2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC

Host Sponsor UNC Charlotte

Paper ID #52544

# GitHub Repository as Handbook, Jupyter Notebook as Report: A Format Redesign for the Electrical Engineering II Lab

**Dr. Yue Cao, Syracuse University**

Yue Cao is an Assistant Teaching Professor in the Department of Electrical Engineering and Computer Science at Syracuse University. He received his Ph.D. in Electrical and Computer Engineering from Purdue University in 2024. His doctoral advisor is Prof. C.S. George Lee. His areas of interest include robotics and artificial intelligent.

# GitHub Repository as Handbook, Jupyter Notebook as Report: A Format Redesign for the Electrical Engineering II Lab

## 1. Introduction

Laboratory courses play a critical role in electrical engineering education by bridging theoretical concepts and hands-on experience [1]. In our institution, the Electrical Engineering II (EE II) laboratory is a core course taken by both Electrical Engineering and Computer Engineering students in the Fall semester of the second year of the curriculum. The laboratory reinforces Laplace-based modeling, transfer functions, frequency domain, and system response analysis through hands-on experimentation.

Based on prior instructional experience, several limitations of the traditional laboratory format were identified:

- **Limited instructional scaffolding in traditional handbooks:** Word- or PDF-based laboratory handbooks typically present experiments as long, linear documents with limited structural hierarchy. For multi-hour laboratories, this format makes it difficult for students to decompose complex experiments into manageable subtasks.
- **Limited support for embedded code in traditional handbooks:** Word- or PDF-based formats provide limited support for embedding executable code. In our laboratory that relies on Python and Arduino, code formatting issues frequently arise when students copy example code from handbooks. These issues include broken indentation, unintended line wrapping, loss of syntax highlighting, and difficulty in code reuse.
- **Limited exposure to Python-based scientific computing tools:** Although all students have learned Python in their first year, this experience focuses on general programming concepts rather than EE-specific scientific computations. For simple examples, students still rely on calculators to compute transfer-function poles or system gains in decibels (dB). This reliance delays the skill development for upper-level courses in signal processing and control systems.
- **Limited exposure to experimental data analysis workflows:** Data analysis skills have become an increasingly important emphasis in undergraduate education [2, 3]. However, across the overall program curriculum, students often lack experience working with real, structured experimental data [4], including data preprocessing, visualization, and analysis within an engineering context.

To address these limitations, we redesigned the EE II laboratory using the following format:

- **GitHub-repository–based laboratory handbook:**
  - Host all laboratory materials in a GitHub repository written in Markdown, replacing static Word- or PDF-based handbooks;
  - Organize laboratories into weekly folders with modular task files and clearly listed report items;
  - Implement a checkpoint-based workflow to structure multi-hour laboratory sessions;
  - Embed syntax-highlighted Python and Arduino code, mathematical expressions, inter-

nal links, highlighted callouts and experimental videos directly within the handbook.

- **Jupyter Notebook–based laboratory reports:**
  - Use Jupyter Notebook as the platform for student reports, integrating documentation, scientific computation, and data analysis;
  - Document laboratory work using Markdown format that align with the GitHub-based handbook format;
  - Apply Python scientific computing libraries, including NumPy and SciPy;
  - Incorporate experimental data analysis workflows using Pandas and Matplotlib.

This paper describes in detail the redesign and implementation of the new laboratory format. Section 2 provides an overview of the Electrical Engineering II laboratory. Section 3 describes the design of the GitHub-repository–based laboratory handbook. Section 4 presents the Jupyter Notebook–based laboratory reports. Finally, Section 5 discusses assessment of the redesigned laboratory format.

The lab handbook is available online at: `https://github.com/BotYue/ELE-292-Lab-SU`.

## 2. Laboratory overview

At our institution, the Electrical Engineering II (EE II) lecture course is a corequisite for the EE II laboratory. Both the lecture and laboratory courses are scheduled in the Fall semester of the second year of the curriculum.

The lab is currently offered in multiple sections, with each section capped at 24 students. Each laboratory section is scheduled for a three-hour weekly session. The laboratory is staffed by one faculty instructor and one teaching assistant, both of whom are present in person during each session.

The development of this laboratory course followed changes made to the main lecture course, which had already been restructured using Jupyter Notebooks and published as a textbook [5].

The core experimental equipment (see Figure 1) used in the lab includes the Digilent Analog Discovery 2 and the Adafruit ItsyBitsy Arduino board, along with additional instructor-designed hardware modules. Students are required to bring their own laptops to interface with the Analog Discovery and Arduino board in order to complete lab tasks.

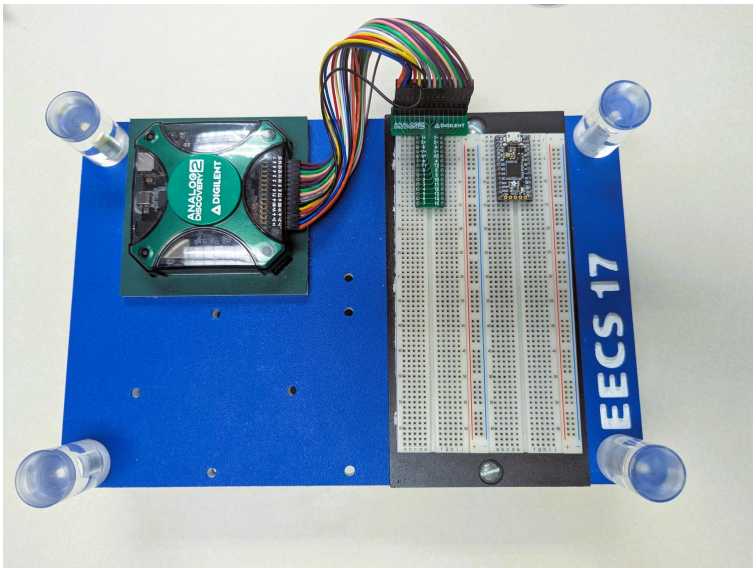


Figure 1: A workbench setup used in the EE II lab. The Digilent Analog Discovery 2 and the Adafruit ItsyBitsy Arduino board are two core components.

An overview of the weekly laboratory topics and activities from the most recent semester is summarized in Table 1.

Table 1: Weekly laboratory topics

| Lab # | Lab title | Short description |
|---|---|---|
| Lab 0 | Software Installation | Installation of the Digilent WaveForms, Arduino IDE |
| Lab 1 | Basic Lab Skills | Report writing in Jupyter Notebook; Analog Discovery – Wavegen and Oscilloscope |
| Lab 2 | Impulse Response | First-order circuit impulse response. |
| Lab 3 | Step Response | First-order circuit step response. |
| Lab 4 | Arduino I | ItsyBitsy Arduino PWM and analog I/O. |
| Lab 5 | Arduino II | ItsyBitsy Arduino analog I/O. |
| Lab 6 | Thermal System | First-order thermal system step response; Analog Discovery – Logger. |
| Lab 7 | Sine and Bode | Sinusoidal response and Bode plot; Analog Discovery – Network analyzer. |
| Lab 8 | Filter Design | Op-amp active filter; Analog Discovery – FFT. |
| Lab 9 | Pendulum I | Second-order physical system impulse response; Arduino serial output |
| Lab 10 | Pendulum II | Second-order physical system state-space model; Arduino serial output |

## 3. GitHub-repository-based laboratory handbook design

### 3.1. Repository structure and task organization

The handbook is organized as a GitHub repository, with each week's laboratory materials contained in a separate folder. Inside each folder, the laboratory is decomposed into two to four Markdown files that describe the required lab tasks, along with an additional `README` file that specifies report requirements and grading rubrics. Within GitHub's web view, this structure is clearly visible through the **left-side file-tree panel**.

The Markdown files within a given folder collectively define the activities to be completed during a three-hour laboratory session. Each Markdown file is further structured using hierarchical headers to break tasks into smaller units such as "Task 2.1", "Task 2.2" shown in Figure 2. Headers such as "Report Item 2-a" explicitly indicate the required elements that students must include in their lab reports. GitHub's **right-side outline panel** of the web view allows students to quickly navigate among small tasks and report items within each file.

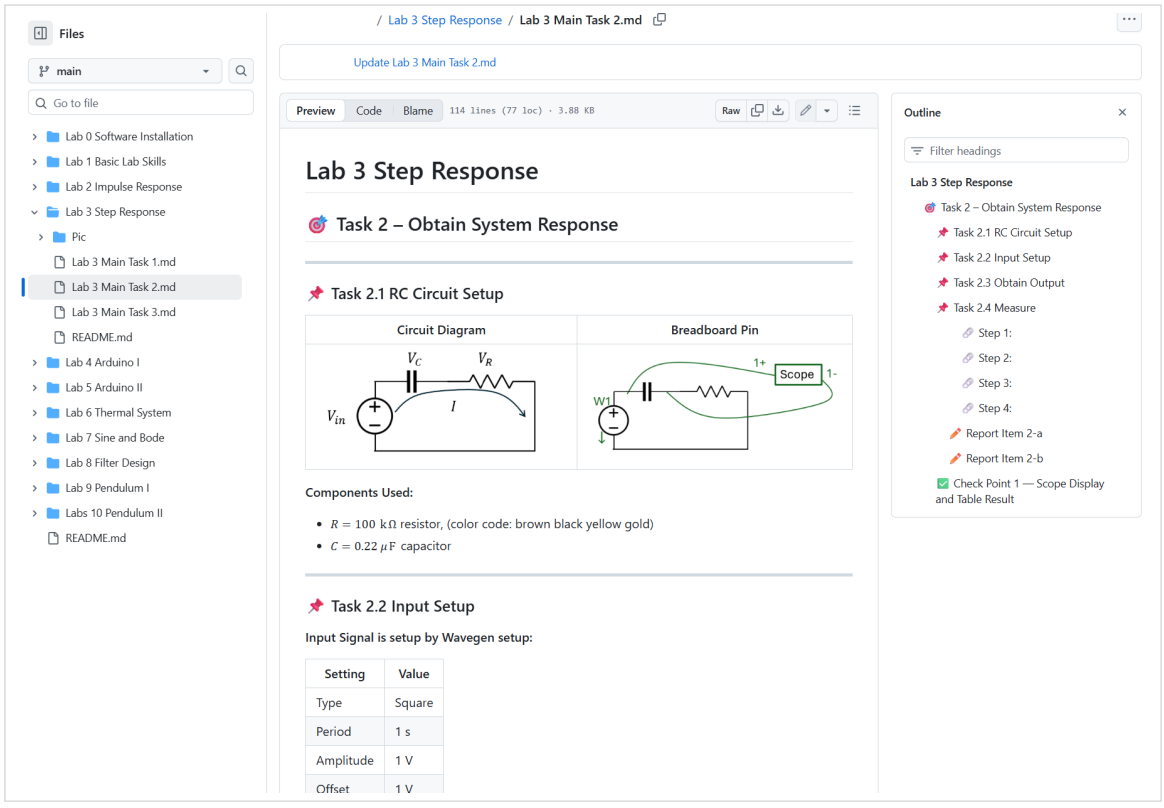


Figure 2: Example GitHub-hosted lab handbook. The left-side file-tree panel shows the repository structure for a 10-week lab sequence. Within the selected "Lab 3 Step Response" folder, 3 Markdown task files define the required laboratory activities, along with a `README` file. The right-side outline panel displays the smaller units for the selected "Lab 3 Main Task 2.md".

### 3.2. Checkpoint-based laboratory implementation

The laboratory is implemented using a checkpoint-based workflow, typically consisting of **two required checkpoints** during a three-hour laboratory session. These checkpoints are designed to mark student progress and provide clear intermediate milestones during in-lab activities.

Checkpoints are embedded directly within the Markdown task files, using clearly labeled format (e.g., "Checkpoint 1 – Expected Result") that explicitly define the required outcomes. During the laboratory session, the instructor or teaching assistant verifies completion of each checkpoint and ticks progress on a check-sheet.

Completion of the second checkpoint does not indicate completion of the entire laboratory. Instead, the second checkpoint typically marks the successful completion of the hardware-related portion of the experiment. Additional tasks such as theoretical calculation and result analysis remain to be completed in the students' individual lab reports.

This two-checkpoint structure provides timely formative feedback and reduces the risk of students needing to redo substantial portions of the laboratory work during the session.

### 3.3. GitHub-flavored Markdown instructional features

The Markdown format on GitHub [6] offers multiple instructional advantages over static documents such as Word or PDF.

**Copyable syntax-highlighted code blocks.** Markdown supports embedding syntax-highlighted code blocks. In this lab, Python and Arduino are the primary programming languages, with limited use of Java when setting the Digilent WaveForms Wavegen. Syntax-highlighted code blocks help students understand code and allow them to easily copy and modify the provided example code for their own work. Language-specific fencing (`python` for Python, `c` for Arduino) in Markdown enables such syntax highlighting .

**Mathematical expressions.** Markdown supports LaTeX-based mathematical expressions, allowing analytical equations to be embedded within the laboratory handbook. Students can switch to the *Code* view (next to the *Preview*) on GitHub to view the raw Markdown and learn how to type mathematical expressions.

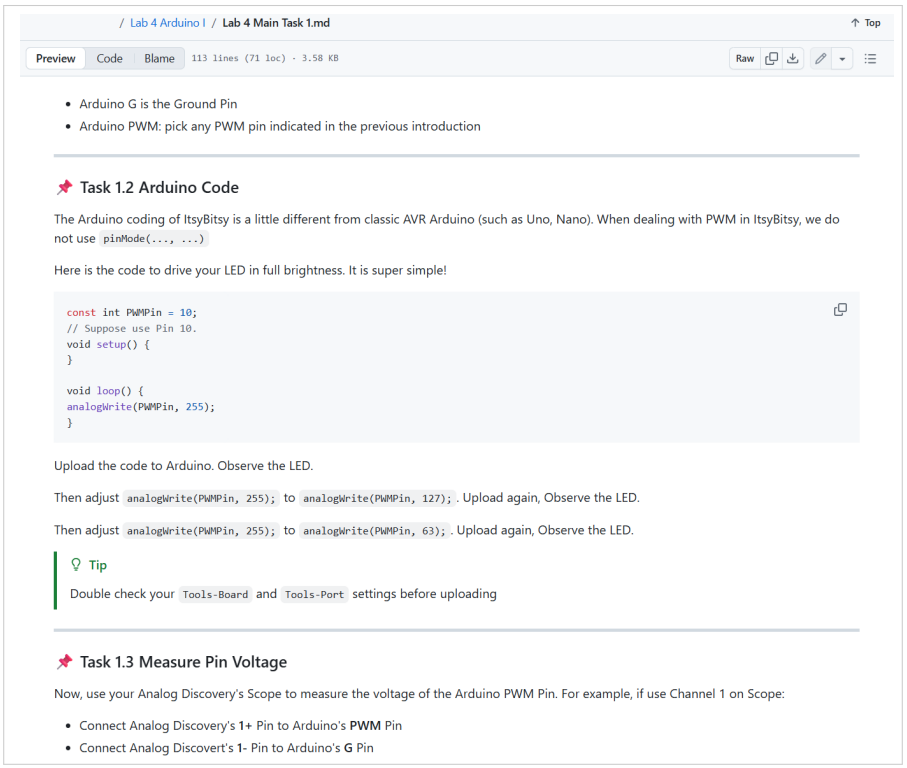


Figure 3: Example Markdown shows syntax-highlighted Arduino code and GitHub's blockquote callout (the green "Tip").

**Internal links for revisiting prior content.** Markdown supports internal linking between files and sections within the GitHub repository. For example, when students conduct time-constant measurements in the second laboratory, an internal link directs them back to the first laboratory file where the concept is initially introduced. This allows students to quickly revisit prior content, supporting just-in-time review and reinforcing conceptual continuity across laboratory sessions.

**Built-in blockquote callouts for highlighting.** GitHub-flavored Markdown supports visually emphasized blockquote callouts that are used to highlight important information in laboratory tasks. There are five GitHub-supported callout types: "Note" (blue), "Tip" (green), "Important" (purple), "Warning" (brown), and "Caution" (red).

**Embedded short videos.** GitHub-flavored Markdown supports embedding short video clips (typically less than 10 MB) directly within the laboratory handbook. Instructors use these embedded videos to demonstrate key experimental procedures and simple hardware interactions that may be difficult to convey through static images or text alone. Videos are embedded by drag-and-drop upload into Markdown files. This feature was introduced in a GitHub platform update in 2021 [7].

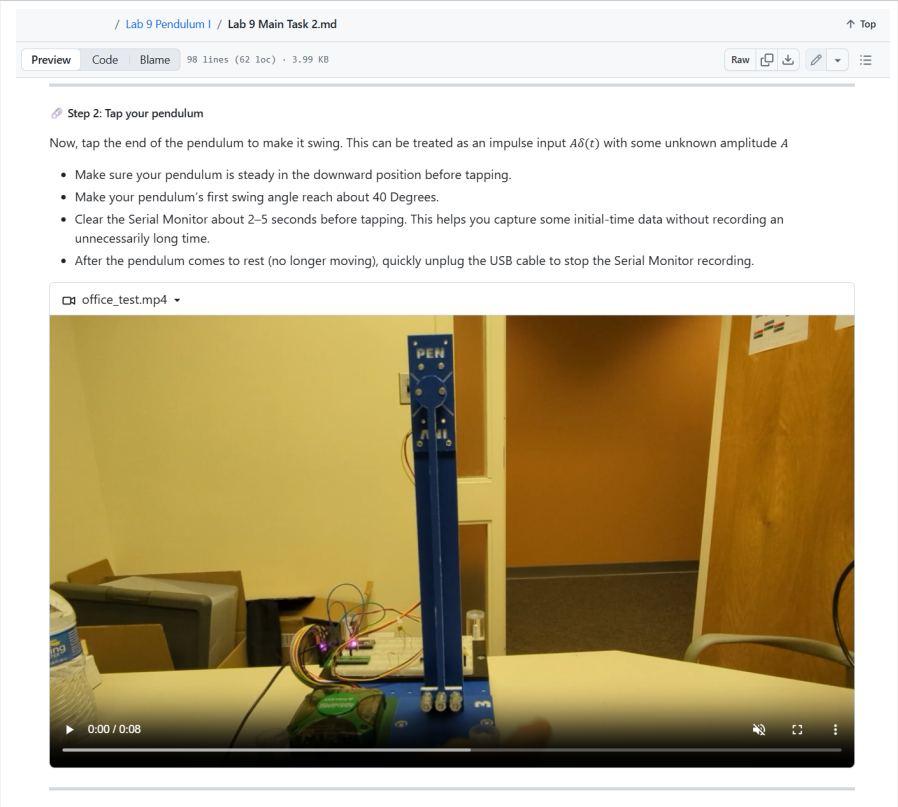


Figure 4: A 8-second experimental video embedded in the GitHub-hosted handbook.

## 4. Jupyter Notebook–based laboratory reports

### 4.1. Jupyter Notebook platform

Jupyter Notebook is a widely used interactive programming platform that supports executable code, documentation, and visualization. It has been adopted across various engineering education disciplines, including electrical engineering [8, 9].

At our institution, Jupyter Notebook is used both in this laboratory and in the corresponding Electrical Engineering II lecture course. For this laboratory, students use the Jupyter Notebook to complete their laboratory reports. The Jupyter Notebook platform enables students to import, analyze, and visualize saved experimental data directly within their reports, as well as seamlessly integrate lab result logging and scientific computation all together.

Students are advised to use Jupyter Lab [10] launched through the Anaconda distribution [11]. Jupyter Lab provides a better development environment than the classic Jupyter Notebook web interface. In particular, Jupyter Lab has a file browser that simplifies folder navigation and makes it easier to import experimental data files such as .csv.

Occasionally, students may encounter difficulties installing Anaconda or may be temporarily unable

to use their personal laptops. To ensure accessibility, students will be guided to use Google Colab as an alternative platform. Google Colab provides the same core functionalities as Jupyter Lab and runs entirely within a web browser. Such setting allows students to complete lab reports even with university computers.

An excerpt from an example Jupyter Notebook laboratory report is shown in Figure 5, presenting the integration of documentation, scientific computation, as well as data processing and visualization. Details are introduced in the next three subsections.

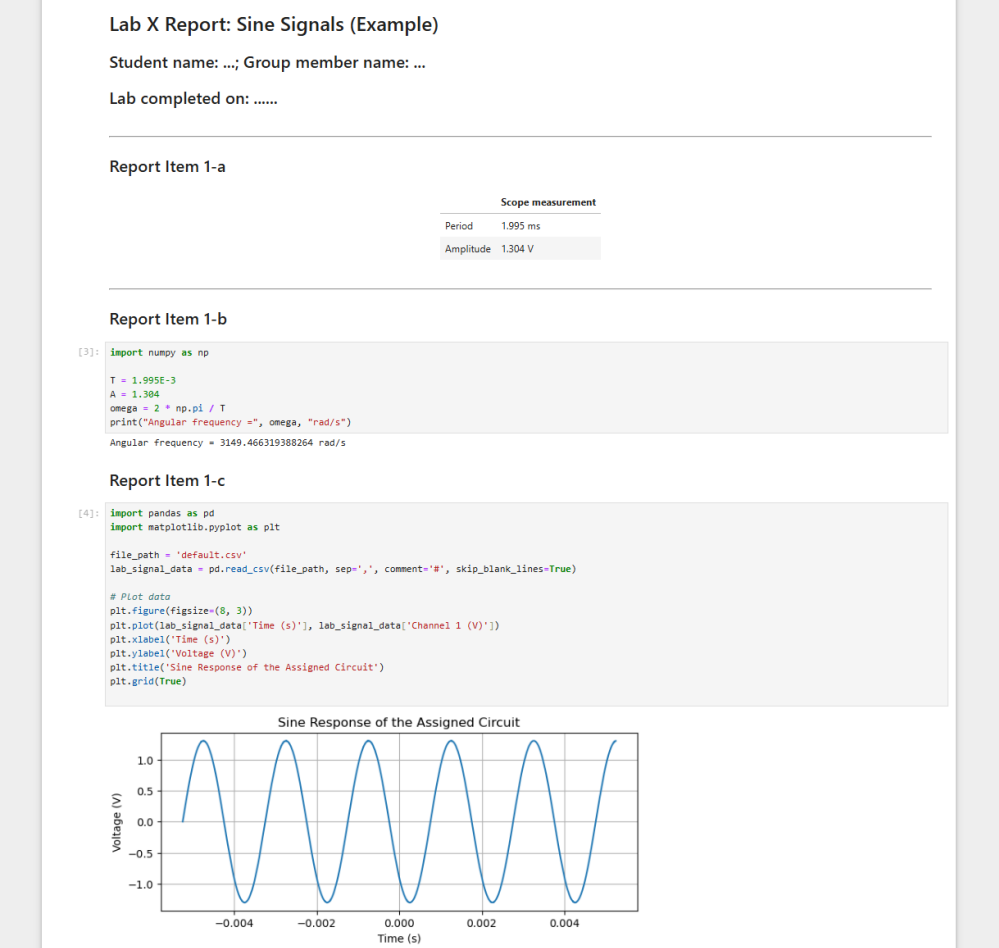


Figure 5: Excerpt from an example Jupyter Notebook laboratory report. Scope measurements are recorded in a Markdown table in Report Item 1-a. Python computations are performed in Report Item 1-b. Data import and visualization are in Report Item 1-c.

## 4.2. Documentation using Markdown

In each Jupyter Notebook, students use Markdown cells to document experimental details, including experimental setup, measured results, analytical procedures, and written conclusions. The use of Markdown within the Jupyter Notebook closely aligns with the Markdown-based laboratory handbook introduced earlier in the paper, resulting in a reduced learning curve of formatting for

students.

For example, our GitHub handbook often provides pre-formatted Markdown tables for logging experimental results with placeholder entries. Students can change to the *Code* view (next to the *Preview*) on GitHub, copy the corresponding raw Markdown script, and paste it directly into their Jupyter Notebook. They then complete the table by entering their measured data. In such way, students are not required to learn Markdown table syntax from the beginning, reducing formatting overhead in documentation.

In addition to text and tables, Jupyter Notebook also allows students to insert images. Common examples in this lab include screenshots of oscilloscope signals captured from the Digilent WaveForms software and photographs of hardware setups.

### 4.3. Scientific computation

Students apply Python-based scientific computing libraries within the Jupyter Notebook environment to perform laboratory analysis. This approach provides students with a better understanding of how computational tools are applied in electrical engineering practice than a standalone programming course [12].

Please note that the corresponding lecture course also introduces these Python functions, typically a couple of weeks prior to this laboratory. This laboratory provides students with additional practice and reinforcement.

NumPy [13] is used for general numerical computation, such as:

- `np.exp` commonly used in first-order system analysis;
- `np.sin`, `np.cos`, `np.pi` commonly used in sinusoidal signal;
- `np.deg2rad`, `np.rad2deg`, `np.abs`, `np.angle` commonly used in phasor analysis;
- `np.sqrt` commonly used in second-order system analysis;
- `np.log10` commonly used in computing gain in decibels (dB);
- and more.

SciPy [14] is used for advanced analysis tasks. Particularly, students use the following functions from the `scipy.signal` module:

- `signal.lti` / `signal.TransferFunction` for defining linear time-invariant system models;
- `signal.impulse` for simulating impulse responses;
- `signal.step` for simulating step responses;
- `signal.lsim` for simulating customized responses;
- `signal.bode` for obtaining Bode;
- and more.

### 4.4. Data processing, visualization, and analysis

Data science concepts are integrated into the laboratory workflow through the use of Pandas [15] and Matplotlib [16] in the Jupyter Notebook platform.

Experimental data in this lab primarily consist of time-series signals obtained from two sources: oscilloscope measurements exported from the Digilent WaveForms software and analog output data acquired via Arduino Serial Monitor. These data are inherently tabular, making them well suited for processing using Pandas. Students are instructed to use Pandas to import the data, perform basic preprocessing, and prepare signals for subsequent visualization.

Once the data are prepared, Matplotlib is used to visualize measured signals in the Jupyter Notebook. In many laboratories, students are asked overlay the analytically predicted signal and measured signal in the same figure to enable direct comparison.

In addition, we also introduce the Matplotlib's interactive measurement tool. This can be realized using `matplotlib.pyplot.ginput` and tkinter backend GUI in Jupyter Lab, or using `plotly` in Google Colab. With this tool, students can use interactive cursors to inspect simple signal features, such as peak values, directly from plotted figure.

Together, these activities promote structured data processing, effective visualization, and reproducible analysis [17] while remaining tightly coupled to the electrical engineering concepts.

## 5. Assessment

This assessment study was reviewed by the Syracuse University IRB and determined exempt (IRB # 26-098).

### 5.1. Student survey

During the Fall 2025 semester, this lab course was offered in 3 class sections, with a total enrollment of 64 students. An anonymous end-of-semester survey was conducted to gather students' perceptions on our new design. The survey was administered using university's Watermark Course Evaluations & Surveys system. A total of 41 students completed the survey, as a response rate of 64.1%.

Overall, the survey results indicate a strong positive student perception of the redesigned laboratory format. Students reported high confidence in using scientific Python tools in an electrical engineering context (Q1, mean = 4.68), suggesting that the integrated GitHub–Jupyter workflow effectively supported skill development. A large majority of students also rated the GitHub-based laboratory handbook as better than traditional paper- or PDF-based manuals (Q2, mean = 4.44). In addition, students strongly supported continuing the checkpoint-based laboratory workflow in future semesters (Q3, mean = 4.71), reflecting the perceived value of structured progress checkpoints. These survey results provide evidence that the redesigned laboratory format improved student laboratory experience.

Table 2: Student responses to survey questions. Responses were collected from 41 students at the end the Fall 2025 semester. Responses use a 5-point scale with numerical weights from 1 to 5. Percentages of responses are reported. Mean values, medians, and standard deviations (STD) are calculated based on the assigned 5-point weights.

| **Q1. After this lab, I feel confident using scientific-Python tools (such as NumPy, Pandas, Matplotlib, SciPy) in the field of Electrical Engineering.** | | | | |
|---|---|---|---|---|
| Disagree (1) | Somewhat disagree (2) | Neutral (3) | Somewhat agree (4) | Agree (5) |
| 0% | 0% | 0% | 31.7% | 66.3% |
| *Statistics: Mean = 4.68, Median = 5, STD = 0.47* | | | | |
| **Q2. Compared with the traditional paper- or PDF-based lab manual, my overall learning experience with the GitHub-based lab manual is:** | | | | |
| Worse (1) | Slightly worse (2) | About the same (3) | Slightly better (4) | Better (5) |
| 0% | 2.4% | 12.2% | 24.4% | 61.0% |
| *Statistics: Mean = 4.44, Median = 5, STD = 0.80* | | | | |
| **Q3. I recommend continuing the checkpoint-based laboratory workflow in future semesters.** | | | | |
| Disagree (1) | Somewhat disagree (2) | Neutral (3) | Somewhat agree (4) | Agree (5) |
| 0% | 0% | 0% | 29.3% | 70.7% |
| *Statistics: Mean = 4.71, Median = 5, STD = 0.46* | | | | |

### 5.2. Handbook repository access metric

Quantitative data were collected from the handbook GitHub repository to characterize student access to the laboratory handbook. Access metric was obtained from GitHub's Insights analytics, which provide rolling 14-day summaries of repository traffic, including unique visitors. These analytics are viewable only to the repository owner.

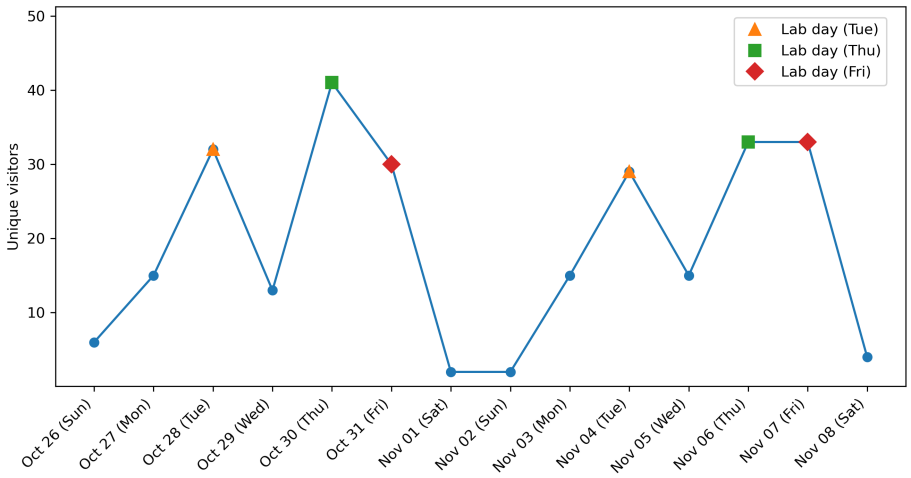


Figure 6: Daily unique visitors of the handbook GitHub repository over a representative two-week period. Lab days: Tuesday (18 students), Thursday (23 students), and Friday (23 students).

To examine typical usage patterns during regular instruction, we selected time windows between Oct. 26 and Nov. 8, 2025 for analysis. The laboratory was offered in three sections meeting on Tuesday, Thursday, and Friday, with enrollments of 18, 23, and 23 students, respectively. As shown in Figure 6, daily unique visitor counts exhibit clear increases on scheduled laboratory days. Such a pattern suggests that a substantial number of students accessed the GitHub-hosted laboratory materials during scheduled lab sessions.

While repository engagement metrics do not directly measure learning outcomes, these data provide complementary evidence that the GitHub-based laboratory handbook supported sustained student engagement with laboratory materials throughout the instructional period.

## 6. Conclusion and discussion

This paper presented a redesign of the Electrical Engineering II laboratory format that replaces traditional Word- or PDF-based laboratory manuals with a GitHub-repository–based handbook and adopts Jupyter Notebook as the platform for laboratory reporting. The redesign was motivated by several observed limitations of the traditional format, including limited instructional scaffolding, poor integration of executable code in handbooks, insufficient exposure to scientific computing tools, and insufficient exposure to experimental data analysis workflows.

The redesigned laboratory format transforms the handbook from a static document into a modular instructional resource. The handbook, written in GitHub-flavored Markdown, features hierarchical task organization, checkpoint-based workflows, embedded syntax-highlighted code, internal links, highlighted callouts, and short instructional videos that better support multi-hour laboratory sessions. The Jupyter Notebook reporting integrates documentation, scientific computation, and experimental data analysis. Student survey responses and repository access metric provide positive supporting evidence for the redesigned format.

One suggestion for implementing this format is to maintain an offline backup of the handbook materials to ensure continued access in the rare event of a GitHub service outage. In our experience, during one Tuesday laboratory section on November 18, 2025, GitHub experienced a temporary platform-wide service outage [18]. As a contingency, the laboratory materials were converted from Markdown to PDF format and distributed to students through the university's learning management system.

## References


[1] I. Mikhelson, "Introduction to electrical engineering: Empowering and motivating students through laboratory-focused teaching," in *ASEE Annual Conference & Exposition*, 2024.

[2] National Academies of Sciences, Engineering, and Medicine, *Data science for undergraduates: Opportunities and options*. National Academies Press, 2018.

[3] M. Y. Naseri, C. Snyder, K. X. Pérez-Rivera, S. Bhandari, H. A. Workneh, N. Aryal, G. Biswas, E. C. Henrick, E. R. Hotchkiss, M. K. Jha *et al.*, "Integrating data science into undergraduate science and engineering courses: Lessons learned by instructors in a multiuniversity research–practice partnership," *IEEE Transactions on Education*, 2024.

[4] K. Suthar, T. Mitchell, A. C. Hartwig, J. Wang, S. Mao, L. Parson, P. Zeng, B. Liu, and P. He, "Real data and application-based interactive modules for data science education in engineering," in *ASEE Annual Conference*, 2021.

[5] D. Marcy and J. Graham, *Fundamentals of Linear Systems*. Kendall Hunt Publishing, 2025. [Online]. Available: https://he.kendallhunt.com/product/fundamentals-linear-systems-0

[6] GitHub Docs, "Basic writing and formatting syntax," https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax.

[7] GitHub, "Now you can upload video to markdown, plus more updates to GitHub flavoured markdown," https://youtu.be/G3Cytlicv8Y, 2021.

[8] A. Zúñiga-López and C. Avilés-Cruz, "Digital signal processing course on jupyter–python notebook for electronics undergraduates," *Computer Applications in Engineering Education*, vol. 28, no. 5, pp. 1045–1057, 2020.

[9] C. K. Frederickson, "Introducing automation for data acquisition and analysis in a sophomore-level electronics course," in *ASEE Annual Conference & Exposition*, 2023.

[10] Project Jupyter, "Jupyterlab is ready for users," https://blog.jupyter.org/jupyterlab-is-ready-for-users-5a6f039b8906, 2018.

[11] Anaconda, Inc., "Anaconda software distribution," Computer software, Nov. 2016, https://anaconda.com.

[12] C. L. Vizcarra, R. F. Trainor, A. Ringer McDonald, C. T. Richardson, D. Potoyan, J. A. Nash, B. Lundgren, T. Luchko, G. M. Hocky, J. J. Foley IV *et al.*, "An interdisciplinary effort to integrate coding into science courses," *Nature Computational Science*, vol. 4, no. 11, pp. 803–804, 2024.

[13] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith *et al.*, "Array programming with numpy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[14] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, "Scipy 1.0: fundamental algorithms for scientific computing in python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[15] The pandas development team, "pandas-dev/pandas: Pandas," 2025. [Online]. Available: https://zenodo.org/records/17229934

[16] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[17] C. Griego, C. Zhang, W. Hu, Z. Ma, and A. Ouyang, "Introducing students to research and reproducibility with open science tools," in *ASEE Annual Conference & Exposition*, 2024.

[18] GitHub, Inc., "Incident report: Intermittent issues loading images," https://www.githubstatus.com/incidents/5q7nmlxz30sk, 2025, accessed: Jan. 2026.