

IOWA STATE UNIVERSITY
COMPUTER AND ELECTRICAL ENGINEERING DEPARTMENT

CPRE EE 491 TEAM 25

**DISTRIBUTED SNIFFER NODES FOR BATTERYLESS SENSOR
NODES
DESIGN DOCUMENT**

December 7, 2023

Client/Advisor- Henry Duwe

Team/Software Lead- Thomas Gaul

Scribe/Software member- Ian Hollingworth

Software member- Spencer Sutton

Hardware Lead- Tori Kittleson

Hardware member- Matthew Crabb

sdmay24-25@iastate.edu

<https://sdmay24-25.sd.ece.iastate.edu>

Executive Summary

Problem Statement

We aim to create an open-source testbench for testing and benchmarking wireless batteryless sensor nodes that have been designed by our clients. This test bench needs to be made of modular units to allow test scalability. The units must interface with the wireless nodes to collect, record, and store information about their operation and performance.

Development Standards & Practices Used

C: Microcontroller code written in standard C with standard coding practices such as commenting and library use

Python: Host computer programming written in standard C with standard coding practices

UART: Standard UART serial communication between Sync Node and Host Computer allows seamless integration to any computer (RS-232)

SPI: Communication between the various portions of the BOB Node

KiCad: Standard open-source CAD software for designing PCBs

GitLab: Industry standard version control software

TIRTOS: Texas Instruments provided a Real-Time Operating System used to handle tasks and threading within code **Low Energy Bluetooth** Bluetooth 5.2 Low Energy (IEEE 802.15.1)

Hardware Design: Consistent with TI product line

Summary of Requirements

Modularity: Start with 9 nodes, ability to scale to 1000

Simulation: Sniffers must be able to provide faux sensor data to BOB Nodes for testing

Low Cost: Roughly \$20 per board

Electrical Isolation: Connecting Sniffer to BOB node causes no measurable change in BOB behavior

Open Source: Build to be well-documented, reproducible, and expandable for future groups to use and build upon

Applicable Courses from Iowa State University Curriculum

EE 201/230: Circuits I / II

EE 285: Intro to C

EE 330: Intro to VLSI

EE 465: Digital VLSI

EE 333: PCB design

EE 224/324: Signals and Systems I / II

EE 321: Communication Systems I

CPRE/EE 185: Intro to EE/CPRE

CPRE 281/288: Digital Logic & Embedded Systems

CPRE 381: Computer Structure and Assembly

CPRE 558: Real Time Systems

New Skills/Knowledge acquired that was not taught in courses

Technological Skills

- PCB design

- Soldering
- Wireless Transmission Protocols
- Python
- KiCad
- System-level Debug

Soft Skills

- Team Management
- Team Communication
- Client Negotiation
- Meeting management

Contents

1 Team	8
1.1 Team Members	8
1.2 Required Skill Sets for Your Project	8
1.3 Skill Sets covered by the Team	8
1.4 Project Management Style Adopted by the team	8
1.5 Initial Project Management Roles	9
2 Introduction	9
2.1 Problem Statement	9
2.2 Requirements & Constraints	9
2.2.1 Requirements	9
2.2.2 Constraints	10
2.3 Engineering Standards	11
2.4 Intended Users and Uses	11
3 Project Plan	12
3.1 TASK DECOMPOSITION	12
3.2 Project Management/Tracking Procedures	14
3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria	15
3.4 Project Timeline/Schedule	15
3.5 Risks and Risk Management/Mitigation	17
3.6 Personnel Effort Requirements	18
3.7 Other Resource Requirements	18
4 Design	19
4.1 Design Content	19
4.2 Design Complexity	20
4.3 Modern Engineering Tools	21
4.4 Design Context	22
4.5 Prior Work/Solutions	24
4.6 Design Decisions	25
4.7 Design 0 (Initial Design)	25
4.7.1 Design Visual and Description	25
4.7.2 Functionality	26
4.8 Design 1 (Design Iteration)	26

4.8.1	Design Visual and Description	26
4.8.2	Functionality	28
4.8.3	Design of Pinouts for the Stack	32
4.8.4	Breakout Board Design	35
4.8.5	MSP Simplified Design	35
4.9	Technology Considerations	42
4.9.1	Two band-pass off	42
4.9.2	Internet Enabled Sniffer Node	42
4.9.3	Two CC1352 Design	43
4.9.4	Both bands running on the CC1352 Simultaneously	43
4.9.5	Single band channel filtering	43
4.10	Design Analysis	43
5	Testing	44
5.1	Unit Testing	44
5.1.1	Software	44
5.1.2	Hardware	47
5.2	Interface Testing	53
5.2.1	Software	53
5.2.2	Hardware:	54
5.3	Integration Testing	55
5.3.1	Software	55
5.3.2	Hardware	55
5.3.3	Sub-System	56
5.4	System Testing	57
5.5	Regression Testing	57
5.6	Acceptance Testing	57
5.7	Results	58
6	Implementation	58
6.1	Hardware	59
6.2	Software	59
7	Professionalism	61
7.1	Areas of Responsibility	61
7.2	Project Specific Professional Responsibility Areas	62

- 7.3 Most Applicable Professional Responsibility Area 63
- 8 Closing Material 63**
 - 8.1 Discussion 63
 - 8.2 Conclusion 63
- 9 Appendices 66**
 - 9.1 Breakout Board Design Files 66
 - 9.2 Simplified MSP-430 Board 72
 - 9.3 Team Contract 80

1 TEAM

1.1 Team Members

Matthew Crabb, Thomas Gaul, Ian Hollingworth, Tori Kittleson, Spencer Sutton

1.2 Required Skill Sets for Your Project

Embedded Systems

PCB Design

Electrical Systems

Radio Communication Protocol

Soldering

Debugging

1.3 Skill Sets covered by the Team

Embedded Systems- All members

PCB Design- Matt, Tori, Ian, Thomas

Electrical Systems- Thomas, Matt, Tori

Soldering- All members

1.4 Project Management Style Adopted by the team

We have committed to an agile management style with goals for each week which will be assigned to each team member. Daily scrum meetings will be in the form of consistently

messaging and two to three meetings a week. We will use GitLab to for goals and milestones.

1.5 Initial Project Management Roles

Thomas- Team lead, Technical Software

Matt- Technical hardware

Tori- Hardware lead, Technical hardware

Ian- Scribe, Technical software

Spencer- Technical software

2 INTRODUCTION

2.1 Problem Statement

We aim to create an open-source testbench for testing and benchmarking wireless batteryless sensor nodes that have been designed by our clients. This testbench needs to be made of modular units to allow test scalability. The units must interface with the wireless nodes to collect, record, and store information about their operation and performance.

2.2 Requirements & Constraints

2.2.1 Requirements

- Provide scalability in design in Sniffer code and communication and hardware for BOB and Sniffer (1 Sniffer per BOB). Goal: 9 nodes, 1000 nodes design analysis.
- Mechanical durability of BOB and Sniffer interface and stand-alone unit for testing use.
- Implement a system that is easy to move for testing.

- Set pin requirements that map to each other.
- Implement hardware for wireless mass reprogramming of BOB.
- Provide thorough documentation on the project.
- Include make file for all project documents.
- MSP430 should have a power switch for the CC1352 so that it is not always on. Need to prototype this and ensure it meets the electrical characteristics/requirements.
- Do not have a modification of BOB code for Sniffer implementation.
- Sniffers communicate with each other to pass log data from BOBs to the Host and compare data from BOB.
- Have a Host system that organizes and stores logs from Sniffers.
- Sniffers record both wireless data and wired. Wired can optionally be disconnected.

2.2.2 Constraints

- Use CC1352 microcontrollers in our Sniffer design
- Our Sniffer network must record all transmission from BOBs at a maximum rate of 1 8-byte message every 5ms.
- Sniffers only cause trivial effects on BOB's performance (Sniffers are electrically isolated from each other and have minimal effect on BOB up time).
- Have 9 Sniffer/BOB pairs fully constructed by the end of the year.
- Central Sniffer Node that cannot lose power and collects data from other Sniffer nodes. The central Sniffer Node can be (statically) selected from any Sniffer Node.
- Sniffer to Sniffer communication does not interfere with BOB communication.
- 20 Dollars or less per Sniffer node.
- Wireless Sniffer Nodes must have the ability to operate continuously for a week.

2.3 Engineering Standards

Software

- UART communication protocol: Standard way for microcontrollers to communicate with PC's, will be used to connect our Sink node to our Host PC. RS232
- Radio communication standards such as , Bluetooth 5.2 Low Energy (IEEE 802.15.1), RS-232, Proprietary 2.4 GHz
- We need these communication standards to communicate between multiple sniffer nodes sending data back to the Host

Hardware

- Standard pin header for connecting between boards
- Revision numbers
- Version control for PCB design
- Standard circuit schematic design
- Effective organization of component library

2.4 Intended Users and Uses

Our clients are Professor Duwe and his graduate assistants, with secondary users being other researchers who might want to use our testing infrastructure for their own work.

Our clients will be using this testbench to test their sensor nodes and write research papers about them, so they need it to be as versatile as possible.

Our design and software will all be open-source for other researchers and designers. They may build upon and expand our designs, so it will all need to be as modular and well-documented as possible.

3 PROJECT PLAN

3.1 TASK DECOMPOSITION

BOB Development

1. Determine specifications with the client.
2. Plan header configurations and rough board sizing.
3. Develop a BOB hardware design.
4. Develop BOB schematic.
5. Design Review with the client.
6. Develop BOB layout.
7. DRC checks.
8. Peer review for errors.
9. Generate design gerber files.
10. Attain quotes and order boards.
11. Generate BOM and order parts.
12. Document design and function of BOB hardware.
13. Generate schematic and board layout documents.
14. Add documents to the design doc.
15. Populate Boards.
16. Hand solder.
17. Clean boards.
18. Visual inspection.
19. Testing functionality of BOB hardware.
20. Power supply test – look for short circuits.
21. Plan functional tests.
22. Perform functional tests.
23. Integrate into the current system.
24. Deliver to the client.

Sniffer Hardware Development

1. Determine specifications with the client.
2. Determine system needs and specifications.
3. Wireless needs.
4. Headers and pinouts.
5. Power system.
6. Processor and processor interface.
7. Programming interface.
8. Board size.
9. Develop Sniffer schematic.
10. Plan header configurations and rough board sizing.
11. Design subcircuits (novel and from reference) - find parts.
12. Outline schematics in KiCad and format.
13. Design Review with the client.
14. Develop Sniffer layout.
15. Component placement and routing.
16. DRC checks.
17. Peer review for errors.
18. Generate design files.
19. Attain quotes and order boards.
20. Generate BOM and order parts.
21. Order stencils.
22. Document design and function of Sniffer hardware.
23. Generate schematic and board layout documents.
24. Add documents to the design doc.
25. Test functionality of Sniffer Hardware.
26. Power supply test – look for short circuits.
27. Plan functional tests.
28. Perform functional tests.
29. Deliver to the software team for testing.
30. Program boards.
31. Perform testing and development tasks.
32. Deliver boards to the client.

Physical Interfacing

1. Physical interfacing between BOB, Sniffer, and testing environment.
2. Design the custom MSP430 board that physically connects to both the CC1352 and the power board.

Sniffer Firmware Development

1. Develop Sniffer firmware.
2. Develop a communication system between Sniffers.
3. Research and decide on a communication protocol.
4. Establish a connection between Sniffers.
5. Successfully sniff with our own firmware.
6. Sniffers accurately monitor BOBs performance.

3.2 Project Management/Tracking Procedures

Waterfall and Agile Approach:

We are looking at semester scheduling in terms of the waterfall approach. These will be our rough deadlines for the semester. For example, what this would look like for hardware is: design planning, schematic, layout, physical hardware testing, and system integration. This will follow a linear approach for each subteam: electrical and hardware.

Similarly, the team is going to be working simultaneously on various aspects of the project as the semester goes on. This is to make use of all our resources throughout the duration of the semester. Deadlines will also be more agile and adjusted as the client's needs change. For example, both electrical and software project planning and work will occur in parallel.

Our group is using GitLab issues for weekly progress and keeping version control with our project files throughout the semester. Teams will be used for group communications and clarifications. For semester planning, we will use the schedule found below.

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

The following are the key milestones in our project:

- Simplified BOB schematic design review
- Simplified BOB PCB layout design review
- Simplified BOB PCB ordered
- Sniffer schematic design review
- Sniffer PCB layout design review
- Sniffer PCB ordered
- Testing of all BOB hardware features completed
- Testing of all Sniffer hardware features completed
- Establish successful communication between two CC1352 development boards
- Successfully listen (sniff) a test actively with a CC1352 development board
- Successfully pass Sniffer data to the Sink Node
- Successfully record data from the Sink Node to the Host Node
- Establish communication between a network of Sniffer Nodes
- Scale the design to achieve 10 Sniffer Nodes accurately communicating with each other

3.4 Project Timeline/Schedule

Our team early on planned out the project early to allow for the completion of the goals.

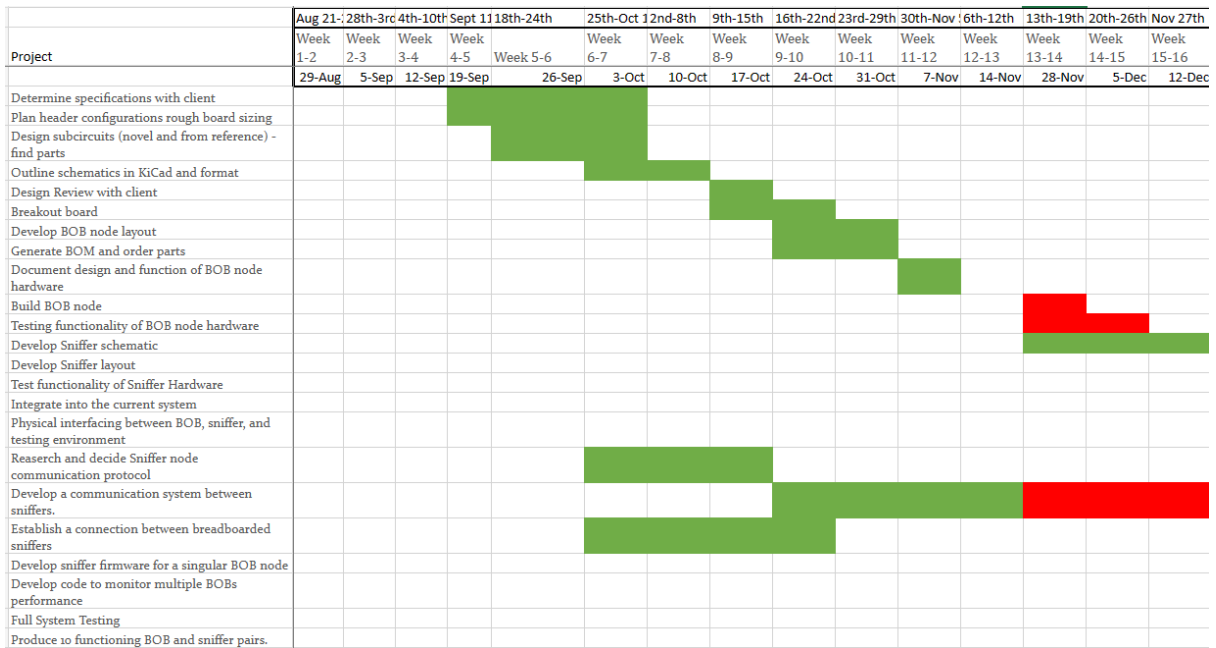


Figure 1: The schedule we followed during CPRE/EE 491

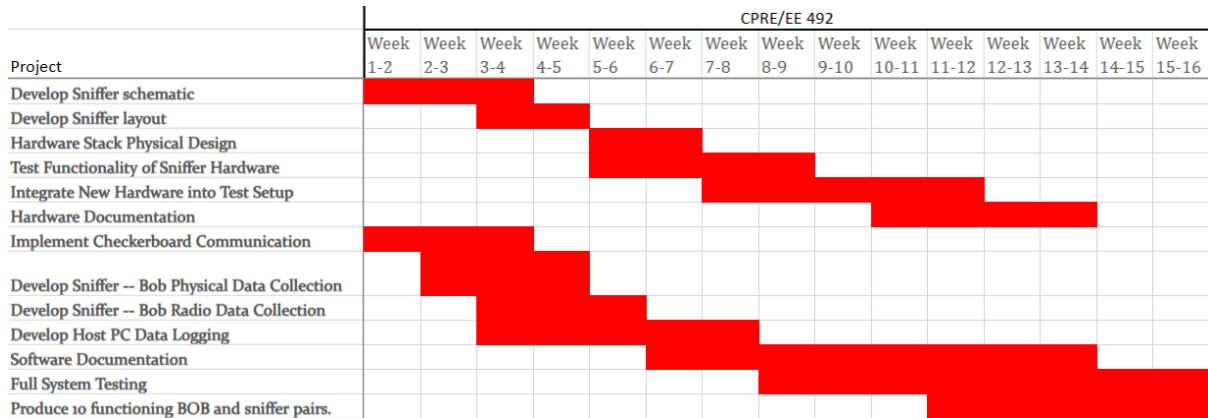


Figure 2: Intended Schedule for CPRE/EE 492

3.5 Risks and Risk Management/Mitigation

Risk Mitigation	Risk Factor	Mitigation Strategy
Hardware order delayed	0.2	Leave time in schedule
Component order delayed	0.1	Order components closely after PCBs are ordered
BOB PCB testing fails	0.2	Leave time for a second PCB revision
Eval board destruction	0.2	Proceed with caution with the eval boards
Incorrect layout delays software programming	0.1	Leave time for a second PCB revision
Damage to physical PCB	0.3	Order extra PCBs to use as backups
Poor BOB soldering	0.3	Have experienced members soldering and teaching solder basics
Unable to establish communication while also listening to BOB communication	0.6	Start early to ensure enough time to develop alternate. Fallback plan is to have the different Sniffers cover for each other complicating the firmware. Alternatively, we can add more microcontrollers

Table 1: Risk and Risk Mitigation

3.6 Personnel Effort Requirements

Tasks	Hours
Develop a BOB hardware design	10
Develop BOB layout	8
Document design and function of BOB hardware	10
Populate BOB boards	8
Testing functionality of BOB hardware	8
Integrate into the current system	8
Develop Sniffer hardware design	25
Develop Sniffer schematic	6
Develop Sniffer layout	15
Document design and function of Sniffer hardware	10
Populate Sniffer Hardware	20
Test functionality of Sniffer Hardware	8
Deliver to software team for testing	1
Deliver boards to the client	1
Physical interfacing between BOB, Sniffer, and testing environment	6
Have a CC1352 listen (sniff) an actively running test	4
Get two CC1352s to communicate	5
Develop Sniffer firmware	35
Test Sniffer firmware	10
Develop a communication system between Sniffers	10

Table 2: Personnel-Effort Requirements

3.7 Other Resource Requirements

- Time with Mentor and Clients
- Soldering Equipment Access
- Code Composer Studio
- KiCAD

- Coffee

4 DESIGN

4.1 Design Content

The design content can be split into four main components: Sniffer design, Host design, BOB updates, and system integration.

The main task of this project is to design a set of Sniffer Nodes, which will provide a testbed for the BOB Nodes by accurately recording data on BOB operation and by emulating an environment in which the sensor resides. Both hardware and software systems must be designed and integrated for the Sniffer Nodes.

The Sniffer hardware tasks consist of designing a PCB that contains an MCU, a wireless communication interface and an antenna, headers to connect to other boards, and a power supply system. The wireless communication system will be designed to pick up BOB communications. The headers must be carefully designed to provide all necessary monitoring without interfering with the BOB operation. The power system contains design decisions on how to power the Sniffer with a battery as well as how the power supply can be switched to a continuous power supply.

The Sniffer software will contain design decisions mainly regarding communication protocols and data aggregation and sorting. The Sniffer software will be configured to work on the MCU. The Sniffer will have to be designed to communicate with 3+ MCUs from other boards simultaneously which will require intensive software design. The Sniffers will also have to move data to the Host Node in some fashion. This will require large amounts of data to be sorted and transferred in an efficient manner. This will also present difficult design challenges.

The testbed also requires a Host Node. The Host Node will have a hardware system that it operates on and software that collects and stores data in a form that is easy for researchers to access and manipulate. Design decisions must be made on what hardware system to use and how it will communicate with and receive data from the Sniffers. A software system will also

need to be designed which provides accurate, easy to access data.

In addition to designing the testbed, Dr. Duwe's research group has asked us to simplify their current BOB design and add some new functionality to it. This task includes making design decisions on how best to simplify the board as well as designing new circuitry.

All systems mentioned above will need to be designed in such a way that they work together functionally as a whole. The hardware requires designs that can interface with each other and easily be connected. The software will require communication protocols which work with one another. The Sniffers and the BOB nodes will need to communicate but also maintain electrical isolation from each other. This leads to a situation where every design decision for each part also affects the whole and must be carefully considered.

Lastly, the systems must also mechanically interface with each other and be able to be easily set up for tests. Dr. Duwe's group is hoping to hang each BOB Node and Sniffer pair from the ceiling together. This will require design on the hardware side as well as the design of mechanical hangers.

Overall, this project contains a vast and varied amount of design content.

4.2 Design Complexity

Our project has a variety of aspects contributing to making it a complicated and challenging project for the capstone course.

To meet the requirements for Senior Design, our project must be complex. There are two main criteria which determine complexity. The first criterion asks if the design has multiple elements that each require different scientific, mathematical, and engineering principles. The second criterion asks if the design task requires solving challenging problems which have not been solved, or have not had industry standard solutions adopted yet.

Our project satisfies the first criterion but does not address problems modern enough to meet the second criterion. Our project requires the design of multiple subsystems which must work together. The main subsystems we will design are:

1. A Sniffer Node PCB

2. A simplified BOB Node PCB
3. A common header pinout which allows all PCBs (4 total) to be stacked and to communicate without interfering with each others operation
4. Software for the Host Node to recieve and organize data
5. A mechanical mounting system to hold all 4 boards together and hang them from the ceiling
6. Software for the Sniffer Nodes to listen and send communications
7. A communication scheme to route all packets from multiple Sniffer Nodes to the Host Node

Designing these subsystems will require the use of a number of mathematical and engineering principles. The hardware design for the Sniffer Node and simplified BOB node require schematic design and capture, calculations to determine system needs accurately, soldering, RF PCB layout, reverse engineering of provided reference designs, usage of test equipment, and hardware debugging. Designing the software for the Sniffer Nodes requires C programming, Python scripting, usage of multiple different communication protocols, coding using operating systems, writing software tests, and debugging software. Designing the communication scheme requires using math to plan out the order of events and usage of communication protocols. Designing the mechanical mounting system will require 3D modeling, usage of workshop tools for fabrication, and potentially 3D printing.

The project contains much complexity in the number of subsystems to design and the various engineering skills required to design each subsystem. Not directly relating to the criteria, there will also be a vast amount of complexity in implementing ten Sniffer Nodes and a Host Node which all seamlessly work together. This will require careful planning, time management, and debugging skills.

4.3 Modern Engineering Tools

We will require a handful of engineering tools to accomplish our goals and project

Code Composer Studio (CCS)- We use CCS to code and program in C for all the micro-controllers within our project.

Visual Studio Code (VS Code)- We use VS Code Python to write and run the program for collecting and compiling the data from tests.

KiCad- KiCad is used for all aspects of our PCB design. It is used for creating schematics and footprints in our parts library and the design schematic and layout for our PCBs (Printed Circuit Board)

AutoDesk Inventor- We will use Inventor (or similar CAD software) to model any of the physical components required by our project.

GitLab- We use GitLab for doing version control and project management tools

4.4 Design Context

Our project is used in the context of research by our clients, who are part of a research team led by Professor Duwe. One of our constraints is to make all aspects of our design available as open source. With open source, other batteryless sensor enthusiasts may choose to use or update our testbed design to fit with their own project needs. This research project is designed to be able to develop systems that could be applied in a wide variety of fields such as manufacturing, farming, infrastructure, and more.

Given the broad scope of this project, it creates a lot of unique constraints and requirements. We must make our design flexible to fit a wide variety of testing environments, particularly with system communication. Along with this comes documentation, and we would like to document our work clear and concisely so that someone with limited experience in this field can quickly comprehend and implement the work themselves.

Area	Description	Example
Public Health, Safety, and Welfare	During the testing process, radio signals will be sent between various MCUs on both BOB and Sniffer nodes.	We will work to avoid interference with other people's devices during the testing process.
Global Cultural and Social	A BOB network can be used in professional and workplace settings to increase overall work efficiency.	In workplaces where collecting sensor data is important, a BOB network will automate the process, saving time for people in the workplace.
Environmental	The goal beyond the senior design project is to create a functional general-purpose network of tens to thousands of sensor nodes that can be used in a variety of situations. These situations could include cases where BOB nodes are distributed throughout a natural environment.	It will be important to dispose of BOB nodes in an environmentally friendly way after they are no longer of use, especially in the case where hundreds to thousands of nodes may be scattered in varying environments.
Economic	After the testing is complete and BOB nodes are distributed on a large scale in a practical setting, the BOB nodes can be used to reduce the cost of monitoring processes for the user. For example, a network of BOB nodes could be distributed around a factory to monitor temperature, pressure, or other relevant information to identify if everything is running accordingly.	The data from the BOB network could help improve factory cost efficiency or could be used as a safety mechanism to turn off certain processes if things go wrong. In any case, increased monitoring capabilities in many cases will reduce the burden of cost on the BOB network user.

Table 3: Project Context

4.5 Prior Work/Solutions

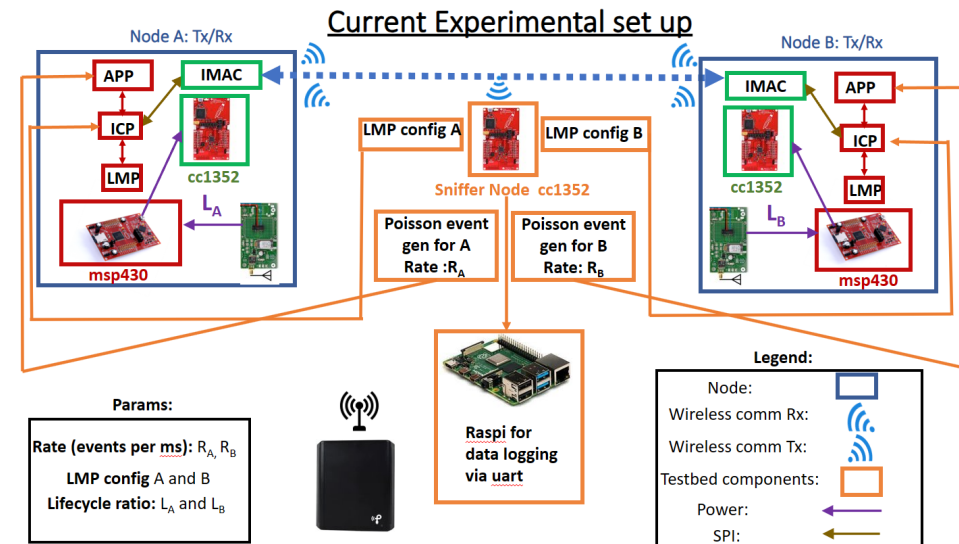


Figure 3: The current setup design the Research Assistants use

Figure 3 shows a testing setup used as a prior solution by the research team to run tests on two BOB nodes with a single Sniffer. This test works and is currently being used but we have been tasked with expanding it. Our goal is to create a setup that has each Sniffer paired with a single BOB as opposed to the current system. Additionally, we want all the communication back to the Sink node to be conducted wirelessly. The other goal is to make the wiring cleaner and completely remove the need for jumper wires. Additionally, we will be making the stack of boards more robust mechanically, resolving that challenging aspect of debugging.

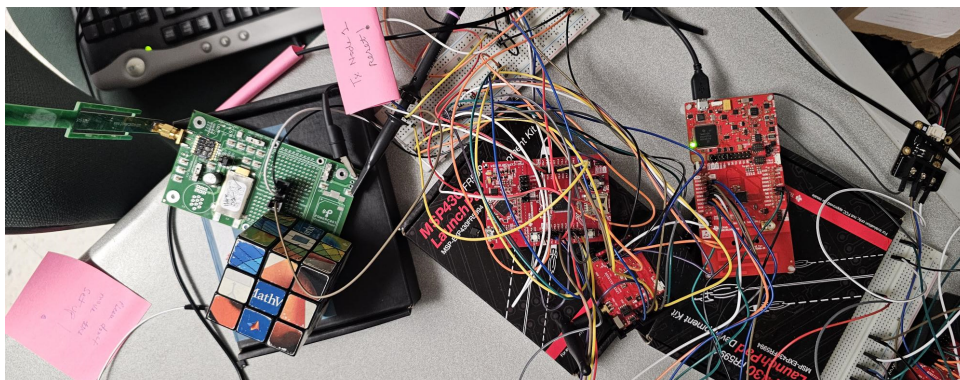


Figure 4: The current setup the Research Assistants use

4.6 Design Decisions

We decided to do communication between the Sniffer nodes with the CC1352 radio and not with an ESP8266 because of power constraints.

BOB communication will take place on a sub-1Ghz band with a custom communication protocol developed by the research team. The Sniffer communication will use 2.4Ghz Bluetooth communication.

We decided to use a load switch to control power between the MSP430 and the CC1352 BOB.

We decided that the boards would be stacked from bottom to top as Powerboard, MSP-430(simplified), CC1352(radio board), and Sniffer.

4.7 Design 0 (Initial Design)

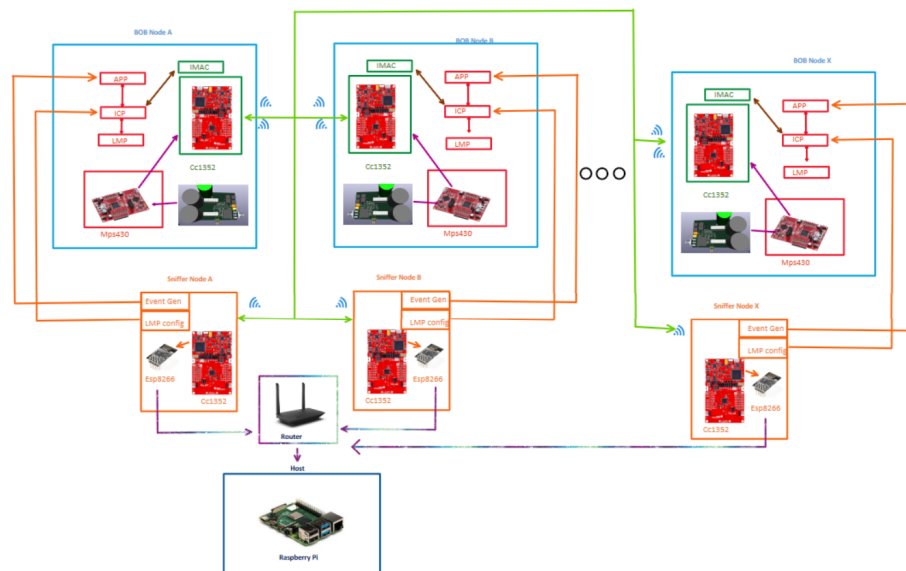


Figure 5: The initial idea we had with ESPs

4.7.1 Design Visual and Description

Our initial plan was to include an ESP8266 on the Sniffer Nodes that would connect to the ISU Wi-Fi network or a router. It would be used to communicate all test logs back to the Host

and would allow us to not worry about range too much and always be able to have Sniffer nodes listening and logging data. This design was intended to be very modular, meaning that more nodes should easily be added to the system. Adding more nodes would be very straightforward; all aspects would be wireless, maintaining a clean setup. The data logging would be on a different band than the BOBs, so no interference would be created.

4.7.2 Functionality

This initial and since scrapped design, would function much like the research team's current design. The Sniffers would listen to wired and wireless communication from the BOB simultaneously, as well as have the functionality to create tests on the BOB boards using faux sensor data. The main difference is that instead of communicating to and from the Host wired via UART, the Sniffers would communicate using an ESP8266 to relay data back to the Host using the internet. This design would likely fill all requirements but harbored a concern of requiring too much power to reasonably be powered off a battery for a week or more. Additionally, our advisor thought adding another microcontroller and the communication work required would add too much complexity for this stage of the project.

4.8 Design 1 (Design Iteration)

4.8.1 Design Visual and Description

Our current iteration involves having the Sniffer nodes harvest all their data independently, then passing them through a predetermined route of Sniffers until the data ultimately reaches a Sink node. This Sink node would be the "test Host" of all Sniffers. We are unsure if we can use two communication bands simultaneously on the CC1352, or if we will have to create a strategic switching protocol to allow the board to both listen and communicate effectively. However, if we achieve it, the data will reach the Sink node and be communicated to the Host via UART for data logging and processing. Conversely, to set up tests, the Host communicates the setup to the Sink Sniffer via UART, and the Sink node will create a ripple distribution throughout the network until every board has received the setup and implemented it.

We took this design as it was suggested by our advisor, and it will require less power than design 0.

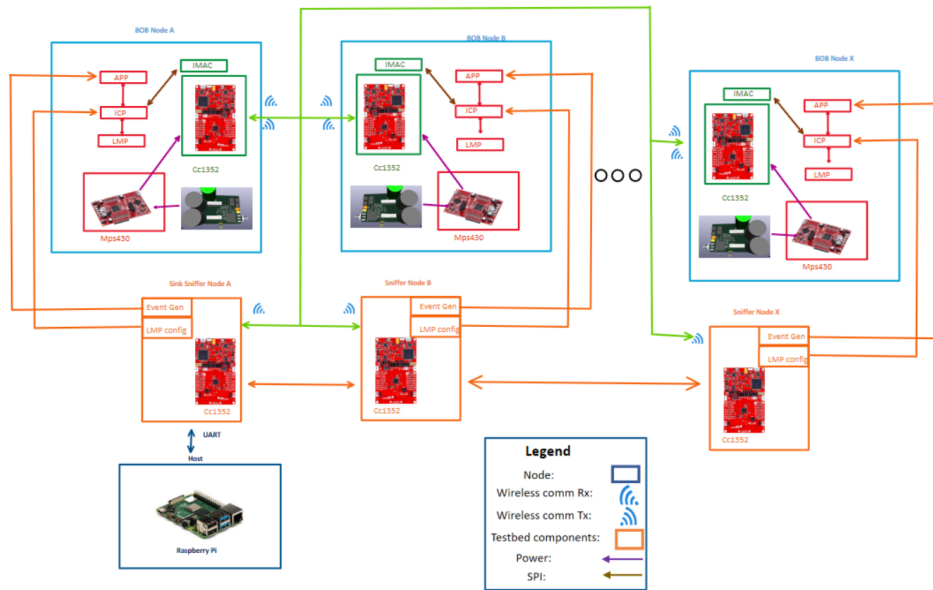


Figure 6: Our current design plan

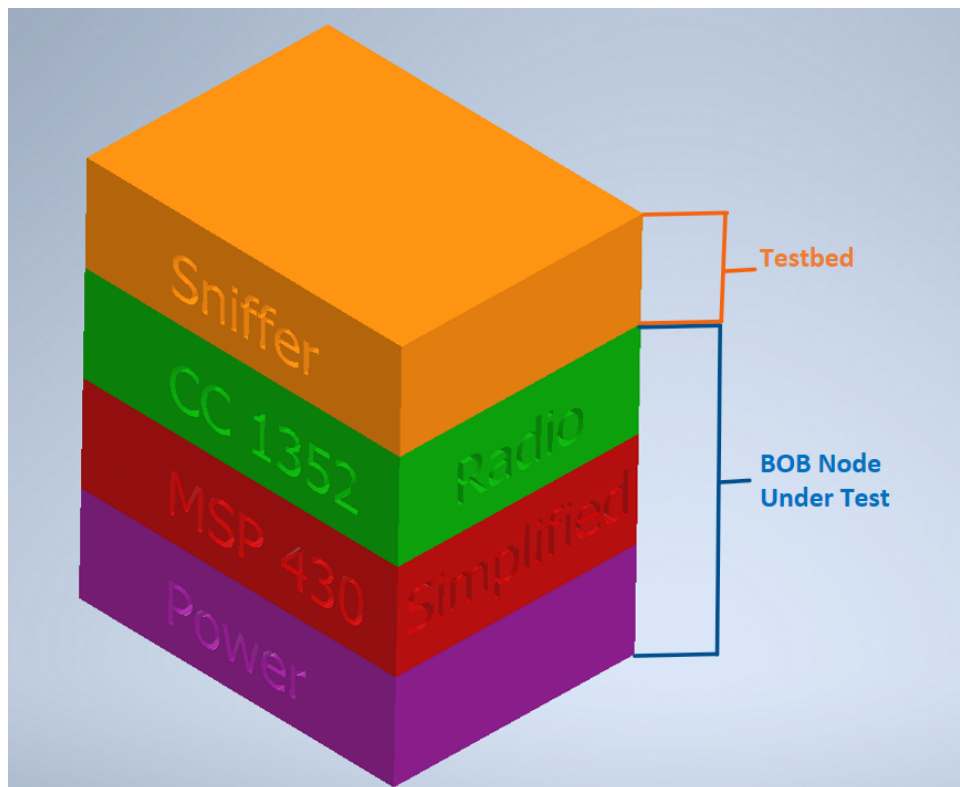


Figure 7: Our current hardware stackup plan

The above diagram gives a model for what our PCB stack up will look like.

The current design involves a 4-board stack. The bottom most board is the power harvester board. This will harvest the RF energy and supply power to the entirety of the stack. The current power harvester board is an off the shelf evaluation board. One of Dr. Duwe's graduate students, Vishak, will be designing an in-house board to replace the current revision. This will allow for more flexibility with the connector stack-up, as far as modification of connector pinout goes.

The next board in the stack is the MSP430. This board is currently an evaluation board and is being simplified down by our senior design team to condense the design and keep only the used functionality of the board. This is included as part of the BOB Node. This board is placed between the MSP430 and the CC1352 as both boards will contain radios and there are concerns of interference between the two. The two radios that connect via SMA connector will be placed on opposite sides of the board stack to account for this.

The CC1352 or "radio board" is used for communications both between each node and to the Host, through the Sniffer testing board.

The topmost board is the Sniffer board which is the primary focus of our senior design project. The Sniffer will be probing into the main connector of the node and evaluating the performance of the BOB Node. The Sniffer will only be used for initial prototyping and implementation. When the product is distributed and functioning beyond these steps, the Sniffer board will not be included in the unit. This is why the board is placed at the top of the stack as it can easily be removed once the unit is ready to be used.

4.8.2 Functionality

One of our main considerations for our current proposed solution is deciding how to allow each of the nodes to switch between the BOB and Sniffer communication bands without losing any information. The idea we landed on was having everything time synced relative to the Sink node, then having half of the nodes on each band at a time swapping off at regular intervals. We will have to have overlap where everything is on the BOB band for a short time so as to not miss any packets. A rough timing diagram of this system is shown in Figure 8.

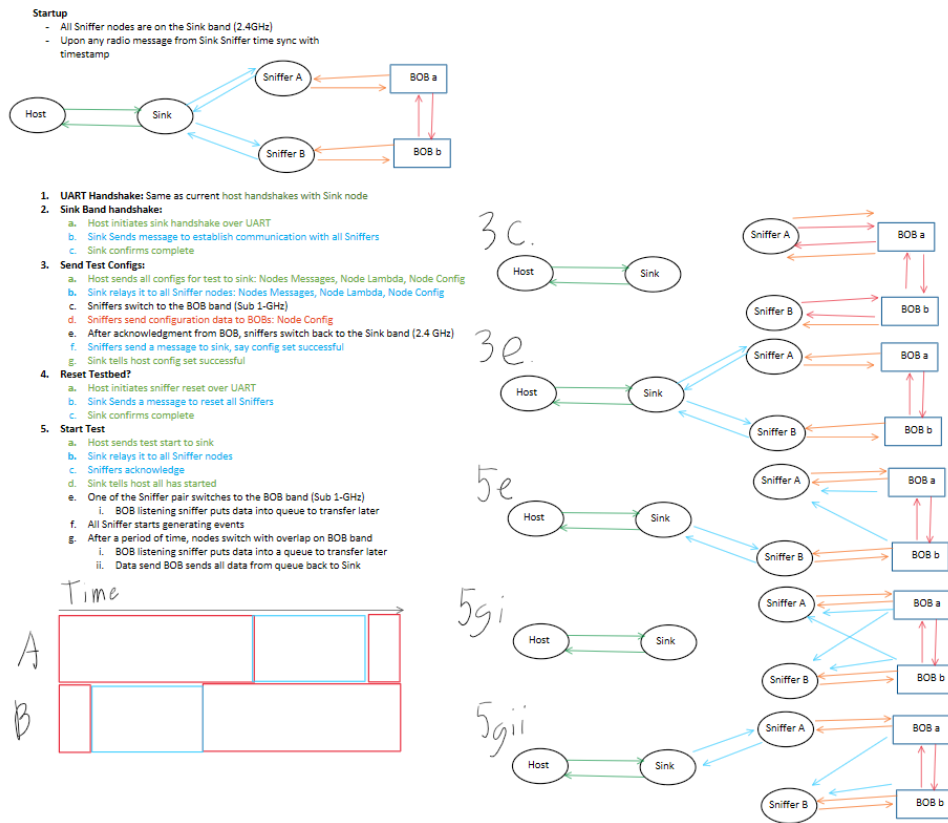


Figure 8: Our two-band pass-off

It can be seen that half of the Sniffers are listening back to the BOB nodes, while the other half are talking and listening to the other Sniffer nodes. This would happen according to the schedule shown at the bottom the image. The diagrams on the right side of the image are an example of how dataflow would work between the nodes over time.

Another important aspect of this band pass-off we decided upon is that in order to make it work without losing packets from the BOBs during pass-off time, we need to make sure that adjacent nodes don't end up on the Sniffer band at the same time. To this end, we decided upon a checkerboard pattern of band swapping, where each Sniffer has a specific data-path back to the Sink node. This simplifies the problem of switching bands, as we will always know which node is on what band based on the checkerboard we design, as everything will be reasonably synced to the Host node's time, as discussed. A diagram of this checkerboard pattern can be seen in Figure 9.

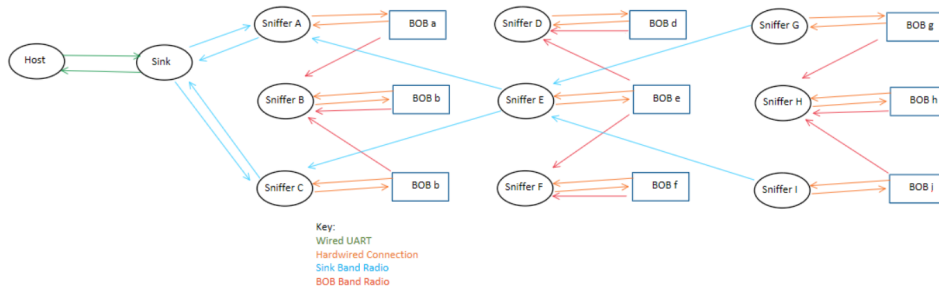


Figure 9: Checkerboard Band Swap

To ensure that we can maintain the band switching algorithm with the rates of transmission, band switching, saving, and loading to the queue under worse case load of one 8-byte transmission every 5ms, we ran some timing tests.

With some prototype testing, we found that we can send a 128-byte (maximum size) packet in 12ms, we can switch between bands (from BOB to Sink, and from Sink to BOB) in about 30 ms, and finally, can create an array-based queue of 2500 10 byte packets. In this queue, we can read or write the entire or it in under 15 ms.

With this data, we can generate a simple table (table 10) to show that we can maintain a band-switching procedure that can save all data while listening to the BOBs in the BOB band and transmit all the data back to the Sink while in the Sink band while being able to maintain the overhead of the band switching time while with the constraint of queue size. In this diagram, we are taking delays incurred by the queue as negligible and giving a 5 ms buffer to handle variation in Band Switching time. In actual implementation, we will have it run for longer on each band to make further use of our array of 2500 data packets and if we can make use of a maximum packet size we can transfer 128 bytes in on-go at the fastest rate.

Time(ms)	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
Odd	[Red bar]																			
Odd(Bytes)	8	16	24	32	40	48	56	64	64	64	64	64	64	64	0	0	0	0	0	0
Even	[Red bar]																			
Even(Bytes)	0	0	0	0	0	0	0	8	16	24	32	40	48	56	64	72	80	88	96	104
Time(ms)	105	110	115	120	125	130	135	140	145	150	155	160	165	170	175	180	185	190	195	200
Odd	[Red bar]																			
Odd(Bytes)	0	[Red bar]																		
Even	[Red bar]																			
Even(Bytes)	112	120	128	136	144	152	160	168	168	168	168	168	168	168	168	40				

Figure 10: Example Timing Schedule

Another main consideration of the Sniffer Node design is how it will be powered. The requirements for the power system is that it is easy to use, can power the Sniffer for up to 7

days continuously, and that the Sniffer node can optionally be powered by a wire to ensure that it does not fail. To meet these requirements, our design is to use a battery system to power the Sniffer and to also provide a wired power connection. When using the wired power connection, the system will either automatically switch over to the wired power option, or a switch will be available for the user to toggle between power modes.

There are a number of considerations to take into account for the design of the battery system. The first is the power, voltage, and current needs of the Sniffer board. The main element on the niffer board is the CC1352R processor. According to the datasheet, the CC1352R has its own internal regulators. This negates the need for a regulator on the chip. The CC1352R requires a supply voltage between 1.8 – 3.8V. According to measurements made in the datasheet, the maximum operating current of the chip is about 25mA. Our requirement will be that the battery system can supply 35mA to give plenty of margin. The battery system must meet all CC1352R power needs and have the capacity necessary to run the Sniffer for 7 days continuously (this requirement is estimated below).

The second consideration is cost of the system. One of the Sniffer Node requirements is to have a low cost. The client desires that the cost is low to enable more nodes to be tested on limited research funds and so that other groups can cost-effectively begin using our test system. The cost of the system includes batteries, ICs required to run them, mounting solutions, and potentially recharging systems. The goal is to reduce the cost per Sniffer. The cost also includes the cost of future use (for example if batteries must be replaced).

The third consideration is ease of use and practicality. The system should be easy for the graduate students to set up and use. For example, the system would not be easy to use if the batteries were difficult to put in and take out or if the batteries took an inordinate amount of time to charge.

The fourth consideration is sustainability. Disposal of batteries can have negative environmental effects. If possible, the environmental impact of our design should be minimized.

The next sections cover an estimation of the worst-case battery capacity requirement and several of the design ideas we generated. These will be reviewed with the client to determine the most optimal design.

4.8.3 Design of Pinouts for the Stack

The PCBs must connect in a simple, mechanically sound way. Mechanically sound means that they all stay together and that the connections between them are secure. The solution chosen to complete this task is to put all the PCBs together in one stack connected by identical headers. All TI development boards use an identically spaced header with a standardized pinout. This is shown for the CC135R development board below.

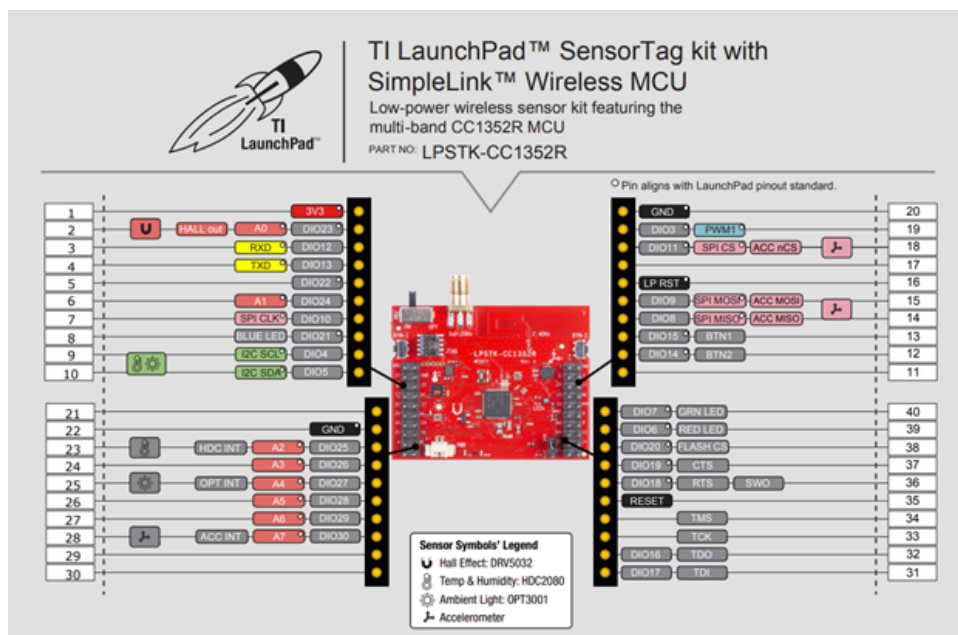


Figure 11: TI CC1352R Development Board Quick Start Guide [1]

The plan is to use the friction of the header connections to hold the stack together. However, this will likely be supplemented by additional mechanical support such as non-conducting bolts through mounting holes in the boards or a 3d printed frame.

To enable this solution, the pinout must be carefully designed to ensure the pins of all four PCBs are properly connected. The middle boards must also have a small footprint due to the antennas and capacitors sticking up or down from the radio board, Sniffer board, and harvester board. This is one of the main motivations for designing a new MSP430 board. Additionally, the Sniffer pinout and Harvester pinout needed to be made compatible with the stack. The research team gave us the details of all connections needed and we designed the pinouts for all boards.

The pinout design centered around the existing pinout of the CC1352R development board because it cannot be altered. There are a number of connections that must be made between only the MSP430 board, Harvester board, and Sniffer board and not the CC1352R board. To make these connections, the no connect pins on the CC1352R were used. However, there were not enough to meet the requirements. To address this problem, fourteen pin headers will be used as offsets to the CC1352R development board. This creates more no connects which allows the pinouts to meet the requirements as shown below.

Current test bed (Updated Pin details for BOB node SD)

Table 1				Table 2					
GPIO	MSP430	CC1352 radio	I/O (as seen from the node)	GPIO	MSP430	CC1352 radio	CC1352 sniffer	Harvester	I/O
Data Received	P5.0	DIO22	I	Powered ON	P7.7		DIO25	DIO28	O
Transmit Request	P5.1	DIO3	O	Event Gen	P7.4		DIO26	DIO29	I
Transmit Done	P5.2	DIO24	I	Testbed Reset	P7.5		DIO27	DIO30	I
SPI Master Ready	P5.3	DIO19	O	EasyLink Tx		DIO25	DIO24	DIO21	
SPI Slave Ready	P5.4	DIO7	I	Event drop	P7.6		DIO9	DIO8	O
FRAM Written	P5.5	DIO11	O	Reset	P7.3			Reset	I
Power radio	P1.4								
SPI MOSI	P6.4	DIO9							
SPI MISO	P6.5	DIO8							
SPI CLK	P6.6	DIO10							
SPI SS	P6.7	DIO20	O						

Note currently in our setup we have only one sniffer for two msp430 nodes. I/O are defined with respect to msp430 node
Code needs update

Figure 12: Plan to Create Extra NC Pins on the CC1352R Development Board

The pin connection requirements provided by the research team are shown in the following figure.

Current test bed (Updated Pin details for BOB node SD)

Table 1				Table 2					
GPIO	MSP430	CC1352 radio	I/O (as seen from the node)	GPIO	MSP430	CC1352 radio	CC1352 sniffer	Harvester	I/O
Data Received	P5.0	DIO22	I	Powered ON	P7.7		DIO25	DIO28	O
Transmit Request	P5.1	DIO3	O	Event Gen	P7.4		DIO26	DIO29	I
Transmit Done	P5.2	DIO24	I	Testbed Reset	P7.5		DIO27	DIO30	I
SPI Master Ready	P5.3	DIO19	O	Easylink Tx		DIO25	DIO24	DIO21	
SPI Slave Ready	P5.4	DIO7	I	Event drop	P7.6		DIO9	DIO8	O
FRAM Written	P5.5	DIO11	O	Reset	P7.3			Reset	I
Power radio	P7.4								
SPI MOSI	P6.4	DIO9							
SPI MISO	P6.5	DIO8							
SPI CLK	P6.6	DIO10							
SPI SS	P6.7	DIO20	O						

Note currently in our setup we have only one sniffer for two msp430 nodes. I/O are defined with respect to msp430 node
Code needs update

Figure 13: Pinout Requirements Provided by Research Team

The final pinouts settled on for the harvester board and the MSP430 Simplified board are shown below. The extra pins that did not have specific functionality are connected to GPIO pins on the MSP430, Simplified with jumper resistors in between. This will allow the research team to use these pins in the future for any functionality that may be desired. This effectively makes the whole stack much more configurable, enabling easier development.

The Sniffer pinout has not been fully completed yet, as the Host to Sink communication is still being determined.

MSP Board Pinout							
Pin #	Func	Pin #	Func	Pin #	Func	Pin #	Func
1	3V3 to CC	21	3V3	40	P5.4	20	GND
2	GPIO	22	GND	39	GPIO	19	P5.1
3	GPIO	23	NC	38	P6.7	18	P5.5
4	GPIO	24	GPIO	37	P3.5	17	GPIO/EN
5	P5.0	25	GPIO	36	GPIO	16	NC
6	P5.2	26	GPIO	35	GPIO	15	P6.4
7	P6.6 (SPI)	27	GPIO	34	RST_MSP	14	P6.5
8	P1.0	28	P7.3	33	P1.1	13	P1.6
9	P7.4	29	P7.5	32	P1.7	12	P2.6
10	P7.6	30	P7.7	31	P2.5	11	GPIO

Figure 14: MSP Simplified Pinout

Harvester Board Pinout								
Pin #	Func	Pin #	Func		Pin #	Func	Pin #	Func
1	NC	21	3V3		40	P5.4	20	GND
2		22	GND		39		19	P5.1
3		23	NC		38	P6.7	18	P5.5
4		24			37	P3.5	17	
5	P5.0	25			36		16	NC
6	P5.2	26			35		15	P6.4
7	P6.6	27			34		14	P6.5
8	P1.0	28	P7.3		33	P1.1	13	P1.6
9	P7.4	29	P7.5		32	P1.7	12	P2.6
10	P7.6	30	P7.7		31	P2.5	11	

Figure 15: Power Harvester Pinout

4.8.4 Breakout Board Design

The Breakout Board 18 was designed to test out some of the modifications that we would be making to the MSP Simplified board. This way we could order and test the Breakout Board which can be manufactured for a much cheaper cost. The Breakout Board is a pretty simple PCB that takes the form factor of the TI MSP430 Board 19 , and includes the following: 20P Boosterpack connectors, load switch circuitry, a 2 header pin for current measurements. Mostly, we wanted to ensure that the load switch would work for our needs, the connector spacing would match the stack up, and overall that fabrication of our board designed in KiCAD, and ordered through JLCPCB, would work as we expected. The breakdown of the estimated per board can be found here 22. It should be noted that this is only an estimate based on the components and boards that we ordered. This however does not include the cost of shipping for the components, as they were ordered through the ETG in Coover. The Breakout Board was tested and everything works as we expected, so we moved onto the MSP Simplified design. The Breakout Board is currently being used by one of the graduate researchers.

4.8.5 MSP Simplified Design

The MSP Simplified 23 was designed to replace the current development board that is being used as part of the BOB Node. The researchers wanted to minimize the board so that would fit better in our 4 layer stack-up, as well as get rid of any unused circuitry. The development board that was used is TI's MSP430fr5994. This board acts as the controller in the BOB Node

setup. The MSP430 board includes an MCU, and communicates with the CC1352, controls power distribution from the Power board, and will be monitored by the Sniffer Node that our team fully develops.

The MSP Simplified Removed the following logic from the MSP430 design: The debugger logic (we will attach separately to debugger when programming), additional LEDs, unused switches, a super capacitor, SD card, and external power connectors. The additions to the board include: change of form factor to roughly match that of the CC1352 radio board (it is actually a bit longer, as we included mounting holes), load switch circuitry to control power to the CC1352 board, mounting holes, a 90 degree debugger pin, a 2 pin connector on the power rail to potentially adjust capacitance, and DNP resistors for unused GPIOs (flexible for future use). The breakdown of the estimated cost per board can be found here: 30. It should be noted that this is only an estimate based on the components and boards that we ordered. This however does not include the cost of shipping for the components, as they were ordered through the ETG in Coover. The first revision of the board has been soldered and tested. We will order a second revision of the board during the beginning weeks of EE 492 2 to account for a mistake found in the pinout of the 40P connector in the first revision.

Sniffer Node Power Supply Design

The first step in designing the power supply is to estimate the battery capacity need; we performed several measurements on the CC1352R development board. TI provides a tool called Energy Trace on their development boards which can accurately measure the power usage of the board. The Sniffer board will be similar to the development board so the estimations made using this tool will give a good estimate for our design.

The development board was tested under two configurations. The first was with no transmissions being sent. This represents typical operation while listening for packets and recording data. The second was transmitting every 5ms. This represents the maximum frequency at which packets would be sent while transmitting data back toward the Host.

The test results for both configurations are shown in Figures 16 and 17 below. The data includes the maximum, minimum, and average power and current draws over a 10 second test window.

Transmitting every 5ms is much more frequent than the we plan to in the final design. Additionally, the Sniffer will be listening and not transmitting at least 50% of the time. So

System	
Time	10 sec
Energy	59.900 mJ
Power	
Mean	5.9900 mW
Min	4.6707 mW
Max	7.5945 mW
Voltage	
Mean	3.3000 V
Current	
Mean	1.8152 mA
Min	1.4154 mA
Max	2.3014 mA
Battery Life	CR2032: 4 day 14 hour (est.)

Figure 16: Power Usage Data When not Transmitting

System	
Time	10 sec
Energy	260.897 mJ
Power	
Mean	26.0897 mW
Min	24.1008 mW
Max	28.3495 mW
Voltage	
Mean	3.3000 V
Current	
Mean	7.9060 mA
Min	7.3033 mA
Max	8.5908 mA
Battery Life	CR2032: 1 day 1 hour (est.)

Figure 17: Power Usage Data When Transmitting Every 5ms

for a worst case estimate of the power usage of the Sniffer, we can assume that the Sniffer is in normal operation 50% of the time and transmitting every 5ms 50% of the time. The calculations for average power usage and energy needs can be calculated as shown below.

$$P_{avg} = 0.5(P_{normal}) + 0.5(P_{trans,5ms})$$

$$P_{avg} = 0.5(5.99) + 0.5(26.09) = 16.04mW$$

The Sniffer will need to run for up to 7 days continuously. The energy needed for 7 days of continuous operation can be estimated as follows.

$$E_{wk} = P_{avg}(7)(24)(60)(60) = 9.701kJ$$

Typically, battery capacity is stated in Ah (ampere-hours) or mAh (milliampere-hours). Ah is calculated by multiplying the number of amps by the hours of usage needed. Similarly, mAh can be calculated by substituting mA instead of A. The battery capacity needed for the Sniffer node is estimated below using the same assumptions that were made for the average power usage estimation.

$$capacity - needed = (0.5(I_{normal}) + 0.5(I_{trans,5ms}))(7)(24)$$

$$capacity - needed = ((0.5)(1.8152) + (0.5)(7.9060))(7)(24) = 816.581mAh$$

Rounding this value up, the worst-case scenario battery capacity needed to meet the requirements can be estimated as 820 mAh. However, the battery will not provide a constant voltage, but rather the voltage will decrease as the battery discharges. The power requirements of the Sniffer will remain the same so more current will be drawn from the battery. This results in a larger mAh need. The voltage affects can be inserted into the mAh estimate as follows.

$$capacity - needed = \left(\frac{P_{avg}}{V_{supplied}}\right)(7)(24) = \frac{2695}{V_{supplied}}mAh$$

To perfectly predict the needed battery capacity, the voltage curve of the battery would need to be modeled and the capacity equation above would need to be integrated with respect to the voltage. This would be tedious so instead a minimum and maximum mAh will be calculated given the minimum and maximum voltages of the battery and judgement will be used to determine if the capacity of the battery will be high enough. Then when the first revision is done, we will perform more testing to determine if the battery needs to be larger or not.

Our team generated three main design ideas for the battery portion of the power supply. The

three ideas are listed below with a rough cost estimation as well as the pros and cons of each idea. All costs are estimates based on research. All rechargeable solutions provide enough charging capacity to charge all batteries in one round of charging.

1. AA batteries which will be replaced when discharged

A typical AA or AAA battery has a voltage of 1.5V and discharges to about 1V before a strong drop-off. Typically, AA batteries have about 2500 mAh of capacity. Standard mounts for the batteries are cheap and readily available. Two AA batteries could be connected in series to provide 3V maximum and 2V minimum to the board. The minimum needed capacity for this setup would be 900 mAh and the maximum would be 1350 mAh. With a typical capacity of 2500 mAh, this setup should easily be able to meet the capacity requirements.

The cost estimation of this solution is as follows:

Item	Cost per Item	Quantity	Total Cost
AA Battery	\$0.80	2	\$1.60
Battery Mount	\$2.00	1	\$2.00

Table 4: Cost Estimation for Battery Solution 1

Total cost per board: \$3.60

Pros:

- Low complexity
- Cheap

Cons:

- Not very sustainable
- Marginal cost of replacing batteries (\$1.60 per board per replacement)

2. AA batteries which will be replaced when discharged

A typical AA rechargeable battery has a voltage of 1.2V and discharges to about 0.8V before a strong drop-off. Typically, these batteries have about 2000 - 2400 mAh of capacity. Standard mounts for the batteries are cheap and readily available. Three batteries could be connected in series to provide 3.6V maximum and a 2.4V supply to the board.

The minimum needed capacity for this setup would be 750 mAh and the maximum would be 1123 mAh. With a typical capacity of 2000-2500 mAh, this setup should easily be able to meet the capacity requirements.

The cost estimation of this solution is as follows:

Item	Cost per Item	Quantity	Total Cost
AA Recharg. Batt. (Amazon Basics)	\$1.10	30	\$33.00
Battery Mount	\$2.00	10	\$20.00
Battery chargers (cheap Amazon ones)	\$15.00	4	\$60.00

Table 5: Cost Estimation for Battery Solution 2

Total cost per board: \$11.03

Pros:

- Low complexity
- Sustainable
- No replacing batteries

Cons:

- Unknown Amazon battery quality (has good ratings though)
- External chargers needed

- Batteries must be kept track of

3. Rechargeable LIPO batteries

Many LIPO batteries have voltages that range from roughly 2.8V – 3.6V and can have various different capacities. It is not difficult to find LIPO cells that have the needed voltage range and have large capacities of over 1 Ah. These batteries easily satisfy the power requirements. The downside is that they need charging circuitry and protection circuitry. They also do not have as standard of sizes which will make them more difficult to mount. Our idea is to use a LIPO on each Sniffer with a protection and or management IC. Then we would also design separate charger boards which have charging ICs. This would allow the batteries to be taken out and charged. This increases modularity.

The cost estimation of this solution is as follows:

Item	Cost per Item	Quantity	Total Cost
LIPO	\$5.00	10	\$50.00
Battery Mount	\$3.00	10	\$30.00
Protection/Management ICs	\$0.50	10	\$5.00
Charger ICs and parts	\$1.00	10	\$10.00
Charger PCB	\$15.00	1	\$15.00

Table 6: Cost Estimation for Battery Solution 3

Total cost per board: \$11.00

Pros:

- High capacity allows for less recharging
- Sustainable
- No replacing batteries
- High voltage supplied

Cons:

- Complexity
- Longer design and testing time
- Non-standard sizes make mounting more difficult

Of these three options, our preference is number 3. However, we will review the options with our client before the final decision is made.

4.9 Technology Considerations

4.9.1 Two band-pass off

The two-band pass-off design uses the CC1352s' dual-band capabilities to switch the Sniffer between the listening to the BOBs and sending data to the next Sniffer or Host in the communication line. This process alternates the Sniffer between different bands, ensuring that the communication back to the Host does not interfere with the communication between BOBs. This is the method described above in Design 1. The advantages of this design are that we have the setup for the CC1352 completed and we know it can be done this way. The disadvantage is that it requires precise timing to ensure that the packets are not missed from the BOB nodes. It also gives challenges and throttle to the packet received and passed back to the Sink due to having less than 50% of the time that it can do it.

4.9.2 Internet Enabled Sniffer Node

This design described in Design 0 uses an ESP microcontroller such as an ESP01 to connect to a WIFI network to pass data directly back to the Host without the need for a Sink node. This design advantages are that it does not require a Sink node and does not need to have a predetermined path back to the Host allowing for a much larger test bed. The disadvantages are that it adds the complexity of a second microcontroller into the system and the complexity of connecting to the internet. The main disadvantage of it is the high energy requirements of it causing a large drain on the battery power of the Sniffer.

4.9.3 Two CC1352 Design

In this design we have two CC1352s built into the design one that transmits data back to the Sink node on the Sink node band and another that collects the communication data back from the BOB and the two communicate via UART or SPI. An advantage of this design is not having to switch between Bands and having to maintain exact timing to ensure packets are not dropped. The disadvantages are the additional cost, complexity, and power of having two CC1352s on the Sniffer.

4.9.4 Both bands running on the CC1352 Simultaneously

Late in the design phase of this semester, we found that we could use both sub-1-GHz and a 2.4GHz BLE on the CC1352s at the same time. In this design we would listen to the BOB nodes on sub-1-GHz and communicate back to the Sink with BLE. The advantage of this design is the simplicity of the design not having to deal with switching bands or a second microcontroller. The disadvantage is that we do not have a clear example to work with.

4.9.5 Single band channel filtering

This design makes use of multiple channels and filtering within a single Band for the CC1352. The design would have the CC1352 for the BOB have a small filtered range around its communication frequency and the Sniffer CC1352 would have a large area of filtering that would include the BOBs transmit frequency and would transmit outside of the BOB's CC1352s filtering range. The advantage of this design is that it only uses one band. The disadvantage of this design is that depending on the way the CC1352s filter it could disturb the BOBs and that packets could be missed.

4.10 Design Analysis

As mentioned in section 5.7, some design iterations have already taken place. Design 0 had planned on implementing a ESP8266 for WIFI communication between Sniffer nodes, but it was decided that adding the WIFI module would add too much power consumption and complexity to the design. The most recent design iteration, (Design 1) has not yet been

implemented, so it is not yet clear if the design will be successful or not. There is still some uncertainty over whether two communication bands of different frequencies can be run simultaneously on the Sniffer nodes. If it is possible to run two bands simultaneously then the Sniffers could receive data from the BOBs and send data to Sniffers farther down the communication network at the same time. But, if only one band is active, then Sniffers will be switched periodically between listening and sending modes to receive data from BOBs and send received data to the next Sniffer in the communication line.

Further iterations of the design will solidify which methods of communication will be used for receiving and transmitting data. Future iterations will also discuss changes made in the Sniffer hardware as well as changes to the physical makeup of the stack. The team will be making a custom Sniffer board in the future to optimize the hardware for the project. Some unused parts may be removed, and other parts may be added, so future implementations will discuss changes made. The Sniffer needs to be wirelessly powered for a week for testing, so a battery will have to be incorporated into the design. Future implementations will discuss how the addition of a battery will affect both the hardware and the test-bed stack (which will be composed of a Sniffer, BOB and battery).

5 TESTING

5.1 Unit Testing

5.1.1 Software

Software design, as a whole, is interrupt-based coding, so our units for testing consist primarily of the main interrupt sections for testing. We will accomplish testing with Code Composer Studio, making use of the debug tool within CCS and a Python script to provide sending and receiving of UART messages with the CC1352. Additionally, we will use a second CC1352 to verify radio communication interrupts.

UART interrupts: Within the Sink Sniffer node, it uses UART to communicate with the Host, running a Python script. Within this module, we have several various commands. That is described in the ID of the UART package. These commands provide different functions

and have to take different paths through the interrupt handler. To test these we will set breakpoints within each of these paths in the intended function and send a UART package with the package ID set to every possible path and make sure it is taken correctly.

Packet ID	Intended Path to Set Breakpoint
UART_SET_NODE_MESSAGES	A function that sets the node messages by sending a radio message to all the Sniffer nodes
UART_SET_NODE_LAMBDA	A function that sends the node Lambda for the test configuration to the Sniffer nodes
UART_START_EXPERIMENT	Function to send a start experiment to the Sniffer nodes via the radio
UART_RESET_EXPERIMENT	Function to send a reset experiment to the Sniffer nodes
UART_SET_NODE_CONFIG	A function that sets the node config by sending a radio message to all the Sniffer nodes
UART_HANDSHAKE	A function to complete the handshake with the Python script

Table 7: UART Unit Tests

Radio interrupts: Within the all Sniffer node, it uses a radio to communicate with the Sink and to listen to their respective BOB node. Within this interrupt set, we have several various commands, many of which are called by the UART IDs above. These commands are set in the payload of the Radio package. These commands provide different functions and take different paths through the interrupt handler. To test these, we will set breakpoints within each of these paths in the intended function and send a radio package with the payload set to every possible path and make sure it is taken correctly. We will send these messages with a CC1352 and ensure that the path is hit by setting a breakpoint within the function.

Payload	Intended Path to Set Breakpoint
RADIO_SET_NODE_MESSAGES	A function that sets the number of events for the Sniffer node to generate.
RADIO_SET_NODE_LAMBDA	A function that sets the node Lambda for the test in the Sniffer node
RADIO_START_EXPERIMENT	A function that starts the experiment on its respective BOB node.
RADIO_RESET_EXPERIMENT	Function to reset the experiment with its BOB node
RADIO_SET_NODE_CONFIG	Function to send the config to the BOB node
RADIO_HANDSHAKE	A function to complete the Radio handshake with the Sink Sniffer
TIME_SYNC	For non-Sink nodes, a message from the Sink is needed to maintain a synchronized clock.
TYPE_ACK	Function to save the acknowledgment back to the queue to be eventually sent back to the Sink.
RAW_DATA	Function to save the raw data from BOB back to the queue to be eventually sent back to the Sink.

Table 8: Radio Tests

*Sink Sniffer Node Only will have a acknowledge version of these packets

**Sink Sniffer Node Only will have an interrupt for this aspect to save it back to the Host.

Time Interrupt: The time interruption needs to occur precisely and urgently to achieve the pass of off the roles of data collection and data aggregation back to the Sniffer node. To test this, we will have the module send a message via UART to the Python script with the exact timing of the interrupt. We will use this test to make sure we can establish and maintain strict timing to ensure no packets are lost.

GPIO Interrupts: The Sniffer node collects data about the life cycle of the BOB node and its transmission patterns via GPIO pins on the Sniffer nodes. To track this and record it we will have interrupts on these pins to record the timing and send data on to the Host. Similar to the UART and Radio we will test it by setting breakpoints and then manually pulling pins up and down to ensure that the interrupt is triggered.

Pin	Record to Make
Powered On	Function to add a power on or power down event to the radio send queue
Easylink TX	Function to add a transmission event to the radio send queue
Event Drop	Function to add an event drop event to the radio send queue

Table 9: GPIO Interrupt Tests

Queue: The final unit for our software team to test is not an interrupt-based function but is a queue that has been previously mentioned that holds a backlog of events that need to be logged by being sent back to the Host. We will test this by setting a timer interrupt rate with a timer to the maximum rate at which packets will be added to a queue. Then, we will add packets to a queue, counting up to the maximum storage amount required of the queue. Once that is complete, we will iterate through the queue removing items from the queue at the required rate and ensuring that every item is accounted for.

5.1.2 Hardware

Below is a description of the general unit tests and process that will be taken for testing each new PCB that is part of the system. For the purposes of our senior design project, this will include 3 PCBs named as following: Breakout Board, MSP_Simplified, and Sniffer. These will be broken into their own sections to provide specific details of unit testing each board.

Step 1 of testing occurs before the board is fabricated. This includes checks that are made during the design process. The ERC, or electrical rules check, is used in KiCAD to verify correct schematic connections. This check will be performed throughout the schematic

design, and will need to be passed before the layout is generated. Next, DRC, design rule checking, will take place in the layout. In KiCAD, this will check for mismatches between the schematic and layout, as well as layout custom rules. These rules are provided by the fabrication company that you are ordering from and defined in the design tool by the designer. Design rules ensure that the board can be reliably manufactured and that it follows good design practices. These two checks are vital when designing a PCB.

Step 2 occurs after the board has been fabricated and populated. The designer visually inspects the board for defects with a microscope. Defects can be pins that do not have a solder connection to the pad, shorted pads, peaks in the solder, and burned components. Often the designer will find a number of issues in this step.

In step 3, the populated board is connected to a power supply and the supply current is measured. The power supply current is limited to a safe maximum to protect the board while allowing adequate current for normal operation. If the supply current jumps to the maximum allowed value, then there is likely a short in the board. This check serves to find major issues before the board is used and protects against damaging the board. The board software independent functionality can also be tested using power supplies, multimeters, oscilloscopes, and logic analyzers.

In step 4, the debugger is connected, and the board is programmed with any software it will run. Often, if there is an issue on the board, the debugger will not be able to successfully communicate with and program the board. This would indicate the need to go back to steps 2 and 3 to find the issue in the board.

In step 5, the software dependent elements of the board are tested by simulating the operating environment. For example, if a motion sensor was used on the board, then the board could be shaken and the data in software could be analyzed to see if motion is sensed or not. Initial tests are very basic and usually just establish communication with peripherals or basic functionality. Further testing and calibration are pursued later to verify that the elements work as expected.

Element Under Test	Testing Task	Expected Result
Solder Joints	Visually inspect the PCB with a microscope for any soldering issues.	Ensure none of the following are present: Shorts between pins, peaks of solder at connections, solder on pin but not connected to pad, component pins not aligned.
Components	Visually inspect the PCB (with a microscope for 0402 or smaller) for any issues with components or the board	Ensure none of the following are present: Leftover flux on board, burned or damaged discrete components, missing pins, melted plastic, damaged pads or traces, missing parts.
Power to ground isolation	Continuity test between power and ground	Should not see any measurable current exchange.
Power rail check	Power the 3.3V rail and GND rail with a power supply. Probe the 3V3 and GND test points with a multimeter.	The expected outputs are 3.3V and 0V, respectively. Significant margins (100s of mV) off this expected value should be evaluated further.
Header pin isolation	Run continuity checks between all neighboring pins with a multimeter.	The expected output is no beeping (this indicates connectivity) from the multimeter, when the nets are not connected on the schematic.

Table 10: Breakout Board Unit Testing

Element Under Test	Testing Task	Expected Result
EN Test for load switch	Provide 3.3V and GND power rails to the PCB with a power supply. Measure the VOUT voltage with a multimeter. Next, provide 1V on the enable pin and measure the VOUT voltage with a multimeter.	With no enable voltage provided we expect to see 0V on the output and with the enable voltage we expect approximately 3.3V (within 100mV) on the output.
Monitor Current output from the load switch	Using an ammeter, probe the open circuit provided on the board. This can be done either across the two-pin connector or the DNP resistor.	Should show very little current drawn from the load switch, as indicated on the datasheet.

Table 11: Breakout Board Unit Tests

Element Under Test	Testing Task	Expected Result
Solder Joints	Visually inspect the PCB with a microscope for any soldering issues.	Ensure none of the following are present: Shorts between pins, peaks of solder at connections, solder on pin but not connected to pad, component pins not aligned.
Components	Visually inspect the PCB (with a microscope for 0402 or smaller) for any issues with components or the board	Ensure none of the following are present: Leftover flux on board, burned or damaged discrete components, missing pins, melted plastic, damaged pads or traces, missing parts.
Power to ground isolation	Continuity test between power and ground	Should not see any measurable current exchange.

Table 12: MSP Simplified Unit Testing

Element Under Test	Testing Task	Expected Result
Power rail check	Power the 3.3V rail and GND rail with a power supply. Probe the 3V3 and GND test points with a multimeter.	The expected outputs are 3.3V and 0V, respectively. Significant margins (100s of mV) off this expected value should be evaluated further.
Header pin isolation	Run continuity checks between all neighboring pins with a multimeter.	The expected output is no beeping (this indicates connectivity) from the multimeter, when the nets are not connected on the schematic.
EN Test for load switch	Provide 3.3V and GND power rails to the PCB with a power supply. Measure the VOUT voltage with a multimeter. Next, provide 1V on the enable pin and measure the VOUT voltage with a multimeter.	With no enable voltage provided we expect to see 0V on the output and with the enable voltage we expect approximately 3.3V (within 100mV) on the output.
Hardware reset buttons	Work with Software to power up and flash the microcontroller. Depress and release the reset button (S2)	Software indicators show a successful reset, and that the device can continue normal operation after a reset.

Table 13: MSP Simplified Unit Tests

Element Under Test	Testing Task	Expected Result
Solder Joints	Visually inspect the PCB with a microscope for any soldering issues.	Ensure none of the following are present: Shorts between pins, peaks of solder at connections, solder on pin but not connected to pad, component pins not aligned.
Components	Visually inspect the PCB (with a microscope for 0402 or smaller) for any issues with components or the board	Ensure none of the following are present: Leftover flux on board, burned or damaged discrete components, missing pins, melted plastic, damaged pads or traces, missing parts.
Power to ground isolation	Continuity test between power and ground	Should not see any measurable current exchange.
Power rail check	Power the 3.3V rail and GND rail with a power supply. Probe the 3V3 and GND test points with a multimeter.	The expected outputs are 3.3V and 0V, respectively. Significant margins (100s of mV) off this expected value should be evaluated further.
Header pin isolation	Run continuity checks between all neighboring pins with a multimeter.	The expected output is no beeping (this indicates connectivity) from the multimeter, when the nets are not connected on the schematic.
Hardware reset buttons	Work with Software to power up and flash the microcontroller. Depress and release the reset button	Software indicators show a successful reset, and that the device can continue normal operation after a reset.

Table 14: Sniffer Unit Tests

5.2 Interface Testing

5.2.1 Software

Many software tests of interfaces between multiple software subsystems need to test the bandwidth of communication between multiple nodes. To be able to achieve the required recording goals of the project we must have certain data rates between aspects of the project. Similar to the unit tests we will use CCS, several CC1352s, and a python script to test these interfaces.

UART: To record the data at the Host, all data from the test needs to be communicated from the Sink to the Host. We will test this with a timer to create a UART data rate being sent from the CC1352 Sink to the Host computer at a rate of 1 packet every 1 ms. As packets are created, it will increase the counter to ensure a unique identifier for each packet. We can then go through all records and ensure all are accounted for.

Radio Transmit: Similar to UART, we will test the data rate, sending data back to Sink from a Sniffer node. We will set a timer to send data in our largest package size of 128 bits back at a desired rate of 1 packet every 5 ms with unique data and ensure that no packets are missed in receiving. This radio interface test will be conducted for both Sniffer to Sink and from Sniffer to Sniffer, as both paths need to maintain a high data rate.

5.2.2 Hardware:

Element Under Test	Testing Task	Expected Result
Debugger and Programmability of Microcontroller	Connect PCB to power. Connect the debugger to the computer and to the PCB. Set up any necessary parameters on the computer side. Flash the microcontroller with the code.	Debugger is able to establish communication with the microcontroller. Device is successfully flashed.
Antenna connection and functionality	Connect the antenna to the proper port. Power up and flash the device with a program that monitors communications. Set up an unused CC1352 board to send communications. Monitor the received communications of the new Sniffer.	Sniffer should receive RF signals. If no signals are received, there is a problem.
Test 3V3_CC line on the CC board	When the board stack is in place and initial unit testing is complete for the MSP_Simplified, probe the 3.3V line on the CC1352 with a voltmeter.	You should read approximately 3.3V on the multimeter for the power rail on the CC1352 board.

Table 15: MSP_Simplified Interface Testing

5.3 Integration Testing

5.3.1 Software

Time Sync: A major aspect of our system is the transition duties between multiple Sniffer nodes, which require them to maintain synchronized time. We will test the time synchronization system by letting multiple nodes run independently, with the Sink sending a time sync message out periodically. At the end of the day-long test, we will check the time of each node and ensure they are still within an acceptable margin of error.

Avoids Lockup: With all the interruptions that are occurring, there is the possibility of a lockup scenario. To test for this, we will take our Sniffer node and drive the GPIO pins high and low while simultaneously having radio transmissions incoming and having the timing interrupts active. We will ensure these all occur correctly via a log sent out via UART packets.

A similar testing set up will be done for the Sink Sniffer node setup, where it also has UART commands being sent to it from the Host.

Band handoff: The band handoff is required for the recording of data from BOBs and the aggregation of data back to the Host. To test the receiving/transmitting pass-off method for Sniffers. A pair of Sniffers will listen to a pair of BOBs. The Sniffers will have to alternate which Sniffer is listening to the BOBs and which is sending data received from the BOBs back to the Host. It's important to not lose data, so some overlap time will exist when both Sniffers are listening to BOBs. We will have a false BOB send out packets relatively constantly incrementing the payload, and we will make sure that no packet of a payload number is missed.

5.3.2 Hardware

Stack integration- Test signals both the I/O and power ports in the main Boosterpack connector that is distributed throughout the stack. Ensure one does not impact the performance of the entire stack.

Integration between the physical boards to ensure mechanical stability throughout the entirety of the stack.

Element Under Test	Testing Task	Expected Result
Stack Integration	Probe the power port and I/O connections in the boosterpack connector. The I/O will need to be run with a series of software tests to set and then measure the recorded I/O values. Ensure that there is no loss in power droop across the stack and no effects from I/O while other boards are running.	We expect to see the programmed I/O values as well as a consistent shared power rail across the stack. Isolation is expected across GPIOs that are being written on another board.
Mechanical Stack Integration	Test mechanical durability between the physical boards to ensure mechanical stability throughout the entirety of the stack.	We expect the stack to stay intact and operable through all mechanically strenuous conditions.

Table 16: MSP_Simplified Integration Testing

5.3.3 Sub-System

MSP430: To test the Custom MSP-430 board we will flash the program that is currently running the tests onto the new board with a different GPIO pinout changed in the software. We will then Run a current test with the custom msp-430 board and ensure that it functions the same as using a development board.

Custom Sniffer Board: Similar to the custom MSP-430 we will load the current test setup onto our custom board and update for our GPIO set and for only one BOB node and ensure it functions in tests correctly.

Custom BOB stack: With the updated code sets above for the alternative pinouts, we can run our full stack doing a test with the original code base, just our pinout and our hardware, ensuring the hardware works properly together.

5.4 System Testing

Running System level tests, we will use a variety of tools to accomplish it. We will continue to use CCS, Python script, and development boards. Additionally, we will build the tools of a CC1352 dev board running custom code to mimic the behavior of a BOB node with a hardcoded performance we can check for all performances in the logs. Additionally, we will make use of the current setup.

We will run a test with our mock BOB discussed above that will test all portions of the system: timers, GPIO interrupts, UART interrupts, radio communication, and queues. We can set that BOB to run a maximum packet rate of a packet every 5 ms. We will set up a test with the Host, Sink, and two Sniffers monitoring two BOBs. Then, we will expand it four Sniffers and 4 BOBs arrayed in a 2 by 2 checkered board setup.

To test the Sniffer setup in a less-known environment, we will set our test next to the current setup running a test and comparing the logs to ensure that they get the same logged data. We can receive the same wireless data being “sniffed” from the BOBs running and be connected into the power and transmit pins with the GPIO of the Sniffer. The data logged by this will then be mapped to the current systems log to ensure that they line up properly.

5.5 Regression Testing

Our regression testing will consist exclusively of the mock BOB node that will be designed for system testing. To ensure that future iterations of our design continue to fit requirements, the mock BOB node can have a test run on it, and if all the predetermined data is recorded, it will help ensure functionality does not get broken. It will continue to use the Python script, our testbed setup, and a CC1352 development board.

5.6 Acceptance Testing

The acceptance tests consist of two parts:

We will be given a simple test setup that the research assistants run on the current setup. We will run this test on our setup and ensure that our code configures things correctly and

that the code has the data output all recorded and organized in an acceptable way for the researchers.

The follow-up acceptance test will give our product setup with documentation to the research assistants and have them set up and run their test with the exclusive help of the documentation we wrote to ensure they will be able to make use of it once we graduate and move on and future research groups can use our infrastructure.

5.7 Results

Hardware Results: For our hardware, we performed unit tests on both of our PCBs that were manufactured. This includes the Breakout Board and the MSP Simplified. The Breakout Board passed all of the unit checks that we had listed in our test plan. The MSP Simplified did not meet all of the unit checks. We found issues with the design of the board as there was an incorrect footprint for one of the components, and the order of the pins was reversed for one of our 20P connectors. This makes it unusable in the hardware stack, and so a second revision will be ordered during the first few weeks of EE 492. This is outlined in our Gantt chart 2.

Through our testing so far, we have found a couple of important numbers we needed for decision making and design work.

- **Worst Case Packet Send Time:** 12 ms
- **Band-Switch Time Cost:** 30 ms

6 IMPLEMENTATION

Much implementation work can be seen in the design section

Our implementation plan is split into hardware and software

6.1 Hardware

1. Assemble MSP Simplified Board
2. Test MSP Simplified Board
3. Complete MSP Simplified second revision
4. Complete Sniffer schematic
5. Complete Sniffer layout
6. Assemble Sniffer PCB
7. Test Sniffer PCB
8. Complete second Sniffer revision, if needed

6.2 Software

1. Write Sink Node code
2. Write time sync function
3. Create mock BOB code for cc1352
4. Test BLE and Sub-1-GHz design
5. Update current Sniffer Code to use radio instead of UART
6. Create a queue structure for holding messages to be sent
7. If Band switching design create Band Switching Algorithm
8. Run Unit interface, and integration tests

Once both hardware and software are complete we will move on to system, acceptance, and regression testing.

7 PROFESSIONALISM

7.1 Areas of Responsibility

Areas of Responsibility	IEEE Definition	NSPE vs. IEEE Comparison
Work Competence	Conduct tasks with excellence, honesty, adherence to schedules, and a demonstration of professional expertise.	IEEE definition builds on the NSPE Canon definition for work competence by including quality and timeliness as important factors.
Financial Responsibility	Provide products and services that offer tangible value and are reasonably priced.	NSPE focuses more on the importance of loyalty to an employer while IEEE focus is more on the reasonable price of the product.
Communication Honesty	Present work with honesty, devoid of deception, and in a manner comprehensible to stakeholders.	NSPE states that only truthful statements should be made to the public, while IEEE definition emphasizes the understanding of the stakeholders specifically.
Health Safety, Well Being	Reduce hazards to the safety, health, and welfare of stakeholders.	NSPE states the health and safety of the public is paramount, while IEEE focuses on the health and safety of the stakeholders.
Property Ownership	Show regard for the property, ideas, and information belonging to clients and others.	NSPE uses the same definition as the one used for financial responsibility claiming the importance of loyalty to the employer. IEEE definition is more specific in the meaning of property.
Sustainability	Safeguard the environment and preserve natural resources both on a local and global scale.	The "Sustainability" area of responsibility does not exist for NSPE.
Social Responsibility	Create goods and services that contribute positively to society and local communities.	Both definitions are similar, but NSPE includes the statement that things should be done lawfully with the end goal of making the profession more useful and improving its reputation.

Table 17: Areas of Responsibility

7.2 Project Specific Professional Responsibility Areas

Work Competence - High. Work competence is the most important IEEE standard for our project. Given that Senior design is limited by assignment deadlines and graduation dates, it is important that we complete tasks in a timely manner. The success of the project is dependent on a large code base working with hardware over a scalable communication network, so it is important that our work is high quality to reduce the chances of errors in the design.

Financial Responsibility - Low. Based on the IEEE standard, there is no financial responsibility for this project, but the BOBs being tested by our design team may someday be used as a means of data collection, in which case the IEEE standard for financial responsibility would apply.

Communication Honesty - High. It is important that teammates are honest with each other and the clients of the project. Honesty will allow for faster progress as problems in the design process will be clear making it easier to find solutions.

Health Safety, Well Being - N/A. There are not any direct safety concerns for anyone involved in the project.

Property Ownership - Medium. Throughout the project team members will use circuit boards and software provided by the client. Team members should treat the equipment used with respect and avoid damaging the equipment.

Sustainability - N/A. There are no direct environmental threats from the project. The testing area will be contained to a single room, and should not have any affect on the environment.

Social Responsibility - N/A. The senior design project does not produce a good or service, so by the IEEE definition for social responsibility the standard is not applicable. It is important to note that the devices we are testing (the BOB communication network) could potentially become a product that has an affect on a community, but that is beyond the scope of the senior design project.

7.3 Most Applicable Professional Responsibility Area

The most applicable area of responsibility is work competence. Considering the constraints imposed by assignment deadlines and graduation dates for Senior design, it is crucial to ensure timely completion of tasks. The project's success relies on a substantial code base interacting with hardware across a scalable communication network, underscoring the significance of maintaining high-quality work to minimize the risk of design errors.

8 CLOSING MATERIAL

8.1 Discussion

The results we had from the semester of EE 491 include the completion of manufacturing the two PCBs, the EnergyTrace calculations on the CC1352, timing communications tests for Software, and the Unit testing for both the Breakout Board PCB and the MSP Simplified. A second revision of the MSP Simplified will be made during the next semester, as we found an incorrect footprint and inverted order of our Boosterpack pin connectors.

8.2 Conclusion

Looking ahead, we have a direction of where we are headed with the project, as well as how to get there. It will take us four to eight weeks to finish up our software and hardware design, at which point we will be able to document, test, and verify the product with the client. We are on pace to meet all of our requirements within the specified time frame.

REFERENCES

- [1] Texas Instruments, “Launch your design start@ ti launchpadtm sensortag kit with simplelinktm wireless mcu.” https://www.ti.com/lit/ug/swau127a/swau127a.pdf?ts=1701604662638&ref_url=https%253A%252F%252Fwww.ti.com%252Ftool%252FLPSTK-CC1352R, 2023. Accessed: Dec. 03, 2023.

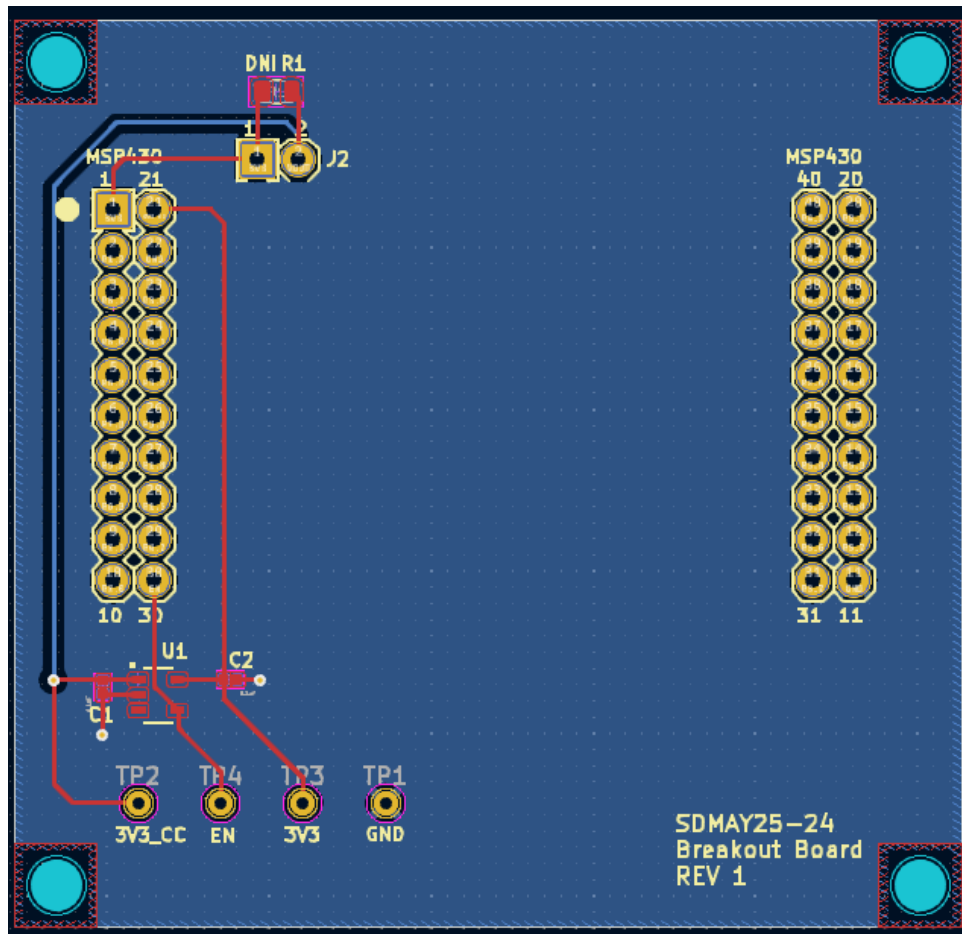


Figure 19: Breakout Board Layout

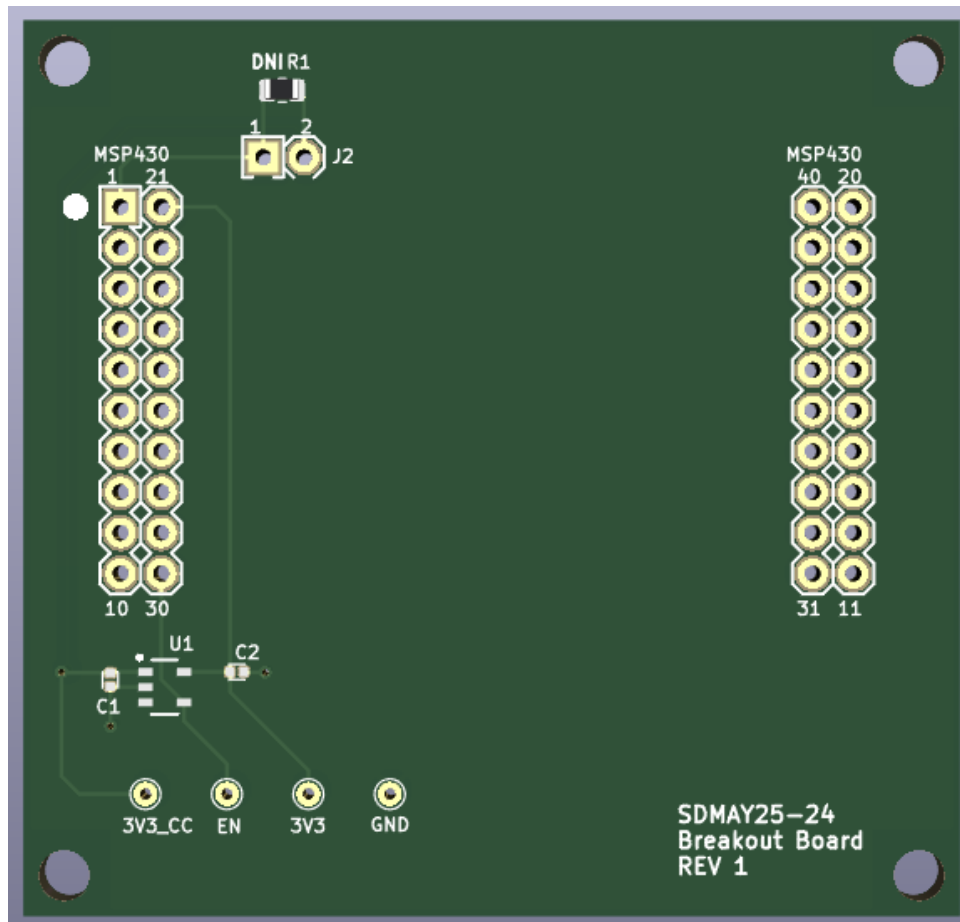


Figure 20: Breakout Board 3D Layout

	A	B	C	D	E	F	G	H	I	J	K
1											
2											
3											
4											
	SD25 2023 Breakout Board BOM										
5	Item #	Designator	Manufacturer	Mfg Part #	Description / Value	Package	Supplier	Link	Qty	Cost	Total Cost
6	1	U1	GLF Integrated Power	GLF1111	Power Switch/Driver P-Channel 2A	SOT-23-5L	DigiKey	https://www.digikey.com	3	0.33	0.99
7	2	C1, C2	Samsung Electro-Mechanics	CL05A104KASNNNC	CAP CER 0.1UF 25V X5R 0402	0402	DigiKey	https://www.digikey.com	10	0.01	0.1
8	3	J1	Samtec Inc.	SSW-110-03-G-D	CONN RCPT 20POS 0.1 GOLD PCB	-	DigiKey	https://www.digikey.com	4	3.89	15.56
9	4	J2	Molex	22122024	TH, Right Angle 2 position 0.100" (2.54mm)	-	DigiKey	https://www.digikey.com	2	0.77	1.54
10	5	R1	Stackpole Electronics	RMCF0805ZT0R00	RES 0 OHM JUMPER 1/8W 0805	0805	DigiKey	https://www.digikey.com	10	0.018	0.18
11									Total Cost		
12										18.37	

Figure 21: Breakout Board BOM Order

Cost for Single Board										
Item #	Designator	Manufacturer	Mfg Part #	Description / Value	Package	Supplier	Link	Qty	Cost	Total Cost
1	U1	GLF Integrated Power	GLF1111	Power Switch/Driver P-Channel 2A	SOT-23-5L	DigiKey	https://www.digikey.com	1	0.33	0.33
2	C1, C2	Samsung Electro-Mechanics	CL05A104KA5NNNC	CAP CER 0.1UF 25V X5R 0402	0402	DigiKey	https://www.digikey.com	2	0.01	0.02
3	J1	Samtec Inc.	SSW-110-03-G-D	CONN RCPT 20POS 0.1 GOLD PCB	-	DigiKey	https://www.digikey.com	2	3.89	7.78
4	J2	Molex	22122024	TH, Right Angle 2 position 0.100" (2.54mm)	-	DigiKey	https://www.digikey.com	1	0.77	0.77
5	R1	Stackpole Electronics	RMCF0805ZT0R00	RES 0 OHM JUMPER 1/8W 0805	0805	DigiKey	https://www.digikey.com	1	0.018	0.018
6	-	-	-	Board Fabrication	-	JLPCB	-	1	3.892	3.892
								Total Cost		12.48

Figure 22: Breakout Board Cost Per Board

9.2 Simplified MSP-430 Board

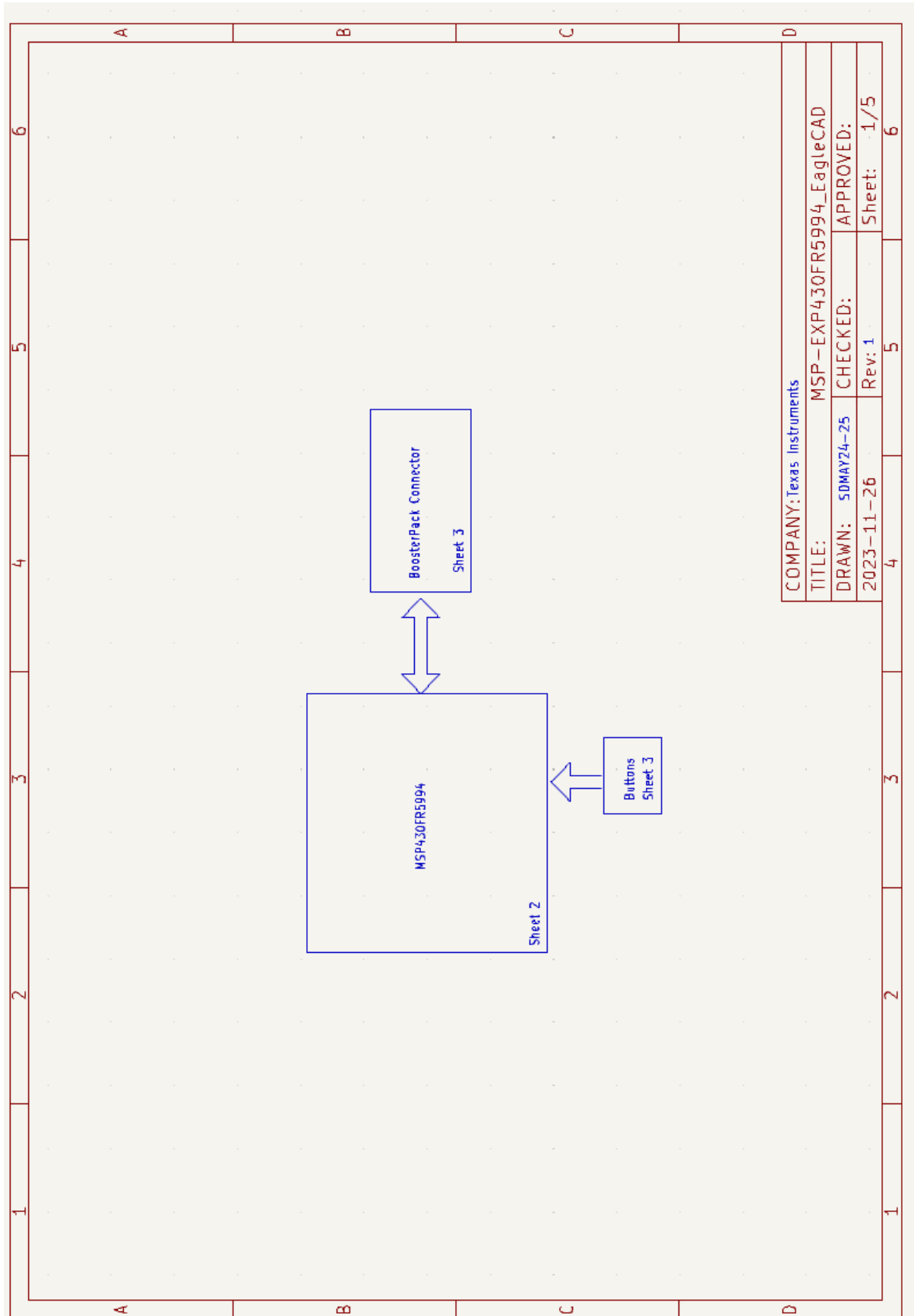


Figure 23: MSP Simplified Schematic Sheet 1

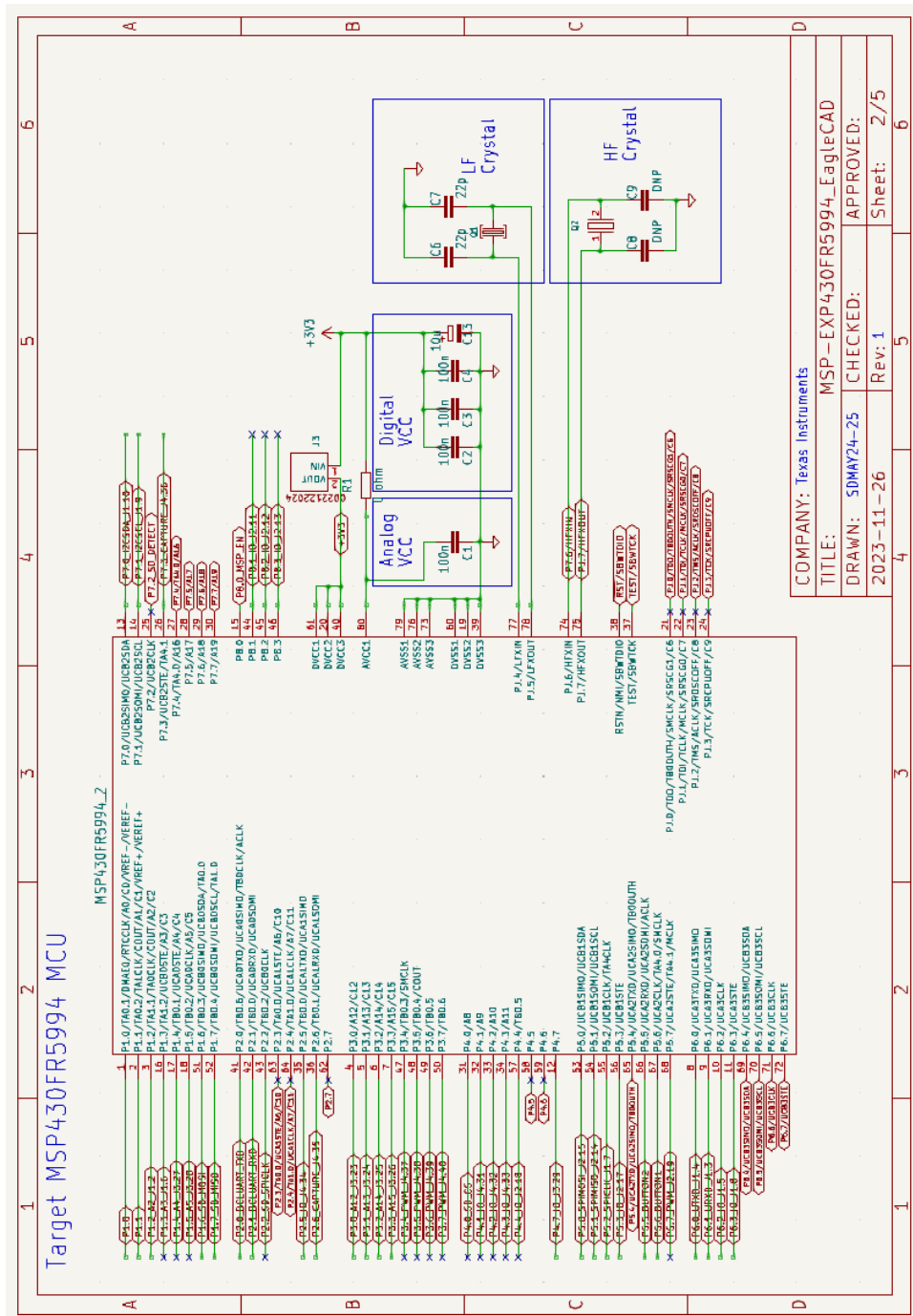


Figure 24: MSP Simplified Schematic Sheet 2

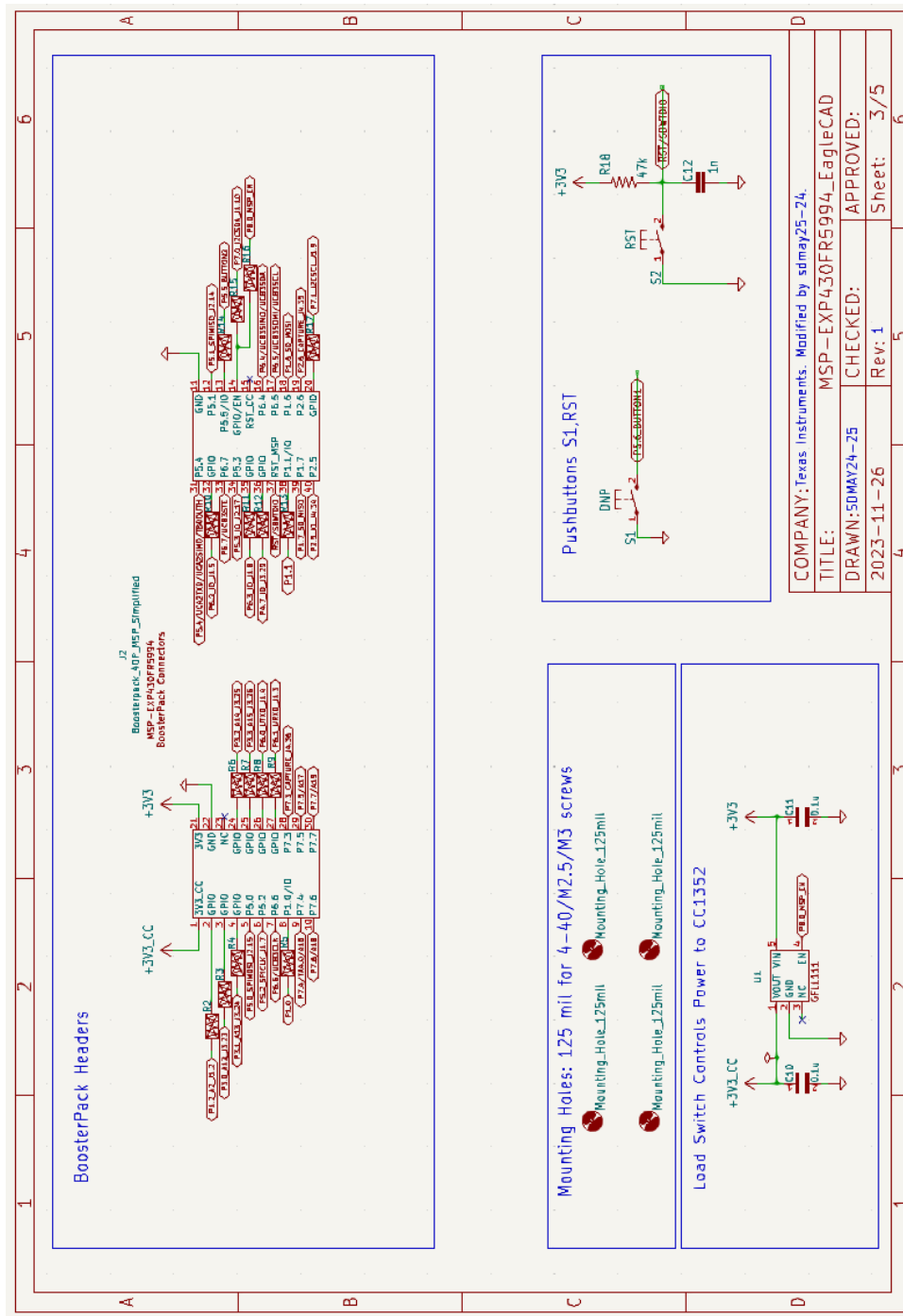
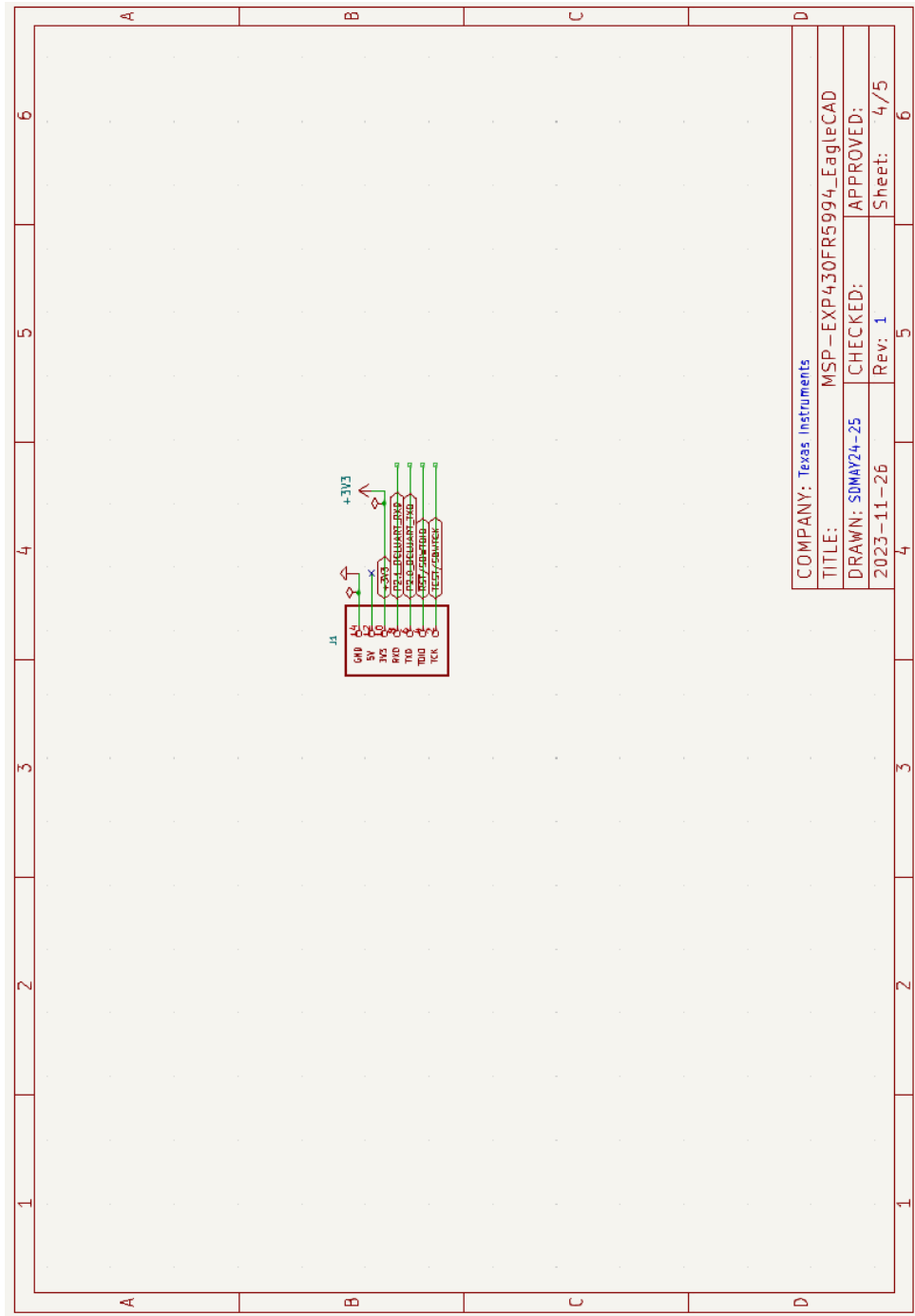


Figure 25: MSP Simplified Schematic Sheet 3



COMPANY: Texas Instruments	
TITLE:	MSP-EXP430FR5994_EagleCAD
DRAWN: SDMAY24-25	CHECKED:
2023-11-26	Rev: 1
	Sheet: 4/5

Figure 26: MSP Simplified Schematic Sheet 4

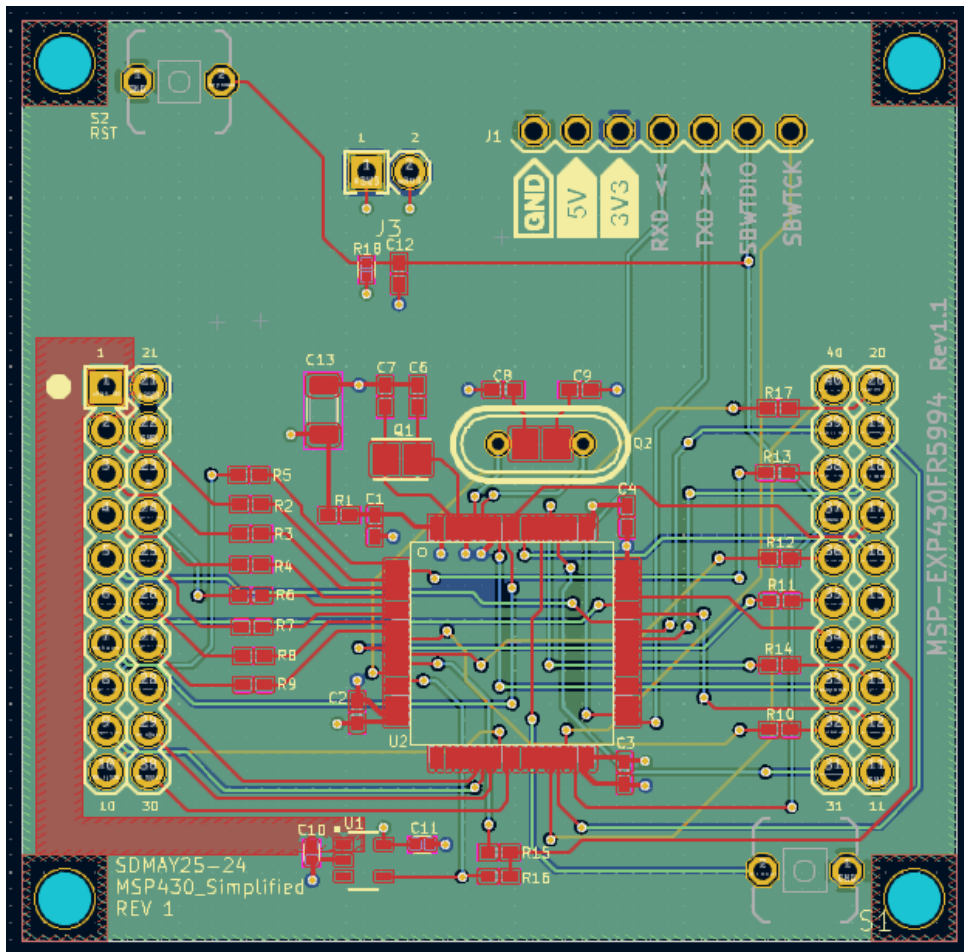


Figure 27: MSP Simplified Layout

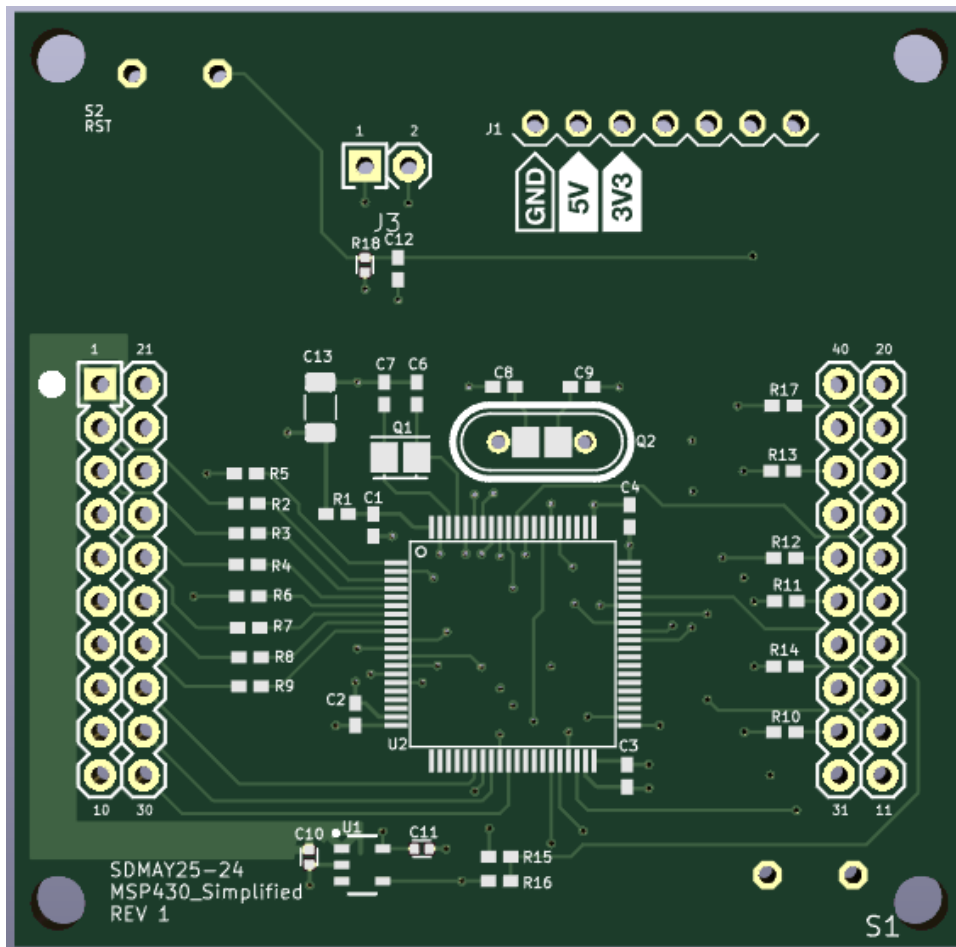


Figure 28: MSP Simplified 3D Layout

SD25 2023 MSP_Simplified Board BOM										
Item #	Designator	Manufacturer	Mfg Part #	Description / Value	Package	Supplier	Link	Qty	Cost	Total Cost
1	U1	GLF Integrated Power	GLF1111	Power Switch/Driver P-Channel 2A	SOT-23-5L	DigiKey	https://www.digikey.com/product-detail/en/1005X5R1A104M050BA/1005X5R1A104M050BA/1005X5R1A104M050BA	12	0.33	3.96
2	C1, C2, C3, C4, C10, C11	TDK Corporation	C1005X5R1A104M050BA	CAP CER 0.1UF 10V X5R 0402	0402	DigiKey	https://www.digikey.com/product-detail/en/1005X5R1A104M050BA/1005X5R1A104M050BA/1005X5R1A104M050BA	65	0.021	1.365
3	J2	Samtec Inc.	SSW-110-03-G-D	CONN RCPT 20POS 0.1 GOLD PCB	-	DigiKey	https://www.digikey.com/product-detail/en/SSW-110-03-G-D/SSW-110-03-G-D/SSW-110-03-G-D	2	3.89	7.78
4	J3	Molex	22122024	TH, Right Angle 2 position 0.100" (2.54mm)	-	DigiKey	https://www.digikey.com/product-detail/en/22122024/22122024/22122024	10	0.64	6.4
5	C6, C7	TDK Corporation	C1005C0G1H220J050BA	CAP CER 22PF 50V COG 0402	0402	DigiKey	https://www.digikey.com/product-detail/en/C1005C0G1H220J050BA/C1005C0G1H220J050BA/C1005C0G1H220J050BA	25	0.047	1.175
6	C12	TDK Corporation	C1005X7R1H102K050BA	CAP CER 1000PF 50V X7R 0402	0402	DigiKey	https://www.digikey.com/product-detail/en/C1005X7R1H102K050BA/C1005X7R1H102K050BA/C1005X7R1H102K050BA	12	0.051	0.612
7	C13	Murata Electronics	GRM155R61A106ME11D	CAP CER 10UF 10V X5R 0402	0402	DigiKey	https://www.digikey.com/product-detail/en/GRM155R61A106ME11D/GRM155R61A106ME11D/GRM155R61A106ME11D	12	0.091	1.092
8	J1	Sullins Connector Soluti	PRPC007SBAN-M71RC	CONN HEADER R/A 7POS 2.54MM	-	DigiKey	https://www.digikey.com/product-detail/en/PRPC007SBAN-M71RC/PRPC007SBAN-M71RC/PRPC007SBAN-M71RC	10	0.191	1.91
9	Q1	EPSON	FC-135R 32.7680KA-AO	CRYSTAL 32.7680KHZ 12.5PF SMD	-	DigiKey	https://www.digikey.com/product-detail/en/FC-135R-32.7680KA-AO/FC-135R-32.7680KA-AO/FC-135R-32.7680KA-AO	2	0.7	1.4
10	R1, R2, R3, R4, R5, R6, R	YAGEO	RC0402JR-070RL	RES 0 OHM JUMPER 1/16W 0402	0402	DigiKey	https://www.digikey.com/product-detail/en/RC0402JR-070RL/RC0402JR-070RL/RC0402JR-070RL	185	0.0045	0.8325
11	R18	YAGEO	RC0402FR-0747KL	RES 47K OHM 1%/16W 0402	0402	DigiKey	https://www.digikey.com/product-detail/en/RC0402FR-0747KL/RC0402FR-0747KL/RC0402FR-0747KL	12	0.015	0.18
12	U2	Texas Instruments	MSP430FR5994IPN	IC MCU 16BIT 256KB FRAM 80LQFP	-	Mouser	https://www.mouser.com/product-detail/MSP430FR5994IPN	12	11.27	135.24
13	Q2	DNP								
14	-	Würth Elektronik	609002134Z1	JUMPER W/TEST PNT 1X2PINS 2.54MM	-	DigiKey	https://www.digikey.com/product-detail/en/609002134Z1/609002134Z1/609002134Z1	10	0.31	3.1
15	S1,S2	E-Switch	TL59NF160Q	SWITCH TACTILE SPST-NO 0.05A 12V	-	DigiKey	https://www.digikey.com/product-detail/en/TL59NF160Q/TL59NF160Q/TL59NF160Q	12	0.284	3.408
16	J2 (trying another comp	Samtec Inc.	SSW-110-23-G-D	CONN RCPT 20POS 0.1 GOLD PCB	-	DigiKey	https://www.digikey.com/product-detail/en/SSW-110-23-G-D/SSW-110-23-G-D/SSW-110-23-G-D	2	5.71	11.42
25	**NOTE: Qty provided above is for 10 boards								Total Cost	179.8745

Figure 29: MSP Simplified BOM Order

Cost for single board												
Item #	Designator	Manufacturer	Mfg Part #	Description / Value	Package	Supplier	Link	Qty	Cost	Total Cost		
1	U1	GLF Integrated Power	GLF1111	Power Switch/Driver P-Channel 2A	SOT-23-5L	DigiKey	https://www.digikey.com/product-detail/en/1005X5R1A104M050BA/1005X5R1A104M050BA	1	0.33	0.33		
2	C1, C2, C3, C4, C10, C11	TDK Corporation	C1005X5R1A104M050BA	CAP CER 0.1UF 10V X5R 0402	0402	DigiKey	https://www.digikey.com/product-detail/en/1005X5R1A104M050BA/1005X5R1A104M050BA	6	0.021	0.126		
3	J2	Samtec Inc.	SSW-110-03-G-D	CONN RCPT 20POS 0.1 GOLD PCB	-	DigiKey	https://www.samtec.com/product/SSW-110-03-G-D	2	3.89	7.78		
4	J3	Molex	22122024	TH, Right Angle 2 position 0.100" (2.54mm)	-	DigiKey	https://www.molex.com/catalog/22122024	1	0.64	0.64		
5	C6, C7	TDK Corporation	C1005C0G1H220J050BA	CAP CER 22PF 50V C0G 0402	0402	DigiKey	https://www.digikey.com/product-detail/en/1005C0G1H220J050BA/1005C0G1H220J050BA	2	0.047	0.094		
6	C12	TDK Corporation	C1005X7R1H102K050BA	CAP CER 1000PF 50V X7R 0402	0402	DigiKey	https://www.digikey.com/product-detail/en/1005X7R1H102K050BA/1005X7R1H102K050BA	1	0.051	0.051		
7	C13	Murata Electronics	GRM155R61A106ME11D	CAP CER 10UF 10V X5R 0402	0402	DigiKey	https://www.murata.com/en-us/products/capacitors/MLCC/GRM155R61A106ME11D	1	0.091	0.091		
8	J1	Sullins Connector Solutions	PRPC007SBAH-M71RC	CONN HEADER R/A 7POS 2.54MM	-	DigiKey	https://www.sullins.com/products/PRPC007SBAH-M71RC	1	0.191	0.191		
9	Q1	EPSON	FC-135R 32.7680KA-A0	CRYSTAL 32.7680KHZ 12.5PF SMD	-	DigiKey	https://www.epson.com/en-us/products/FC-135R-32.7680KA-A0	1	0.7	0.7		
10	R1, R2, R3, R4, R5, R6, R7	YAGEO	RC0402JR-070RL	RES 0 OHM JUMPER 1/16W 0402	0402	DigiKey	https://www.yageo.com/product-detail/en/RC0402JR-070RL/RC0402JR-070RL	17	0.0045	0.0765		
11	R18	YAGEO	RC0402FR-0747KL	RES 47K OHM 1% 1/16W 0402	0402	DigiKey	https://www.yageo.com/product-detail/en/RC0402FR-0747KL/RC0402FR-0747KL	1	0.015	0.015		
12	U2	Texas Instruments	MSP430FR5994IPN	IC MCU 16BIT 256KB FRAM 80LQFP	-	Mouser	https://www.ti.com/product/MSP430FR5994	1	11.27	11.27		
13	Q2	DNP	-	-	-	-	-	-	-	-		
14	-	Würth Elektronik	60900213421	JUMPER W/TEST PNT 1X2PINS 2.54MM	-	DigiKey	https://www.wuerth Elektronik.com/product-detail/en/60900213421/60900213421	1	0.31	0.31		
15	S1,S2	E-Switch	TL59NF160Q	SWITCH TACTILE SPST-NO 0.05A 12V	-	DigiKey	https://www.e-switch.com/product-detail/en/TL59NF160Q/TL59NF160Q	2	0.284	0.568		
16	J2 (trying another comp	Samtec Inc.	SSW-110-23-G-D	CONN RCPT 20POS 0.1 GOLD PCB	-	DigiKey	https://www.samtec.com/product/SSW-110-23-G-D	0	5.71	0		
17	-	-	-	PCB Fabrication	-	JLPCB	-	1	4.96	4.96		
									Total Cost		27.2025	

Figure 30: MSP Simplified Cost Per One Board

9.3 Team Contract

Team Contract: _sdmay24-25__

TEAM MEMBERS

1. Thomas Gaul
2. Matthew Crabb
3. Spencer Sutton
4. Tori Kittleson
5. Ian Hollingworth

TEAM PROCEDURES

Regular Team Meetings

- Team meeting: Wednesdays at 7:30 pm in person at TLA.
- Client/advisor meeting: Tuesdays at 8:30 am in person at Durham 353 Conference room.
- TA meeting (Thomas, Matt, Tori): Wednesdays at 2:30 pm in Senior Design Lab.
- TA meeting (Spencer, Ian): Fridays at 1 pm in Senior Design Lab.
- Workday Monday 3:30 PM in the TLA

Communication

- Teams account for updates, reminders, and scheduling issues within the group and with the research team.
- Teams account for updates with the client/advisor.

Decision-making Policy

The majority vote for the final decision, focusing on consensus when possible. Diverted opinions can be discussed with the client/advisor for advice.

Record Keeping

Ian and Matthew will handle shared document redundancy on Teams.

PARTICIPATION EXPECTATIONS

Attendance and Punctuality

Attend when possible, communicate scheduling conflicts, catch up on missed meetings, and notify if running substantially late.

Responsibility for Assignments

Complete tasks on time, give warnings if deadlines might be missed, and avoid procrastination.

Communication with Team Members

Respond to team messages within 24 hours on weekdays, 36 hours on weekends, and a reasonable time on breaks.

Commitment to Team Decisions

Put full effort into assignments and ensure fair work distribution.

LEADERSHIP

Roles

- Thomas: Lead, Technical software
- Matt: Technical hardware
- Tori: Technical hardware
- Ian: Technical electrical systems
- Spencer: Technical software

Supporting and Guiding Work

Weekly meetings and sub-meets for team members working on project portions together.

Recognizing Contributions

Weekly meetings to discuss progress and record progress in weekly deliverables.

COLLABORATION AND INCLUSION

Team Member Skills

- Ian Hollingworth: C programming, PCB design, Digital circuit design, Verilog, Cadence
- Tori Kittleson: Hardware circuit design, Cadence, Altium PCB design
- Matthew Crabb: C Programming, circuit design, PCB layout, Cadence, Altium, MATLAB, PCB debugging, circuit measurement

- Thomas Gaul: Embedded Systems, hardware design, firmware experience
- Spencer Sutton: Embedded Systems, Digital Logic/microprocessors, Cadence, Physics

Encouraging Contributions

Weekly meetings for idea exchange and progress presentation.

Resolving Collaboration Issues

Bring up issues in weekly team meetings or with the team lead or Dr. Duwe if uncomfortable.

GOAL-SETTING, PLANNING, AND EXECUTION

Team Goals

- Concrete plan and presentation for project completion in the second semester.
- First revision of hardware completed and ordered for a second revision next semester, with completed testing code.

Planning and Assigning Work

Volunteer for tasks and select the most qualified individual if no one is particularly interested.

Keeping on Task

Create weekly goals and plan work on a week-by-week basis for the remainder of the semester.

CONSEQUENCES FOR NOT ADHERING TO TEAM CONTRACT

Handling Infractions

Discuss conflicts as a team and try to resolve to fit the needs of all team members.

Continued Infractions

If problems persist, discuss with the client/advisor for suggestions to adjust to the team's needs.

ACKNOWLEDGMENT

Confirmation of Understanding

1. Tori Kittleson DATE 9/8/23
2. Ian Hollingworth DATE 9/8/23
3. Matthew Crabb DATE 9/8/23
4. Thomas Gaul DATE 9/8/23
5. Spencer Sutton DATE 9/8/23