, the overall amplifier has a gain of close to 1000 V/V, or 60 dB on a logarithmic scale, which is implemented by cascading two amplifiers with individual gains of approximately 32 V/V (30 dB). We chose the MAR-6SM+ amplifier by Mini Circuits for our design. The amplifier is considered “low-noise” because it does not distort the output signal or significantly degrade the signal-to-noise ratio: the noise figure of the amplifier is 2 dB at 300 MHz. Another important characteristic of the amplifier circuit is wide bandwidth: because we want to amplify radio frequency signals per the project proposal, we will require an amplifier bandwidth in the MHz range, which the MAR-6SM+ amplifier offers (with a wideband frequency range from DC to 2 GHz). The operating bandwidth of our overall amplifier circuit is limited to the range of 100 kHz to 2 GHz through the use of a simple RC network and current-limiting inductor at the output of each amplifier. The other notable features of the amplification & filtration circuit are: a 50 Ω input impedance (which determines the necessary impedance of the signal output by the transducer); a current draw of at most 70 mA corresponding to a power dissipation of at most 864 mW across the circuit (achieved after the complete prototype of the device with a power management & optimization study); and lastly, a single-rail 12 V DC power supply, which will ultimately be sourced from the battery and fed through a voltage regulator (LM7805 from Texas Instruments) to ensure consistent voltage. *Figure 3* below shows the complete schematic for the low-noise amplifier. From left to right, the schematic shows the RF input (through an SMA port), the two MAR-6SM+ amplifiers cascaded together (rectangular symbols), the RLC biasing networks used to control the gain and bandwidth of the amplifier, and the RF output (input to the ADC circuit). The small box with arrows coming out of it on the upper left shows the pins for the 12 V DC supply and the common ground.

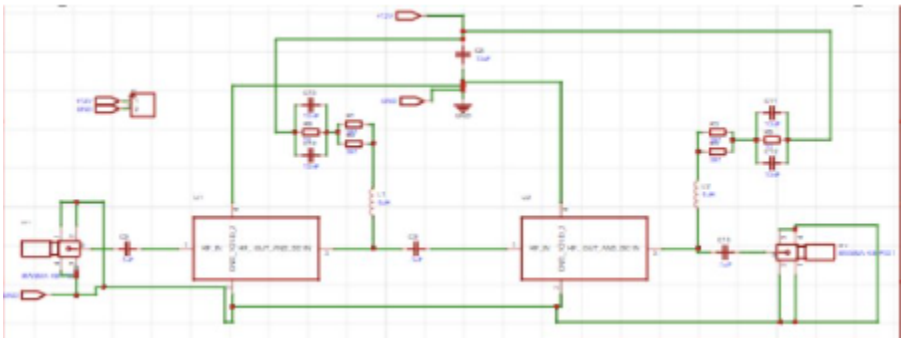

Figure 3: Schematic of the Low-Noise Amplifier & Filtering Circuit

- **Stage 3: Analog-to-Digital Converter (ADC):** The ADC circuit converts our amplified analog signal to a digital signal, i.e., a signal that takes on only discrete amplitude values (like a square wave) rather than a continuous spectrum (like a sine wave). The ADC chip we are using, the TI ADC12020, has 12 bits of resolution and operates at a sampling rate of 20 MS/s (megasamples per second) or more using an internal sample-and-hold circuit to minimize power consumption. The circuit utilizes a differential pipeline architecture, so it takes both a positive and negative-ended analog input for digitization. The differential inputs are produced from the output signal of the low-noise amplifier circuit through the use of a differential driving circuit that incorporates an instrumentation amplifier, a general-purpose op amp, a separate power supply circuit containing a precision bandgap shunt voltage reference (used to reduce the voltage from 12 V to 5 V DC (note that the LM7805 IC from Texas Instruments may be used in place of this voltage reference), and a multitude of resistors and capacitors that maintain consistent signal amplitudes and characteristics between the input and output of the instrumentation and general-purpose amplifiers. A schematic for the ADC circuit is shown below in *Figure 4*. In the middle of the schematic is the differential drive circuit, which takes the output of the low-noise amplifier (denoted by the SMA port symbol) and converts it to the differential ADC input. The schematic mainly uses nets (the arrow symbols) to minimize the use of wires, so it is difficult to see some of the connections, but the ADC is powered by the power supply circuits on top of the schematic and outputs to the digital output pins on the left side.

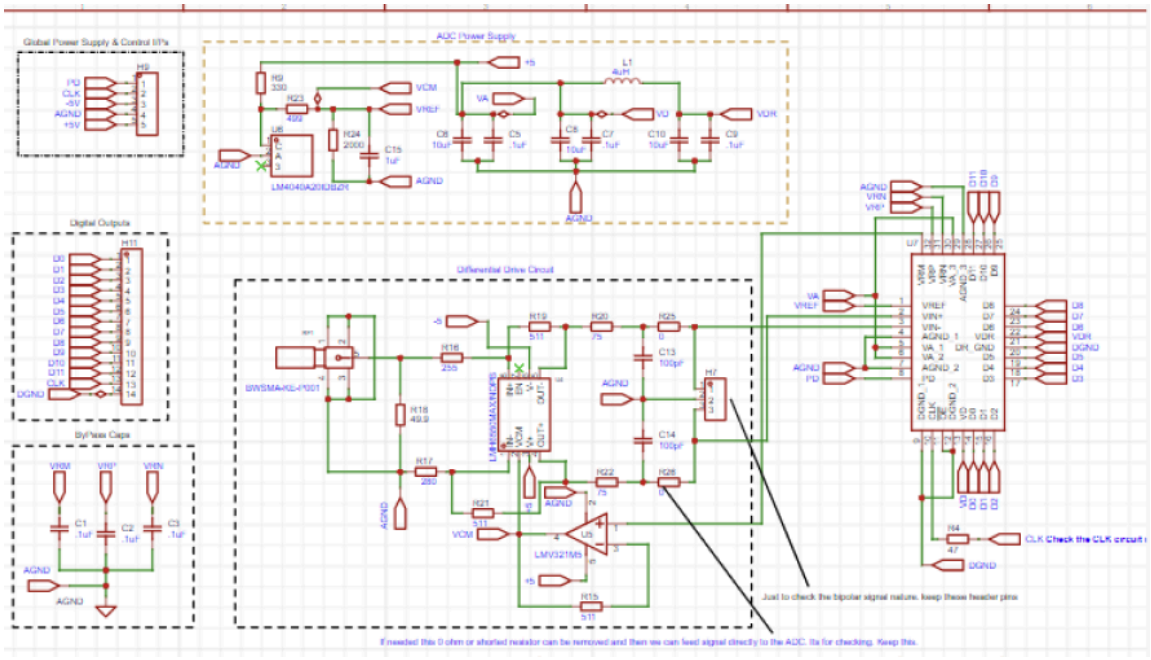

Figure 4: Schematic of the ADC and Differential Driving Circuit

- **Stage 4: STM32F4 Microcontroller:** The MCU, an STM32F407, handles the synchronization of the ADC output, the reception of the ADC data, and the packaging of that data into neatly arranged messages that the ESP32 can receive and send over the air to the user's computer. The ARM Cortex M4 processor that this microcontroller is equipped with is a very capable device, running at nearly 200 Mhz versus the typical 16 Mhz that AVR-based boards run at, with a more comprehensive and cycle-efficient instruction set. It also contains a wide array of peripherals and features for development use, but our team is mainly interested in simply receiving parallel inputs, using the Direct Memory Access (DMA) module to automatically move inputs into memory, then writing SPI messages to the ESP32 as quickly as we receive the data. The STM32 will also output a clock signal to indicate to the ADC how fast it should send its parallel bits of digitized signal data for processing by the STM.

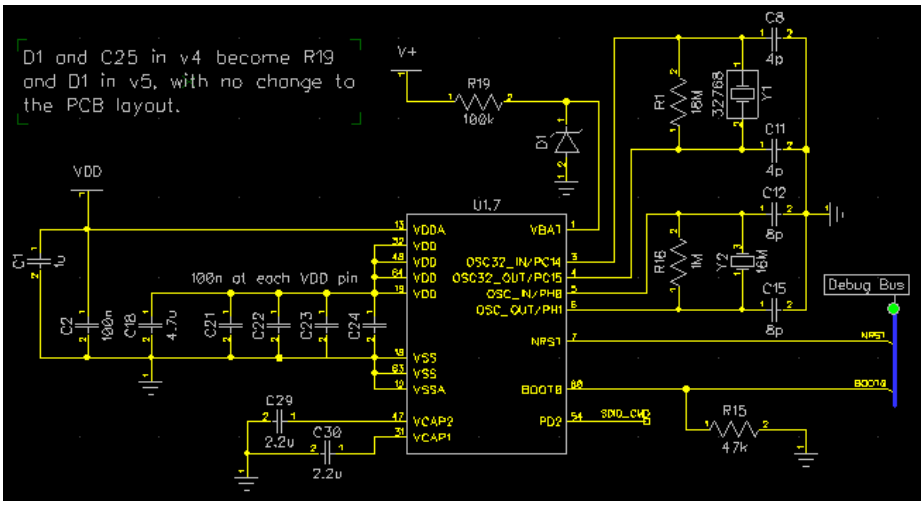

Figure 5: Minimal STM circuit for pin-compatible F4 MCU

- **Stage 5: ESP Wi-Fi Module:** For our wireless communication, we will be utilizing an ESP32 with our STM32F4 Microcontroller. The microcontroller will package messages and provide them to the ESP32 which will then send them as a server in six byte messages. The image below shows the physical Wi-Fi module that we will be adding to our device.

Figure 6: ESP32 Wi-Fi Module

With this module, a user will be able to see the device pop up on their laptop Wi-Fi setting and will be able to connect. The LabVIEW interface will provide the IP address and port of the device which will make the connection in the GUI. Once the device is connected to the laptop and GUI then messages are able to be read that the ESP32 is constantly sending over the server.

- **Stage 6: User Interface:** The GUI will be a very important part for the user experience. The goal for our user interface is to provide data to the user with minimal latency, provide a useful interface for data analysis, and allow the user to record data. LabVIEW will allow us to connect to the device using a Transmission Control Protocol for network communication. For this we will need to provide the IP address and port which we can set/ discover using the Arduino IDE ESP32 Library (this should only need to be done one time). Once the IP address is known and the port is set we can configure a TCP block in LabVIEW. LabVIEW will also provide a graph for data and a data logging option to a text file which are both capabilities of LabVIEW software.

Lastly, the flowchart and block diagram that visually depicts the process flow of the six stages of our device from start to finish is shown below in *Figure 5*.

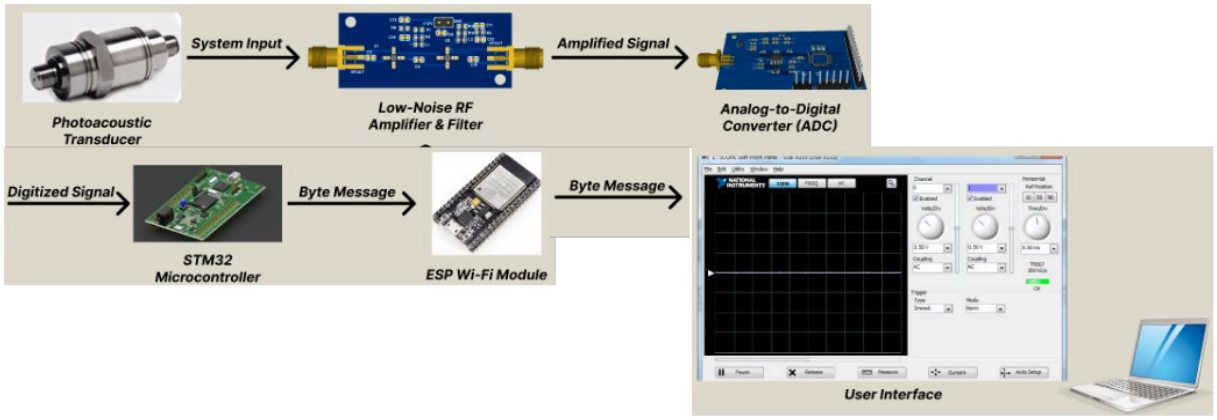

Figure 7: Block Diagram of the Process Flow (Split into Two Rows)

4.3.3. FUNCTIONALITY

Our device is intended to be used at the ISU biomedical imaging lab (BILab) with a Photoacoustic Tomography (PAT) system that conducts experiments on small animals. Reference *Figure 3* for visual depiction of the PAT system. The wDAQ will be used to capture high frequency (MHz range) signals, created from a device called a transducer that converts physical signals to electrical signals, and wirelessly transmit the data in real time to the users' computers. There will be two channels on the device: one to trigger and another for signal acquisition. The user will be

able to view the data through an application called LabVIEW with the support of an easy-to-connect Wifi module. LabVIEW offers users the ability to completely control the information they wish to view. For instance, they will have the option to modify axis scales and select which window(s) they are viewing.

Upon successful completion of the first device, a batch of 10 to 20 copies of the device will be produced and connected to an array of transducers in a circular configuration within the rotating PAT system. The primary users of this device will be lab technicians and graduate students that are familiar with technical systems within the BILab. Although the primary user will have experience with sophisticated systems, the device will be designed to cater to users of all levels of experience and expertise.

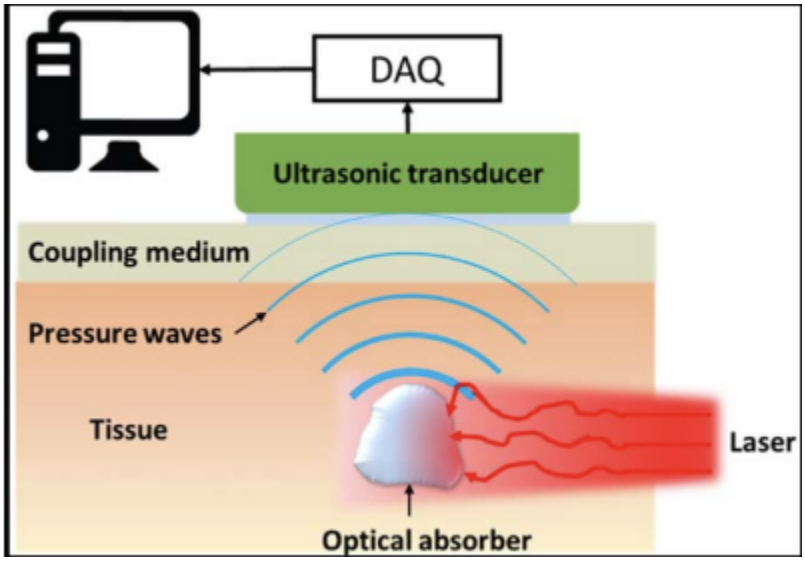

Figure 8: Visual demonstration of how the PAT system operates [2]

4.3.4. AREAS OF CONCERN AND DEVELOPMENT

Our most recent iteration of the design for our device satisfies the majority of user needs and requirements. The Low-Noise Amplifier & Filter (LNA), the Analog-to-Digital Converter (ADC), the STM32F411 microcontroller, the ESP32 Wi-Fi Module, and LabVIEW user interface have all been designed to a functional level so far. Additionally, the PCBs for the LNA and ADC have been designed and we have ordered PCBs to be soldered for these parts of the design. Once we have completed testing the boards and software for each individual portion of our design, we are aiming to combine the individual components of the design in late April/early May to create a functional prototype, which will be tested and then revised in the Fall semester.

As we work on creating future iterations of the design and eventually our final product, we will want to be sure we are placing a focus on compact sizing, clean traces for the PCB, a strong wireless connection between the device and user software, and meeting all user needs and requirements given in the project proposal. Our device is relying on the STM32F411, ESP32, and GUI to communicate a lot of data quickly, so for this to work with minimum latency, our programming needs to be as efficient as possible. Our team will also need to test our device and interface to inspect its capabilities within different environments and ensure the data being presented is accurate.

The primary concerns our team has right now related to delivering a prototype and final product that addresses all requirements are that, first of all, some of the traces in our PCBs are somewhat sloppy and messy, leading to certain PCB designs being overcrowded with wires and confusing connections when they potentially don't need to be and the final design can be made cleaner. Additionally, the initial tests of the Wi-Fi module have indicated some unanticipated latency with the data transfer process, as well as some unexpected difficulties with connecting the device to Wi-Fi initially, which may or may not require further testing and steps to resolve. On the whole, we are all slightly concerned about being able to fully design, order, and test a complete prototype for our device before the end of the semester in May, because we only have about a month to complete our work and combine our individual designs into one.

Our immediate plans for addressing these concerns and solving these problems is consistent communication and more frequent team meetings from now through the end of the Spring semester. We are facing issues with certain portions of the design not being completed in a timely manner, while others are being completed ahead of schedule, so the best way to resolve these discrepancies and ensure we are working at the same pace is by meeting and communicating more frequently. By sticking to this slightly more rigid schedule and committing to better communication for the rest of the semester, we can set ourselves up for success in completing the design and buildout of our first functional prototype by May.

4.4. TECHNOLOGY CONSIDERATIONS

With the overall complexity of our project, we have been required to use innovative technologies to accomplish our goals and requirements. The first technology we have worked with is a free to use program called EasyEDA. This is a program that allows for its users to build circuit schematics from their wide range of libraries, and export them to PCB layouts. EasyEDA also offers functions like auto-route to route traces in the board all while minimizing the distance and amount of turns. EasyEDA also offers a unique way for teams to collaborate on projects through their full-online version.This has allowed for everyone to work, analyze, and review on a subsystem. Once a project has been completed, EasyEDA offers easy to export bill-of-materials

and files that make sourcing the parts and boards easier than ever. This paired with its built-in functions and libraries have saved countless hours of work.

Another main aspect of our design is chip selection because multiple portions of the project require chips to perform an action. The two main blocks of the project that require some form of a chip are the Low-Noise-Amplifier and ADC. Some considerations for the chip selection in the LNA are signal-to-noise-ratio(SNR), if the device requires unipolar or bipolar biasing, system gain, size, and cost. The most important of the criteria is SNR and biasing because it plays a direct role into the functionality of the device. As for the ADC, some technology considerations are resolution, sampling rate, size, cost, and input. Zooming in further, our project requires an ADC with a differential input. So, another technology consideration comes into play because there are a few methods to establish a differential input. One of them being use of an instrumentation amplifier; the other being the use of a special transformer supplied by Texas Instruments. Both increase the overall complexity of the device, size, and cost but the aspect that separates the two is that a transformer operates poorly at high frequencies. Given that our project will operate in the MHz range this all but eliminates the possibility of a transformer.

For our wireless communication aspect, our team considered the options of Bluetooth or WiFi. We had distinct needs of message size, connectivity range, and latency. We had researched to find two modules for each type of wireless communication that we considered to fit our constraints. Our device currently only needs a range of 10 meters, which made the HC-05 Bluetooth module a good option compared to the ESP32 WiFi module which provided up to 100 meters of connectivity. The ESP32 allows much more of a range than we need, but could potentially provide room for growth. Our main concern for the modules is the baud rate and the latency. By creating test stands we would be able to observe the reaction of our interface connection with the modules to discover how the latency would be affected by the amount of data being sent.

For the signal processing, our team considered two primary options. We have to target a sample rate on the order of tens of megahertz, and although an FPGA board provides the most outright performance, the programming overhead and actual cost of implementing this option makes it non-scalable and unrealistic, rendering a microcontroller of some kind the only viable type of solution to choose from. A combination of raw clock speed and instruction efficiency have to be considered when looking for a suitable microcontroller, so Arduino-like boards were out of the question. The ESP32, an Arduino compatible board, was considered for the speed of development and neat packaging that includes the WiFi and data processing components in one place. While the raw clock speed is potentially double that of similarly priced options, the design of the peripherals crucial to our project were lacking in speed and features, and the instructions require significantly more CPU clock cycles for an equivalent program on many other platforms.

The STM32 is a widely-used ARM Cortex microcontroller with a diverse range of features, variation in products for different applications, and simply better performance in the areas that we seek. The clock speed on even the most affordable options eclipses AVR options by tenfold in most cases. The power consumption is also lower than an ESP32 and even the lower power Arduino boards. Instruction-set wise, there are a larger number of types on an ARM CPU than with an ATMega Arduino CPU, consuming just a single clock cycle to accomplish what would take between 2 and 4 cycles on an AVR-based chip. These things all mean that the ARM CPU can process and send data at a rate that meets our requirements for sample-rate, and by extension bandwidth and latency.

4.5. DESIGN ANALYSIS

So far, our project is finalizing the design for a few of the project stages. Starting off with the low-noise-amplifier(LNA) we were able to test a single rail IC chip called the MARS-6SM+ on a PCB board and test its characteristics. Our team was able to determine that with 2 of these ICs cascaded together and single 12 volt supply, we were able to get a gain of about 60dB. This prompted us to create circuit schematics and PCB layouts of our own using the MARS-6SM+ and a combination of passive components to establish a bandwidth of 100kHz-2.2GHz. The team is currently waiting for the boards to be soldered so test and evaluation can be conducted to determine if the design meets the requirements. The next stage we are in the midst of completing is the ADC. For our design, we are using a 12 bit ADC, ADC 12020. The chip has a differential input so we have implemented an instrumentation amplifier, and a network of passive components to establish a differential input and drive the ADC. Similar to the LNA, its circuit schematic and PCB layout have been completed and the team is waiting for the device to be soldered so test and evaluation can be conducted. It is worth noting that the devices make use of 50-Ohm SMA connectors. This will allow all the signals to propagate from one stage of the device to the next and tests and evaluations are conducted.

Development on a completely bare-bones platform to maximize speed has its tradeoffs, but the benefits factor in more heavily when the scope of the microcontroller's tasks are reliant on just a few simple instructions executed at the upper edge of available speed. The program so far uses several dummy values for debugging purposes, but the peripheral modules on the Cortex M4 that are in use are configured to run as fast as possible, at minimum 2.5 times the frequency of the ADC's highest possible sample rate. The largest task is to configure the inputs and data buffers to minimize latency, and efficiently format/package incoming data in a way that uses as little CPU overhead as possible. All of this must happen reliably, and be synchronized with only a small window for error. The next step will be to understand how to utilize the DMA module to increase program parallelism and further reduce processor cycles used.

Throughout the semester we have tested different wireless modules to learn their capabilities. To do this we created two test setups using a WiFI module and a Bluetooth module programmed with an Arduino. These test setups allowed us to learn the capabilities and restrictions of the wireless connections which led us to choose an ESP32 WiFI module for our device. We found that the module provided the least amount of latency, a further connection distance, and allowed us to transmit a large amount of data. We also tested both connected Bluetooth and WiFi to LabVIEW and different methods of presenting the data we wanted to the user. Today, our team has made a test stand with a ESP32 WiFi module and LabVIEW Interface that successfully communicates and presents data to the user. In the future, we want to work on this interface to improve the optimization of the LabVIEW program to faster process data and present it. We would also like an option in LabVIEW for the user to be able to record data to a text file. Currently, our interface has also been made for a developer which allowed us to test factors such as delays, errors, message size, etc., in the future we will need to redesign and decide which features the user will actually need available.

Works Cited

[1] Dewesoft. (2022, September 20). What is an ADC Converter? Dewesoft. https://dewesoft.com/blog/what-is-adc-converter

[2] T. Vu, D. Razansky, and J. Yao, "Listening to tissues with new light: recent technological advances in photoacoustic imaging," J. Opt., vol. 21, no. 11, p. 113001, 2019. [Online]. Available: https://iopscience.iop.org/article/10.1088/2040-8986/ab3b1a

[3] “Cortex M-4 Technical Reference Manual.” *Documentation – Arm Developer*, ARM Limited, 2010, developer.arm.com/documentation/ddi0439/b/CHDDIGAC