

## **Empowering Engineering Education with the Raspberry Pi 5**

**Himnish Jain, University of Illinois Urbana-Champaign, Siebel School of Computing and Data Science | Department of Statistics**

Himnish Jain is a researcher with a background in computer science, statistics, and economics. His work spans machine learning, computer vision, and embedded systems, with a focus on real-time applications and educational technologies. He has contributed to AI-driven projects, including satellite imagery analysis, human-in-the-loop systems, and assistive technologies. In this work, he explores the potential of the Raspberry Pi 5 in engineering education, emphasizing its role in hands-on learning and computational problem-solving. Himnish is set to graduate with dual bachelor's degrees in Computer Science, and Statistics, from the University of Illinois Urbana-Champaign in May 2025.

**Eisha Peyyeti, University of Illinois-Champaign, Siebel School of Computing and Data Science**

Eisha is a researcher with a background in computer science, embedded systems, and hardware design. Her work spans high-power rocketry, robotics, satellite data image decoding, drones, and printed circuit board design, giving her a strong foundation in both software and hardware development. In this work, she offers insights into integrating the Raspberry Pi into the classroom, exploring best practices for enhancing learning through embedded systems. Eisha is set to graduate with a Bachelor of Science in Computer Science from the University of Illinois Urbana-Champaign in May 2025.

**Taniya Agrawal, University of Illinois-Champaign, Siebel School of Computing and Data Science**

**Professor Lawrence Angrave, University of Illinois-Champaign, Siebel School of Computing and Data Science – Teaching Professor, Gies RC Evans Innovation Fellow, CITL Fellow**

## **Introduction**

Hands-on learning is a crucial element of a computer science education, as it allows students to engage with real, complex systems in a tangible, interactive way. However, traditional embedded systems can be challenging to new learners, due to their steep learning curve and associated high costs. The Raspberry Pi 5, however, offers a more accessible entry point into system-level computing.

The Raspberry Pi is an affordable, easy-to-use embedded system, with a strong support community behind it; this makes it an ideal tool for students to explore topics such as operating systems, networking, filesystems, and hardware integration, without the barriers that often come with specialized embedded platforms.

By reducing the overhead required to get started, the Raspberry Pi allows students to focus on learning core, embedded computing principles, through experimentation and real-world applications.

A prior ASEE paper in 2020, “Infusing Raspberry Pi in the Computer Science Curriculum for Enhanced Learning,” discussed a larger deployment of the Pi across courses and how projects at each level differ. While the paper focuses more on the broad integration of the Pi into the curriculum, our work presents the strategies necessary to implement such deployments at scale, particularly in courses with large student enrollments.

This paper presents a multi-perspective exploration of the Raspberry Pi’s educational potential in system-level learning. From a college professor’s standpoint, Professor Lawrence Angrave highlights its role in teaching fundamental systems concepts. Eisha Peyyeti, as a teaching assistant running the honors section of UIUC’s Systems Programming course, shares observations and insights on strategies for fostering problem-solving and resilience in students as they work with hands-on, embedded devices. Himnish Jain, as the Lead Teaching Assistant for an introductory engineering course at UIUC, offers insight into how exposure to hands-on projects can bridge foundational learning with advanced, industry-relevant skills. Taniya Agarwal, a first-year undergraduate, offers a fresh perspective on how the Raspberry Pi 5 (Pi-5) supports early engagement in research and how she benefits from a community-driven support ecosystem, including resources from the Raspberry Pi Foundation and online forums.

Beyond these perspectives, this paper also details the development of a custom Pi-5-based kit, designed to make the Raspberry Pi more accessible to students. Additionally, we explore practical strategies for integrating the Raspberry Pi into classrooms, from small honors sections to larger university courses, emphasizing its role in promoting engagement, accessibility, and real-world skill development.

By examining both student and educator experiences, as well as logistical considerations, this work advocates for the Raspberry Pi’s wider adoption as a transformative educational tool in systems programming and engineering education. We highlight specific challenges and solutions encountered during implementation, offering insights into curriculum design, hardware setup, and student learning outcomes. Ultimately, this paper aims to provide a roadmap for institutions

looking to incorporate Raspberry Pi-based learning into their courses effectively.

### **Overview of the Raspberry Pi 5 Kit**

To make the Raspberry Pi as accessible as possible to students, we put together a comprehensive kit that includes essential hardware components. The cost of the initial version of the kit was approximately \$80. It included a 4GB Raspberry Pi 5 (\$60), a 64GB microSD card (\$12), a retro USB gamepad (\$2), and a mini USB microphone (\$6), all organized in a pencil box (\$2). To keep costs down, an output screen and case were not included. Items were purchased from Newark, Adafruit, B&H Photo Video, and Amazon. Based on student difficulties in the first semester, the revised version of the kit for the Spring 2025 semester will also include a micro-HDMI to HDMI converter and USB-C data cable. This approach is both cost-effective and practical, as it provides students with a powerful yet affordable embedded system for hands-on learning. By standardizing the hardware across all students, the kits also eliminate compatibility issues and ensure that everyone has access to the same tools. Though the supply of Raspberry Pi devices was limited during the COVID pandemic, today (2/7/2025) Newark (Newark.com) reports over 2500 4GB Pi-5 devices in stock and available.

At the core of the kit, the Raspberry Pi and the microSD card provide the foundation for all projects. The SD card allows students to easily flash and install different operating systems, enabling them to set up a functional computing environment for programming, networking, and embedded systems work. This basic setup ensures that students can start with fundamental tasks such as configuring a Linux system, running Python scripts, and connecting remotely to the Pi via SSH.

Beyond this foundation, the included microphone and gamepad serve as additional tools that expand the scope of standard Raspberry Pi projects. For example, the gamepad can be used to build a custom retro gaming console, enabling students to learn about emulation and system configuration. The microphone also opens up opportunities for projects involving voice recognition, audio processing, or even simple machine learning models for speech-based interaction.

### **Study Design & Methodology**

To best evaluate the impact of the Raspberry Pi on engineering education, we structured our study into two components: a qualitative section and a quantitative section.

The qualitative section, titled "Participants & Perspectives," explores firsthand experiences from both new learners and instructors, highlighting their observations on accessibility, engagement, and the learning curve affiliated with the Pi. The quantitative section, titled "Measured Impact," analyzes survey data from students who interacted with the Raspberry Pi 5, focusing on their comfort levels with embedded systems and hardware before and after using the device, as well as tools that they found to be helpful in their work with the Pi.

By combining these two approaches, we aim to provide a comprehensive understanding of how the Raspberry Pi contributes to both skill development and confidence in engineering education.

## **Participants & Perspectives**

### **1. Supporting Students**

#### **a. Systems Programming Professor**

The course instructor's perspective focuses on the long term, building institutional know-how over multiple semesters and taking a pedagogical, low-risk, exploratory but playful approach to adding Pi's to the computer science undergraduate degree program. The System Programming class at ANONYMIZED is the last required programming class for CS Majors before they take the upper elective elective courses. At this point in their undergraduate program there is pedagogical value in exposing undergraduate students to computational environments that are outside of the comfort zone of their own laptop's resources. Since 2012, students in this class have been assigned their own 4-core Linux virtual machine (that runs on university hardware) and provided with SSH and sudo privileges. However, with the cost of complete Linux systems (e.g. Pi-5) falling dramatically, the course instructor is interested in exploring the use of Raspberry Pi as an alternative and interesting platform for system programming assignments. With an enrollment of up to 400 students per semester, a tight schedule of programming assignments, and later courses depending on the multiple learning outcomes of this course, it would be difficult and chaotic to simply switch the course from one platform to another, and the course quality would suffer. In particular, the course staff would have no prior experience in supporting students on the Pi platform.

Instead, a more sensible approach is to add the Pi platform to the honors section of the System Programming course (with a typical enrollment of up to 15 students per semester) and ask these students to create interesting projects using this new platform or to explore use of the Pi platform for the main course's assignments. This is a low-risk approach; there are no expectations by later courses that these projects will succeed. There are many unknowns to address that can now be addressed using this approach such as: Will development tools (e.g. clang, valgrind and gdb) and build scripts work without modification? Will it be necessary to delay other course content by one lab, to allow students to learn how to set up and use the Pi? What interesting uses of the Pi platform are possible given limited student time each time?

The main course has a large supporting staff—up to 15 undergraduate course assistants and 6 graduate TAs. Each semester, the best students in the course may return as future undergraduate course assistants. Thus, over time it will be possible to build sufficient course staff knowledge to fully support student development on the Pi platform at the scale of supporting hundreds of students. It will also be possible to understand how to create new programming assignments that are engaging, satisfy system programming learning objectives, are suited to our Pi kit and can be completed within 1 - 2 weeks.

#### **b. Systems Programming Course Assistant**

Eisha is a senior, undergraduate course assistant who ran an honors section of the University of Illinois Urbana-Champaign's junior-level Systems Programming course. The course objectives focused on deepening students' understanding of system-level programming, hardware-software integration, and networking, while fostering key real-world skills such as problem-solving, debugging, project management, and collaborative work.

Her first year leading the honors section was also the first time the section focused on the Raspberry Pi as its central platform. She structured the semester around the theme of "embedded systems," guiding students to develop challenging projects that integrated an embedded device while applying their newfound systems programming skills.

To provide structure, she outlined clear expectations, including key deliverables. These included a project proposal, progress checkpoints with assigned mentors, and regular opportunities for feedback. While encouraging students to push their technical limits, she also emphasized that the honors project should be an exciting learning experience, rather than a source of stress.

Her goal was for students to create projects they could be proud to showcase on their resumes and also earn class credit along the way. To do this, Eisha reassured students that success in the course was based on meaningful progress and effort, ensuring that those who engaged deeply with their projects could earn honors credit for the Systems Programming course.

Eisha structured the honors section of the course to emphasize hands-on learning with the Raspberry Pi 5, aiming to create an environment where students could explore systems concepts through direct experimentation. Her approach focused on fostering problem-solving skills and resilience, encouraging students to engage deeply with embedded systems development. By providing students with the Raspberry Pi kit, as described above, they had the flexibility to explore additional peripherals and expand their projects, without significant overhead. This approach allowed students to experiment with hardware and software integration in a hands-on way, which ultimately reinforced key embedded systems concepts.

One of the primary challenges she faced was guiding students with varying levels of hardware and programming experience. While some students had prior exposure to microcontrollers and Linux-based development, others were entirely new to embedded systems. To accommodate these differences, she provided structured introductory materials, while promoting self-directed learning. These introductory materials consisted of mini-lessons on new systems concepts, a list of project ideas for the Raspberry Pi, documentation for all the components within the Pi toolkit, as well as a resource list of general Raspberry Pi documentation. She additionally emphasized debugging as a fundamental skill rather than a roadblock. This ultimately helped students see the process of troubleshooting issues with the Pi as a natural part of the learning process.

Eisha made sure to maintain a balance between independent problem-solving and providing direct answers through weekly, assigned mentor meetings. These mentors were other course assistants from the Systems Programming course, who were also passionate about embedded systems and the Pi. Mentors were matched with groups based on project and systems concept interest. These meetings provided a structured opportunity for mentors to check in with project groups, answer questions, and offer debugging assistance when needed. Mentors also helped students anticipate challenges by providing insights on upcoming milestones and potential roadblocks. This regular feedback loop ensured that students had the support necessary to stay on track, and also encouraged them to take ownership of their own problem-solving with the Pi.

Reflecting on the first semester of the “embedded honors course”, Eisha found that assignments encouraging exploration, like modifying existing systems rather than following rigid step-by-step instructions, was an effective strategy to help students learn. These activities fostered creativity and deeper engagement with the material.

### **c. Engineering Course Lead Teaching Assistant**

Serving as the Lead Teaching Assistant for a first-year engineering course (Engineering 100), Himnish Jain brings a vantage point that straddles introductory education and industry-relevant practice. In Engineering 100, many new students grapple with transitioning from largely theoretical high school coursework to more tangible, hands-on collegiate-level engineering assignments. Himnish has consistently observed a sense of trepidation among these students when they first encounter hardware components, wiring, and command-line interfaces—especially as they realize that hardware-related mistakes can feel more daunting and less forgiving than purely software-based ones.

The Raspberry Pi 5 mitigates this anxiety by offering a more intuitive starting point. It is compact, relatively low-cost, and backed by a vast online community that provides abundant tutorials, troubleshooting tips, and real-world project examples. Himnish notes that simply knowing such resources exist lowers students’ fear of “messing up,” and helps them view the Pi as a natural extension of their problem-solving toolkit rather than a mysterious black box. In his words, the Pi “removes the hesitation barrier” that commonly inhibits experimentation in the early stages of an engineer’s education.

A typical first assignment might involve configuring the Pi, connecting to it via SSH, and running a basic data-logging script. This single exercise often yields a surge of excitement among students: they tangibly see how their code can gather and store sensor readings or system metrics. Such a relatively straightforward project, says Himnish, “instills early confidence” in hardware manipulation and fosters a sense of accomplishment that fuels motivation for subsequent challenges.

A notable observation from Himnish’s tenure is how quickly students progress from beginner-level tasks to more interdisciplinary and advanced explorations. With the Pi, many choose to move beyond the “Hello World” of LED blinking or temperature monitoring; they soon gravitate toward data-centric applications. Because the Pi supports

a familiar Linux environment with Python, students can implement small-scale machine learning models or sensor fusion algorithms directly on the device. As Himnish puts it, “They see the entire data pipeline—collection, processing, and even basic inference—running on a single board.” This practical end-to-end exposure is rare in early undergraduate courses but deeply valuable, making the leap to more complex capstone projects or research labs less intimidating.

Himnish also notes that the Raspberry Pi provides a “low-stakes yet high-impact” environment for learning resilience. Hardware-related bugs, misconfigurations, and compatibility issues are inevitable, but the Pi’s affordability and online community give students multiple safety nets. This encourages them to debug persistently instead of giving up at the first sign of trouble. By viewing mistakes as part of the experimentation cycle, learners develop perseverance—an essential trait in real-world engineering.

Moreover, Himnish observes that these Pi-based projects naturally expose students to other disciplines. Mechanical engineering students might attach servos or stepper motors to their Pi, while electrical engineering students delve into GPIO pin mapping and advanced signal processing. Data science-inclined students learn about real-time sensor data handling and how to refine computational models for constrained systems. These cross-disciplinary overlaps help cultivate a broader engineering mindset and sharpen students’ appreciation for how hardware, software, and algorithmic concepts intersect in modern technology.

A key outcome of integrating the Pi into Engineering 100 is that students become more prepared—and sometimes more enthusiastic—about tackling upper-division coursework. Himnish has seen students, initially anxious around electronics, pivot toward actively seeking more ambitious embedded systems or data science projects in later semesters. Even those who do not specialize in embedded technologies become more comfortable with the notion of interacting with hardware and diving into technical documentation.

This confidence pays dividends in internship and job interviews as well. While discussing personal projects, students can reference practical achievements—such as building a networked sensor system on the Pi or experimenting with basic ML models for classification tasks. These experiences showcase both their technical abilities and their ability to learn new tools quickly. In Himnish’s words, “The Pi is an incredible launchpad that allows you to fail fast, learn quickly, and ultimately build something meaningful—whether you’re a first-semester freshman or a senior diving into data science and beyond.”

Reflecting on several semesters of use, Himnish believes that the Pi’s greatest strengths are its adaptability to different project scales and its robust support ecosystem. For a course that emphasizes professional development, the Pi neatly embodies the central theme of engineering: learning by doing, embracing complexity, and solving problems creatively with collaborative support. The minimal overhead and cost encourage repeated, iterative experimentation—an approach that fosters resilience and genuine skill-building more effectively than purely theoretical exercises.

In short, from Himnish's experience, the Raspberry Pi 5 in Engineering 100 offers not just an introduction to embedded systems, but an early and lasting spark that propels students toward deeper technical exploration. The accessibility, supportive community, and clear real-world applications make it an ideal on-ramp for first-year students seeking to merge theoretical engineering principles with practical, hands-on problem-solving.

## **2. New Learners**

### **a. First-Year Undergraduate Researcher**

Taniya, a first-year undergraduate student, joined a research group that aims to make STEM education more accessible for the deaf and hard-of-hearing community by optimizing real time transcriptions. Since this was an ongoing project, Taniya had to join a team to help them with their existing work, and she joined Eisha who focused on working with various systems (like the different versions of Raspberry Pi and Google Coral) in order to make the implementation of ScribeAR more cost-effective and accessible. Taniya started working with Raspberry Pi 5, and her goal was to use Gadget Mode so that the Pi could be connected to a computer or monitor like a USB (irrespective of the network the monitor and the Pi are in) and run the live transcription. Her biggest initial challenge was remotely accessing Pi so that she could work on the Pi without using a full PC system. She was successful in resolving that issue through the official Raspberry Pi website/forums and LLMs. Although it was Taniya's first time using a Raspberry Pi, she was able to tackle the challenges she encountered because of the Pi's widespread usage and extensive community driven knowledge-base, allowing her to learn more about the Pi and problem-solve. After remotely accessing the Pi through her laptop, Taniya's next step was to implement Gadget Mode.

As she is working on her implementation, reflecting back, Taniya found this experience extremely enriching. She is not only able to build her technical skills but also work towards contributing to a research project by learning how to work with a new technology.

### **b. Systems Programming Honors Students**

The first semester of the Systems Programming Honors Section with Raspberry Pi yielded impressive student projects and valuable learning experiences. Students worked on a variety of challenging technical projects, including the development of a USB Rubber Ducky, a FUSE-based Course Explorer Filesystem, and a Raspberry Pi Network Firewall. These projects required students to implement and optimize critical system operations like file system calls, API query scripts, metadata caching, and traffic sniffing.

Real-world skills targeted by the course included:

- Embedded Systems Design and Debugging: Students learned to troubleshoot issues related to hardware integration and software performance.

- Networking and Security: Through projects like the network firewall and API integrations, students gained practical knowledge in networking protocols and secure communication.
- Systems Programming: Students worked directly with system calls, file systems, and networking protocols, which allowed them to bridge theoretical knowledge with practical application.

Reflecting on their experiences, students expressed an increased appreciation for the Raspberry Pi's versatility in real-world applications. Many found the process of debugging and troubleshooting to be the most challenging but also the most rewarding part of the course. The student feedback, collected through a Google Form, provided valuable reflections on their experience making a project with the Pi in the Honors Section. One student shared their significant contributions, saying, "I played a key role in both ideation and implementation of this project," where they worked on the integration of real-time data retrieval from the University Course Explorer API. They also noted the importance of optimizing system calls and caching mechanisms. One student reflects how they "would focus on further optimizing the caching mechanism by implementing an LRU cache" to improve performance. Many students reflected on their challenges and the importance of planning, with one remarking, "Do more research before finalizing on the approach to the project." This highlights the importance of students understanding the Raspberry Pi project's scope before diving into implementation. When students were asked what they would do with more time, another student responded, "I would introduce parallel API requests to fetch data for multiple directories simultaneously, improving overall performance and responsiveness." These insights showed how students grew in their understanding of system-level programming, and how it deepened their technical knowledge. A great note was that of several expressing a desire to mentor future cohorts: "I'd love to mentor students working with the Pi in future semesters." The feedback suggests that the course not only helped students develop technical skills but also fostered a deeper engagement with Raspberry Pi projects and systems programming.

### **Measured Impact**

To assess the impact of the Raspberry Pi 5 on student learning, we conducted a survey involving students enrolled across two semesters of the Systems Programming Honors section, who worked directly with making a project from the Pi Kit. The primary goal of the survey was to evaluate changes in students' comfort levels and proficiency with embedded systems and hardware before and after working with the Raspberry Pi 5 as part of their coursework. The survey was designed to capture both quantitative and qualitative data, providing a comprehensive view of how the Raspberry Pi 5 influenced their learning experiences.

Participants were asked to respond to a series of Likert scale questions (ranging from 1 to 5) that measured their comfort and familiarity with various aspects of embedded systems and hardware. The scale ranged from "Not Comfortable At All" (1) to "Very Comfortable" (5), allowing us to quantify shifts in students' confidence and understanding over the course of the semester. These questions focused on specific areas such as working with sensors, programming microcontrollers, interfacing with hardware, troubleshooting, and designing embedded systems projects.

In addition to the Likert scale questions, the survey included a set of open-ended questions aimed at capturing more nuanced insights into students' experiences. These questions were designed to explore their perceptions of mentorship and how guidance from instructors or peers impacted their learning process. The open-ended section also encouraged students to reflect on the resources they found most helpful, whether it was documentation, online forums, or physical tools, and to suggest ways these resources could be improved or expanded.

For data analysis, responses were collected and analyzed to identify trends and patterns in how students' comfort levels evolved throughout the course. The quantitative data from the Likert scale were subjected to statistical analysis to determine if there were any significant changes in student confidence and understanding. The qualitative responses provided additional context and depth to the numerical findings, offering insights into how students approached learning, overcame challenges, and what support structures were most effective in facilitating their success.

To protect student privacy and ensure the integrity of the study, all personal identifying information was anonymized and excluded from the dataset used for analysis and display. This approach allows us to focus on the general trends and insights gleaned from the students' collective experiences with the Raspberry Pi 5 in the context of engineering education.

**Quantifiable Questions (Likert Scale: 1-5)**

- A) How comfortable are you with embedded systems? (1-5)
- B) How comfortable were you with hardware before you got the Pi 5 kit? (1-5)
- C) How comfortable are you now that you have the Pi 5 kit? (1-5)

We have included a small sample of the survey responses to illustrate trends in student comfort levels before and after using the Raspberry Pi 5.

<b>Student</b>	<b>A) Comfort with Embedded Systems (1-5)</b>	<b>B) Comfort Before Pi 5 (1-5)</b>	<b>C) Comfort After Pi 5 (1-5)</b>
<b>1</b>	2	2	3
<b>2</b>	4	2	4
<b>3</b>	1	1	3
<b>4</b>	2	1	2
<b>5</b>	2	1	2
<b>6</b>	2	3	3
<b>7</b>	3	2	3
<b>8</b>	2	2	4

## Open-Ended Questions

- A) How valuable have you found your mentor/having one in general? Why?
- B) What resources have you made use of so far when working with the Pi?
- C) Would you recommend the Raspberry Pi to other students working on similar projects? Why or why not?

To provide insight into students' qualitative experiences, we have also included a subset of responses.

<p><b>A) How valuable have you found your mentor/having one in general? Why?</b></p>	<p>"Very valuable, as he seems really experienced and knowledgeable."          "Fairly valuable, my mentor helped guide us through issues and gave development tips."          "Pretty valuable, gave some advice!"          "Our mentor has been helpful in guiding us on obtaining equipment and approaching our project."          "Nice having someone to rely on because I am not experienced in the field."          "Helped us understand how to get started and gave us direction."          "Very valuable, helped us decide on a more specific direction for our project."          "Extremely valuable. Having someone provide wisdom and advice is crucial for a project like this."</p>
<p><b>B) What resources have you made use of so far when working with the Pi?</b></p>	<p>"Online forums and articles."          "Online forums, Stack Overflow, and generative AI."          "Online tutorials (websites and YouTube) and Stack Overflow."          "Mainly online forums and articles."          "Used AI and online coding forums mostly."          "Raspberry Pi forums and Stack Overflow to understand how to connect to the Pi from our laptops."          "Multiple websites, mostly forums, to learn how to do things."          "Online forums like Reddit and Stack Overflow were helpful. I was looking into emulating games and YouTube guides were helpful for that."</p>
<p><b>C) Would you recommend the Raspberry Pi to other students working on similar projects?</b></p>	<p>"Yes, it's a great tool for beginners interested in embedded systems."          "I would, because it's a great way to get used to Linux, and having a physical device gives more control when things go wrong (like fork bombs)."          "Yes, nice for hardware projects."          "Yes, I would!"          "Yes, it provides a brilliant interface for low-level systems projects."          "Yes, very versatile for a variety of projects. For ours, a VM might be just as effective, but the Pi could be more useful later."          "I would recommend it because the GPIO ports make controlling devices</p>

<b>Why or why not?</b>	easier." "Yes, best introductory SBC on the market."
------------------------	---

While we could not quantifiably evaluate the open-ended questions from students, these perspectives were still considered when presenting our key findings. The qualitative feedback provided valuable insights into students' experiences, particularly in relation to the mentorship and resources they utilized while working with the Raspberry Pi. For instance, many students found their mentors to be extremely valuable, providing guidance, development tips, and project direction. Students also mentioned relying on resources such as online forums, Stack Overflow, YouTube tutorials, and generative AI to troubleshoot and learn new skills. This qualitative feedback helped contextualize the quantitative data, offering a more comprehensive understanding of students' learning experiences and challenges.

With the quantifiable data in hand, we used Python to generate plots that visually represented student experiences. The following steps were carried out to analyze and visualize the students' comfort levels before and after using the Raspberry Pi 5 kit.

First, we loaded the data from the CSV file of survey results, ensuring that the column names were standardized and blank rows were removed. Then, numerical columns were converted to integers for proper analysis. We calculated the improvement in comfort level by subtracting the initial comfort with hardware from the comfort level after using the Raspberry Pi.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

file_path = "pidata.csv"
df = pd.read_csv(file_path)

df.columns = df.columns.str.strip()

df = df.dropna(subset=['How comfortable are you with embedded systems?', 'How
comfortable were you with hardware before you got the Pi 5 kit?', 'How
comfortable are you now that you have the Pi 5 kit?'])

df['How comfortable are you with embedded systems?'] = df['How comfortable are
you with embedded systems?'].astype(int)
df['How comfortable were you with hardware before you got the Pi 5 kit?'] =
df['How comfortable were you with hardware before you got the Pi 5
kit?'].astype(int)
df['How comfortable are you now that you have the Pi 5 kit?'] = df['How
comfortable are you now that you have the Pi 5 kit?'].astype(int)
```

```
df['Comfort Improvement'] = df['How comfortable are you now that you have the Pi 5 kit?'] - df['How comfortable were you with hardware before you got the Pi 5 kit?']
```

To visualize the data and present our findings, we created several plots that provide insight into students' experiences and their improvement in comfort with embedded systems.

### 1. Comfort Levels Before and After Pi 5

The first visualization is a boxplot that compares students' comfort levels before and after using the Raspberry Pi 5, along with their general comfort with embedded systems.

```
plt.figure(figsize=(8,6))
sns.boxplot(data=df[['How comfortable are you with embedded systems?','How comfortable were you with hardware before you got the Pi 5 kit?','How comfortable are you now that you have the Pi 5 kit?']])
plt.xticks([0,1,2], ['Embedded Systems', 'Before Pi 5', 'After Pi 5'])
plt.ylabel("Comfort Level (1-5)")
plt.title("Comfort Levels Before and After Pi 5")
plt.show()
```

### 2. Distribution of Comfort Improvement

Next, we created a histogram to visualize the distribution of comfort improvement. This shows how much students' comfort levels improved after using the Raspberry Pi 5.

```
plt.figure(figsize=(8,6))
sns.histplot(df['Comfort Improvement'], bins=5, kde=True)
plt.xlabel("Improvement in Comfort Level")
plt.ylabel("Count")
plt.title("Distribution of Comfort Improvement After Using Pi 5")
plt.show()
```

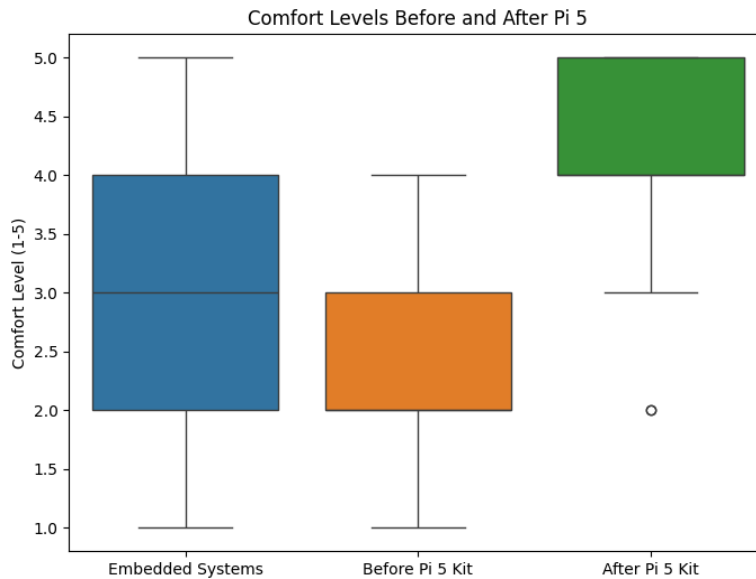
### 3. Average Comfort Levels

We also calculated the average comfort levels before and after using the Raspberry Pi and visualized them using a bar plot.

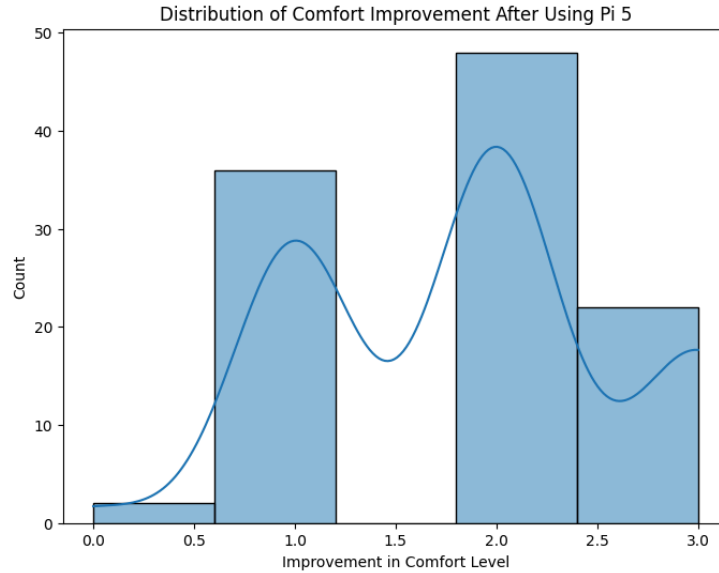
```
avg_comfort = df[['How comfortable were you with hardware before you got the Pi 5 kit?','How comfortable are you now that you have the Pi 5 kit?']].mean()
```

```
plt.figure(figsize=(8,6))
sns.barplot(x=avg_comfort.index, y=avg_comfort.values)
plt.ylabel("Average Comfort Level")
plt.title("Average Comfort Levels Before and After Pi 5")
plt.show()
```

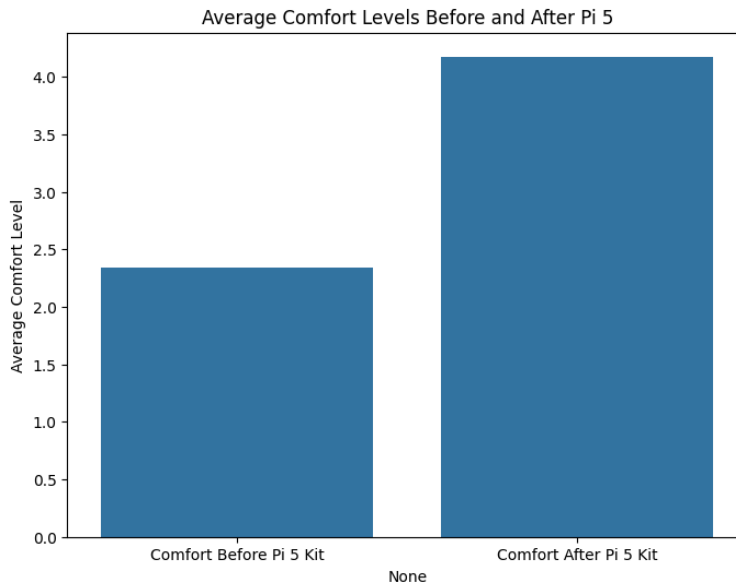
The following plots were generated based on the code provided above. They visually represent the comfort levels before and after using the Raspberry Pi 5 kit, as well as the improvement experienced by the students.



The boxplot above compares comfort levels among three groups: Embedded Systems, Before Pi 5 Kit, and After Pi 5 Kit. The "Before Pi 5 Kit" group has a median comfort level of around 2.5, with a lower quartile of around 2 and an upper quartile around 3. On the other hand, the "After Pi 5 Kit" category shows an increase with a median of around 4.5. This change indicates that the majority of participants began with low confidence but showed enhanced overall confidence in level of comfort "After Pi 5 Kit." The "Embedded Systems" category has a higher range but larger median than in "Before Pi 5 Kit," which perhaps suggests that existing experience with embedded systems is accompanied by greater confidence.



This graph above, a histogram with density curve, shows the distribution of comfort level improvement after using the Pi 5 Kit. The most common improvements are at points 1.0 and 2.0, with peaks at these points. A few individuals saw no improvement (at zero), and some had the highest degree of improvement at around 3 points. The bimodality of the distribution reveals that while most of the participants found improvement to be moderate, there exist distinct groups—a group with minimum confidence gain and a group that had considerable improvement.



This graph above, which is a bar chart, illustrates the average levels of comfort prior to and after using the Pi 5 Kit. The "Before Pi 5 Kit" bar has an average comfort level below 2.5, reinforcing the trend of the first graph that the subjects started with quite low comfort levels. The "After Pi 5 Kit" bar is considerably higher, around 4, indicating a big increase. The gap between these two

bars graphically illustrates the effectiveness of the Pi 5 Kit in increasing the comfort level, reinforcing the trends of the other graphs.

### **Analysis on Educational Impact**

The data above reiterates the perspectives and general observations shared throughout this paper. The improvement in comfort levels with embedded systems and hardware after working with the Raspberry Pi 5 aligns with the common sentiment that the tool is accessible, valuable, and effective in enhancing their understanding of practical computing. These findings support the view that hands-on, experiential learning fosters greater confidence and proficiency, reinforcing the Raspberry Pi's role as a powerful educational tool in computer science.

The Raspberry Pi 5 offers a hands-on approach that allows students to explore a wide range of topics, including networking, the Internet of Things (IoT), operating systems (OS) programming, filesystems, and even running models, making it a powerful tool for computer science education. By engaging with real-world applications, students get an opportunity to apply theoretical knowledge from classrooms, bridging the gap between abstract knowledge and practical implementation.

One of the most important advantages of the Raspberry Pi is its approachability for beginners. Raspberry Pi has an extensive online support system, in the form of articles, videos, forums, and documentation, that makes this technology extremely accessible for students and encourages them to engage in real-world applications of computing concepts. For instance, with the help of online resources, beginners can get started by developing simple projects like controlling LEDs; the integration of hardware and software helps foster an intuitive understanding of computing principles that traditional classroom instruction often lacks. Students can actively seek support and problem-solve and continuously expand their skills by developing more complex projects, instilling a sense of confidence, and deepening technical proficiency.

Raspberry Pi's hands-on, experiential learning approach helps students experiment, troubleshoot, and build projects, fostering skills both technical and soft skills. The skills gained, ranging from writing scripts to automate a task to problem-solving, are highly transferable, ensuring that students are prepared for real-world challenges in the tech industry.

### **Key Takeaways and Recommendations**

#### **Takeaways:**

#### **1. The Raspberry Pi Enables Accessible, Hands-On System-Level Learning**

- a. Perspective 1 (Systems Programming Professor): Professor Angrave underscores the importance of tangible learning in a systems programming context. He notes that the Pi-5 lowers barriers to exploring operating systems, filesystems, and networking, allowing students to engage deeply with core concepts that can be elusive when taught in purely theoretical settings. Because students can flash and

configure the Pi on their own, they develop an immediate familiarity with Linux, scripting, and hardware interaction – skills that extend far beyond the classroom.

- b. Perspective 2 (Systems Programming Course Assistant, Eisha): From Eisha's vantage point, the Pi's low cost and user-friendly ecosystem make it an ideal platform for honors projects in a junior-level Systems Programming course. By standardizing the kit across her student groups, she streamlines the setup process and ensures everyone can focus on the project's embedded systems objectives. The consistent environment reduces the learning curve, enabling students – some entirely new to microcontrollers – to rapidly gain confidence in experimentation and debugging.

## **2. Strong Community and Documentation Foster Self-Directed Learning**

- a. Perspective 1 (Engineering Course Lead Teaching Assistant, Himnish): Himnish reflects on first-year engineering students' initial anxieties around hardware-intensive tasks. The Pi's extensive documentation, official tutorials, and thriving online communities mitigate these fears by offering step-by-step guides, troubleshooting tips, and a variety of project ideas. This ecosystem fosters self-sufficiency: novices can quickly address basic setup issues, while advanced learners tap into specialized forums discussing machine learning and IoT deployments on the Pi.
- b. Perspective 2 (First-Year Undergraduate Researcher, Taniya): Taniya highlights the Pi's community-driven support as crucial for her work in an ongoing STEM accessibility research project. With no prior Raspberry Pi experience, she leaned heavily on Raspberry Pi forums and related resources for resolving technical hurdles, such as remote device access and Gadget Mode configuration. As a result, she could meaningfully contribute to a cutting-edge research endeavor in her very first year, underscoring how supportive resources catalyze rapid skill acquisition.

## **3. Mentorship Structures Elevate Student Engagement and Skill-Building**

- a. Perspective 1 (Systems Programming Course Assistant, Eisha): Eisha's classroom strategy involves pairing student groups with mentors who guide them through challenging technical milestones. Regular feedback sessions ensure that students remain on track yet still have the autonomy to experiment, fail, and discover innovative solutions. This balance between structure and independence fosters not only technical competence but also collaborative skills and resilience – attributes that are vital in real-world engineering settings.
- b. Perspective 2 (Systems Programming Honors Students): Student feedback from the honors section reveals that frequent mentor interactions helped contextualize complex Pi projects and demystify embedded systems concepts. One student credited mentor feedback with shaping caching mechanisms for a FUSE-based Course Explorer Filesystem, while another emphasized the value of planning and parallelizing API requests. The mentorship model thus complements self-directed

exploration, ensuring learners tackle sophisticated tasks without feeling overwhelmed.

#### **4. Real-World Applications Prepare Learners for Industry Demands**

- a. Perspective 1 (Engineering Course Lead Teaching Assistant, Himnish): Himnish observes how Pi-5 projects align closely with contemporary industry needs, especially when projects incorporate network security, data analytics, or IoT workflows. By engaging in small-scale machine learning or sensor-based logging, students experience an end-to-end pipeline – encompassing data collection, modeling, and deployment – much like modern tech environments. This practical exposure bolsters graduates’ readiness for roles that demand interdisciplinary competencies.
- b. Perspective 2 (Systems Programming Honors Students): The Pi-based projects in the honors course, such as a USB Rubber Ducky or a custom network firewall, mirror real-world computing scenarios where security, performance, and low-level operating system knowledge are paramount. Students cited these projects as instrumental in helping them hone debugging strategies and optimize system calls – two skills critical for professional software development. Several students expressed a desire to mentor future classes, indicating that the Pi’s multifaceted use cases inspire sustained engagement and knowledge transfer.

#### **Recommendations:**

##### **1. Incorporate Raspberry Pi Across the Curriculum**

- a. *Introductory Courses*: Introduce first-year students to basic hardware/software setups, letting them gain hands-on confidence through simple, structured tasks (e.g., LED control, sensor input).
- b. *Advanced Courses*: Offer open-ended, collaborative projects where students design, build, and optimize complex systems (e.g., network firewalls, filesystem drivers, embedded AI).

##### **2. Provide Standardized Kits and Support**

- a. *Uniform Hardware*: Supply all students with the same Pi-5 kit to guarantee parity, ease of troubleshooting, and streamlined instructions.
- b. *Resource Repositories*: Maintain a repository of key documentation, tutorials, and sample code so students can reference them throughout the course.

##### **3. Develop a Layered Mentorship Model**

- a. *Peer Mentors*: Assign experienced students or teaching assistants to project groups to facilitate consistent check-ins.

- b. **Structured Feedback Loops:** Schedule regular progress reviews and milestone presentations, ensuring timely identification and resolution of technical or conceptual challenges.

#### 4. Emphasize Debugging and Reflective Practice

- a. **Frequent Checkpoints:** Encourage students to document their debugging processes, reflecting on challenges faced and lessons learned.
- b. **Resilience Training:** Frame troubleshooting as an expected, integral part of embedded systems work to build confidence and perseverance.

#### 5. Leverage Community and Online Tools

- a. **Official Documentation:** Integrate Raspberry Pi Foundation resources into lesson plans.
- b. **Online Forums and Q&A:** Cultivate a sense of community by having students share solutions, code snippets, and troubleshooting tips on class forums or broader public forums (e.g., the official Raspberry Pi forums).

#### 6. Encourage Interdisciplinary Projects

- a. **Data Science and IoT:** Promote collaborations that combine hardware integration with topics like machine learning, sensor analytics, or cloud connectivity, mirroring contemporary industry practices.
- b. **Applied Research Opportunities:** For students interested in research, support the use of the Pi for project prototypes, bridging academic inquiry with practical skill development.

#### 7. Continual Curriculum Refinement

- a. **Collect Student Feedback:** Deploy periodic surveys to capture real-time reflections on project challenges and successes.
- b. **Iterate on Project Scope:** Update assignment guidelines to better match student preparedness, interests, and evolving industry trends, ensuring that project scopes remain engaging, relevant, and feasible.

#### 8. Foster a Culture of Knowledge Transfer

- a. **Student-to-Student Mentoring:** Encourage advanced students and recent alumni to mentor new cohorts, promoting sustainable expertise within the program.
- b. **Document and Showcase Projects:** Archive exemplary student projects in online repositories or departmental showcases, providing inspiration and practical references for future classes.

## **Conclusion**

The collective experiences of Professor Angrave, Eisha, Taniya, and Himnish, along with the student responses collected, demonstrate that the Raspberry Pi 5 is a versatile and transformative educational tool, enabling authentic systems-level learning across a spectrum of technical proficiencies. Its low-cost and supportive ecosystem make it especially well-suited for diverse cohorts, from first-year undergraduates taking their initial steps into hardware-driven projects to advanced students bridging data science and machine learning research.

Through hands-on experimentation, learners develop critical problem-solving, debugging, and collaborative skills, far exceeding the scope of purely theoretical instruction. The Pi's community-driven support structure – encompassing official documentation, forums, and peer mentorship – further bolsters student engagement and autonomy. The Pi's capacity to handle sophisticated tasks, such as speech-to-text inference and IoT applications truly highlights its relevance to cutting-edge research and industry needs.

Ultimately, the Pi-5's integration into engineering curricula holds promise for nurturing a generation of graduates who are both technically proficient and adaptable. By carefully planning mentorship models, logistic frameworks, resource pools, and well-scaffolded assignments, educators can harness the full educational potential of the Raspberry Pi. In doing so, institutions can empower students to bridge the gap between classroom theories and real-world innovation, resulting in a richer, more holistic learning experience that prepares them for the multifaceted challenges of modern technology ecosystems.

## **References and Resources**

[1] Khan, F., & Quweider, M. K., & Qubbaj, A., & Tomai, E., & Zhang, L., & Lei, H. (2020, June), *Infusing Raspberry Pi in the Computer Science Curriculum for Enhanced Learning* Paper presented at 2020 ASEE Virtual Annual Conference Content Access, Virtual On line .  
10.18260/1-2--34828

[2] The Raspberry Pi Foundation, "Learning at home with the Raspberry Pi Foundation – Raspberry Pi," Raspberry Pi. <https://www.raspberrypi.org/learn/>

[3] T.R.P. Foundation, "Teach computing and digital making," Raspberry Pi. <https://www.raspberrypi.org/teach/>

[4] R. Pi Foundation, "Research," Raspberry Pi Foundation, Apr. 08, 2024. <https://www.raspberrypi.org/research/>

[5] Raspberry Pi Foundation, "The Computing Curriculum | Free Teaching Resources | Key Stages 1 2 3 4," Raspberry Pi Foundation, 2025. <https://www.raspberrypi.org/curriculum>

[6] "Raspberry Pi Forums - Index page," forums.raspberrypi.com. <https://forums.raspberrypi.com/>