I started at Iowa State in Fall 2022 with a strong interest in electronics and a solid math start from taking Calculus 1 and 2 in high school. My dad is an electrical maintenance technician, and growing up around troubleshooting and hands-on electronics shaped how I approached engineering. Early on, I was more comfortable building and experimenting than I was explaining the theory behind what I was doing. That gap closed over time as my coursework gave me the models to understand what I was building, and as my projects forced me to communicate clearly, document decisions, and justify tradeoffs. The most significant shift in my mindset has been a commitment to consistency and responsibility in documentation. I used to solve problems, move on, and lose the details that mattered. I've developed a system that treats documentation as part of the engineering work itself, as documentation is how a system is proven to be correct, and ensures the same mistakes do not repeat later.

The three experiences that best represent my growth are my home thermostat project, the CPRE 2880 CyBot project, and my ongoing part-time role at John Deere. Together, they show how my education at Iowa State has prepared me to design systems, communicate technical work, recognize ethical and professional responsibility, and continuously learn new knowledge when requirements change.

1. Embracing the Bigger Picture

My thermostat project pushed me to think like a systems engineer. It was not enough to make it "work." I had to define what "work" meant in terms of safety, reliability, and user needs. I designed and integrated a microcontroller on a custom PCB to control my apartment thermostat and connected it to Home Assistant for remote access and automation. That meant thinking about electrical interfaces, failure modes, and long-term stability. I had to treat the thermostat as a real system that interacts with a real environment, not a controlled lab setup.

The CyBot project reinforced the same lesson in a team context. A robot that "kind of works" is not helpful if it cannot produce consistent data or if the team cannot understand what is happening when it fails. My role on the team focused on embedded system communication and sensor data evaluation. That role forced me to see how each subsystem affects the rest of the robot. Minor issues in communication timing or noisy sensor data can cascade into bigger failures in navigation and autonomy. This experience strengthened my ability to recognize a problem at the system level, break it into testable parts, and work with teammates to resolve it.

At John Deere, the "bigger picture" is always present because real work impacts other teams and real products. I spend a lot of time developing and correcting system requirement documentation so software teams and technical writing teams can use it. That work taught me that clarity is an engineering responsibility. Ambiguous requirements create wasted time, incorrect implementation, and potential safety or reliability issues downstream. Iowa State gave me the technical foundation, but these experiences taught me how technical work connects to people, processes, and outcomes.

2. Beyond the Classroom: Tapping into the Richness of Resources

All three anchor experiences required learning beyond lecture content. For the thermostat project, I relied heavily on external resources like component datasheets, application notes, wiring guidance, and Home Assistant documentation. I had to use these sources to select components, understand operating limits, and avoid design choices that could lead to unstable behavior. The project also showed me how valuable online learning communities are. Online documentation, tutorials, and forum discussions helped me validate ideas and troubleshoot issues faster than trial-and-error alone.

For the CyBot, the external “resources” were often a mix of references and people. When sensor readings were inconsistent due to the lab environment, I had to research software-side filtering strategies and compare them against what we learned in class. I also learned from peers by watching how other teams tested, what they struggled with, and what solutions did not hold up in practice. Seeing repeated failure modes across teams made it evident that good testing practices and solid data handling are just as important as writing code that compiles.

At John Deere, I learn by reading internal documentation standards, reviewing how experienced engineers write and structure requirements, and iterating based on feedback. The work environment reinforces that learning is not optional. If you do not understand a system boundary, a requirement dependency, or a terminology convention, the output will be wrong. This role has improved my ability to find and synthesize information, and to produce documentation that another team can implement without guessing.

3. Embracing Lifelong Learning Beyond the Classroom

My learning at Iowa State has extended into professional development and technical curiosity. I have attended guest lectures and company events that expanded my view of engineering beyond the classroom. Those events helped me connect coursework topics to real systems and real industry priorities. They also helped shape my direction toward embedded systems, RF topics, and signal processing.

The thermostat project is an example of learning driven by need. I taught myself Fusion 360 to design and fabricate a PCB. It took time and persistence, but it built a skill I still use today, including in my senior design work. That experience also improved my confidence in learning new tools independently. I learned that the first version would be slow and imperfect, but progress compounds when you stick with it.

4. Pioneering Growth Through Adaptation

Adaptation has been a recurring theme across these experiences. In the thermostat project, I made a major mistake by not properly defining the system's input early enough. I forgot that many HVAC control systems operate on 24VAC rather than DC. That misunderstanding led to damaging a voltage regulator. The technical fix was simple compared to the change it forced.

After that, I started taking requirements capture more seriously. I began documenting assumptions, verifying interfaces before powering on the hardware, and conducting a second review of gathered information before committing to a design.

The CyBot project also required adaptation. The lab environment affected sensor readings for many teams, including ours. Rather than accept unreliable data, I focused on improving the software handling of sensor input using filtering. This was a direct example of adapting engineering knowledge to real constraints. The environment was not going to change, so the system had to become more robust.

At John Deere, adaptation shows up as continuous iteration. Requirements evolve. Stakeholders need to shift. Documentation must keep pace and stay consistent across teams. I have learned to treat all feedback as an opportunity to improve. That mindset is directly connected to how I now approach both engineering work and communication.

5. Crafting a Narrative of Growth

If I were to restart my undergraduate journey, I would build stronger networking and study habits earlier. Early semesters were more isolated than they needed to be, and that limited opportunities. I would also take office hours and peer support more seriously from the start, especially in classes that felt “busy” or outside my main interests. Over time, I learned that the classes you underestimate often improve your thinking in unexpected ways, especially in ethics, communication, and system-level reasoning.

I would also push myself to expand my circle beyond proximity-based groups. Diverse teams create stronger results because people bring different strengths, and you learn faster when you are exposed to other approaches. My recent experiences have shown me that engineering is a team sport, and the quality of collaboration often determines the quality of the outcome.

6. A Glimpse into the Acquisition of Knowledge

A consistent pattern in my growth has been improving my learning. Earlier in my degree, I would solve a problem and move on without reflecting on what I had learned. Now my learning loop is more structured. I gather information, implement, test, identify failure points, and iterate until it meets requirements. I also write down questions as I work so I can follow up later with a TA, professor, teammate, or documentation source.

This shift is visible in both my personal projects and professional work. The thermostat project forced me to learn quickly, revise designs, and integrate new information as the system evolved. At John Deere, I have learned that asking questions early prevents major rework later. I used to worry about asking “dumb” questions however it’s better to treat questions as a quality tool. If something is unclear, it is better to clarify it immediately than to build an entire solution on an assumption.

7. Transformative Applications of Knowledge

The CyBot project is my clearest example of translating theory into practical results. In CPRE 2880, we worked with embedded systems at a level where timing, communication, and sensor handling directly affected system behavior. I applied filtering concepts by implementing a Butterworth low-pass filter in software to improve the usability of a noisy sensor. A Butterworth filter is designed to have a flat passband response, making it helpful in reducing high-frequency noise without distorting the primary signal more than necessary.

That filtering improved the reliability of our sensor readings compared to what we saw before. We verified this through repeated tests at different distances and conditions, comparing stability and consistency across trials. Better sensor data enabled better decisions in the control logic, improving overall system functionality. That experience taught me that, smaller targeted improvements make the whole system more dependable.

8. Evolution of Learning Strategies

My learning strategies have changed from reactive to intentional. Early on, I focused on getting through assignments. Now I focus on building reusable understanding. Documentation is a key part of this. Writing down what I tried, what failed, and why it failed has improved my performance in labs, projects, and professional work. It also improved my communication because I can explain what I did with evidence instead of relying on memory.

My communication growth is tied to this shift. I regularly explain technical topics to non-technical people because I enjoy talking about what I am learning. However, I have learned to check for understanding and adjust the level of detail. One meaningful example was explaining my thermostat project to a peer mentor with no technical background. I simplified the system description, checked for comprehension, and focused on purpose and behavior rather than technical jargon. That is the same communication skill I now use in requirements writing at John Deere. The audience changes, but the goal stays the same: make the information usable.

9. The Path Ahead: Continuous Development

My immediate plan is to continue my education toward a master's degree while deepening my skills in embedded systems, RF, and digital signal processing. I also want to keep improving my networking and collaboration skills because engineering success depends on both technical ability and the ability to work effectively with others.

In the next 6 to 12 months, my priorities are to refine my PCB design skills, build more profound DSP intuition through practice, and expand my professional connections. My plan for lifelong learning combines structured education through graduate school with continuous learning through projects, feedback, seminars, and on-the-job growth. The experiences that shaped me most at Iowa State all share one theme: progress comes from building, testing, documenting, and iterating with purpose. That is the approach I plan to carry forward as I continue developing as an engineer.