

Kaitlyn Redman

ME 476C

HW03: Self-Learning

2/28/2025

Arduino Self-Learning

For my self-learning experience, I explored coding and wiring with Arduino. I chose to learn how to use Arduino because of its ability to create interactive electronic projects with hardware and software components. Arduino seemed like the perfect starting point for a hands-on learning experience. I used an online simulation [1] platform to familiarize myself with the Arduino environment. This allowed me to experiment with different circuits and code without the risk of damaging any hardware. I followed several YouTube tutorials [2] that covered the basics of Arduino programming and wiring components. These tutorials provided step-by-step guidance, which made it easier to grasp fundamental concepts such as digital and analog inputs/outputs, loops, and conditional statements. One of the first projects I completed was the classic "Blink" program [3], which made an LED turn on and off at set intervals; the screenshot of the wiring and code for this simple tutorial can be found in the Appendix (Figure 1). While simple, this tutorial helped me understand how to write code for the Arduino board. I then moved on to more complex projects, such as controlling LEDs with a button, reading temperature data from a sensor, and experimenting with servo motor control. These projects allowed me to practice writing and modifying code while learning how different components interact with the microcontroller. However, I encountered several challenges along the way. My circuits sometimes didn't work as expected, and I had to troubleshoot by checking wiring connections and debugging my code. Online videos and Arduino tutorials became very valuable resources whenever I got stuck. Now that I have a solid understanding of the basics, I want to take my Arduino skills further by working on more advanced projects.

Learning Arduino develops key design attributes, particularly problem-solving, creativity, and user-centered thinking. One of the most important skills I learned is system thinking, as working with Arduino requires understanding how hardware and software interact with each other. This helps in designing complex systems where different components work together efficiently. Arduino enhances problem-solving skills by encouraging an iterative approach to troubleshooting. Debugging circuits and code requires testing, identifying errors, and refining

designs to improve functionality. Another essential design attribute developed through Arduino is creativity and innovation. Since Arduino allows for open-ended experimentation, learners are encouraged to find creative solutions to technical challenges and prototype unique projects. This ties into user-centered design, as many Arduino projects focus on real-world applications, such as assistive technology. Arduino supports rapid prototyping, allowing for quick iterations and modifications based on testing results. This accelerates development time and leads to more refined designs. Another key design attribute developed through Arduino is efficiency in design. The simulation's limited memory and processing power encourage learners to write optimized code and create circuits that use minimal resources. Arduino promotes interdisciplinary thinking, as it sits at the intersection of electronics, programming, and mechanical design. This multidisciplinary mindset is valuable in fields ranging from robotics to product development. Learning Arduino strengthens design skills by encouraging experimentation, iteration, and user-focused innovation. It equips learners to think critically about how technology is designed and applied, making it an excellent foundation for future projects in engineering, interactive design, and automation.

For my capstone project, I applied the skills I learned in Arduino by coding and wiring the current sensor for the dynamometer in the Arduino simulation. Throughout my self-learning experience, I have developed a strong foundation in working with sensors, debugging circuits, and writing efficient code, which proved invaluable when implementing the current sensor in my project. Using the Arduino simulation, and a how-to video [4], I was able to test and refine my current sensor circuit design. Unfortunately, I could not find the exact sensor in the online simulation, so I asked ChatGPT [5] which of the available sensors was most like the sensor I wanted to use. I wrote the code to read the sensor's output, process the data, and display it in real-time ensuring that the dynamometer could accurately measure current variations. Through this process, I reinforced my understanding of analog inputs, data conversion, and serial communication, all essential for working with microcontrollers. By applying problem-solving strategies learned from my earlier Arduino projects, I identified and corrected errors efficiently. This experience not only strengthened my technical skills but also highlighted the importance of precision in sensor integration. By leveraging Arduino's simulation tools and coding capabilities, I created a reliable system for measuring current in the dynamometer, contributing to the overall success of my capstone project; the screenshot of my final coding and wiring can be found in the

Appendix (Figure 2). This hands-on application of Arduino reinforced my confidence in designing and implementing real-world electronic solutions.

Appendix:

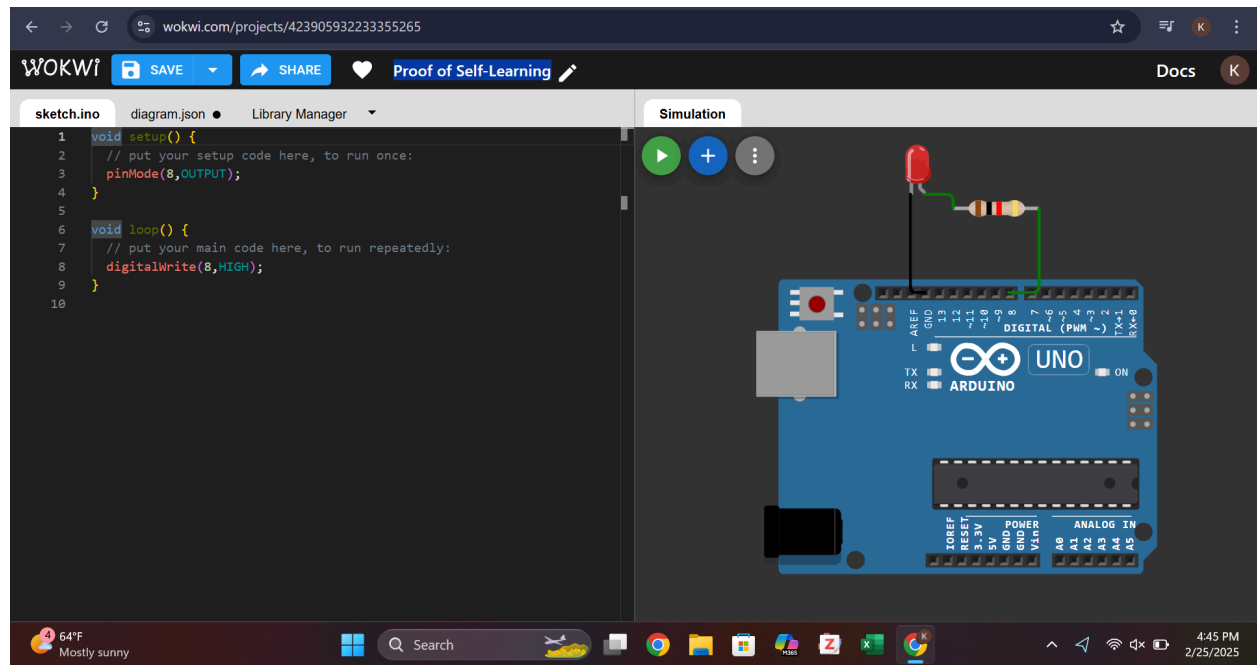


Figure 1: Proof of Self Learning - Classic “Blink” Programming and Wiring

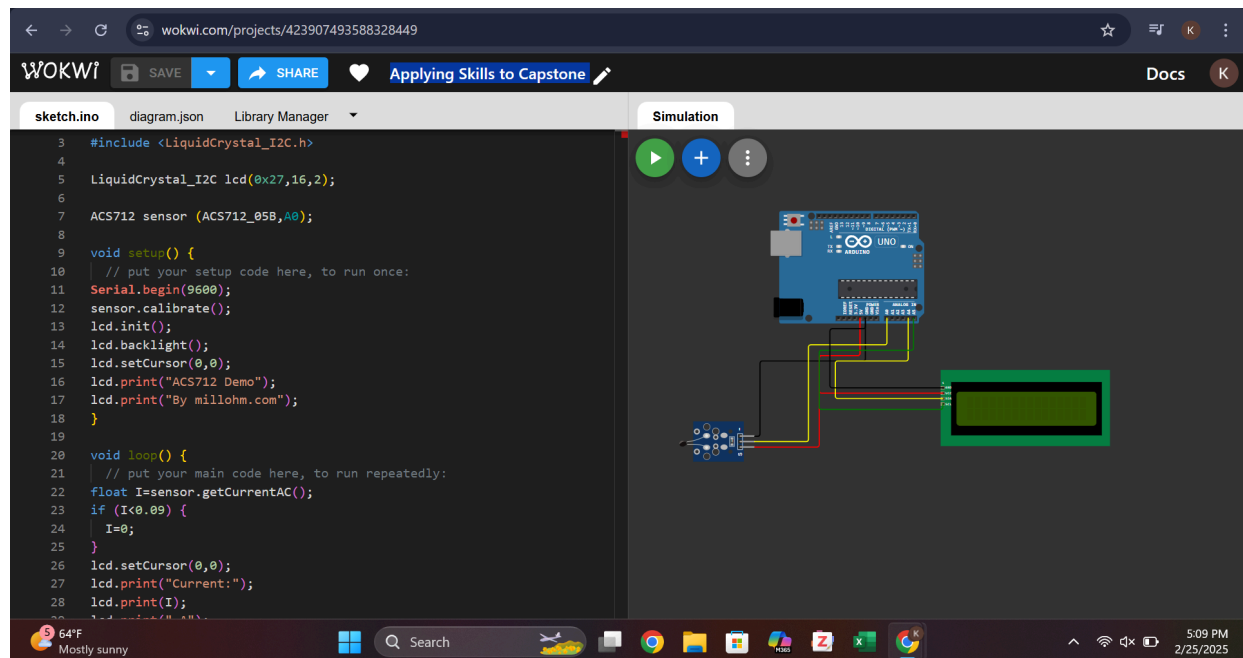


Figure 2: Applying Skills to Capstone - Coding and wiring for Current Sensor on Dynamometer

References:

[1] Arduino on Wokwi - Online ESP32, STM32, Arduino Simulator, <https://wokwi.com/arduino> (accessed Feb. 24, 2025).

[2] “Wokwi - Arduino tutorial,” YouTube, https://www.youtube.com/playlist?list=PLfdwhnay89Kn3sd6n_kGYg70cWQKG1n6j (accessed Feb. 25, 2025).

[3] “LESSON 1: How To Easily Code LED By Using Arduino Simulator”, YouTube, https://www.youtube.com/watch?v=w9_UrWdPWEO&list=PLfdwhnay89Kn3sd6n_kGYg70cWQKG1n6j&index=1 (accessed Feb. 25, 2025).

[4] “ACS712 Current Sensor Tutorial with Arduino”, YouTube, <https://www.youtube.com/watch?v=d6MnA4aPDag> (accessed Feb. 25, 2025).

[5] Chatgpt, <https://chatgpt.com/> (accessed Feb. 25, 2025).