You Never Forget Your First Project

Scooter Willis, Avera Cancer Institute

New editor Scooter Willis reflects on his journey to becoming an electrical engineer and how we can promote the work of students who've found their passion.

I clearly remember the day I realized I wanted to be an electrical engineer and, more importantly, the project that convinced me. The summer before my senior year in high school, I attended a math and science camp at Florida State University. That particular day, each student was handed a box of wires, microchips, a breadboard, and a wiring diagram—but no instructions, detail, theory, or even hint about what we were being asked to make. I spent a couple of hours getting the breadboard to match the picture, and then connected the power. Three 7-segment displays lit up and the breadboard started counting. That was the moment I knew I’d found my life’s work.

I located my instructor straightaway, informing him that I had a Kaypro II computer that I wanted to use to control a robot. (With no battery and weighing 27 lbs., the Kaypro II—released in 1982 and based on the Z80 microprocessor and CP/M—was one of the first portable computers.) He explained that I could use the parallel port and a 2N222 transistor to control a relay that turned the motor on and off, and then showed me how to wire a double-pole, double-throw relay to run the motor in forward and reverse. In the span of three hours, my life had changed—all because I’d built a clock and someone had shown me how to turn a motor on and off with a parallel port.

THE JOURNEY BEGINS

Upon returning to school in the fall, I decided to design a robotic control system for my science fair project. Back then, the library didn’t have books on building robots and I needed help. So the school administrator picked up the phone directory, looked up “robotics,” and dialed the number for Sally-Corp, a local company specializing in museum and theme...
park animatronics. SallyCorp’s head engineer agreed to answer my many questions. What’s more, he offered me a job programming Z80 assembly language and laying out circuit boards for their animatronic editing system.

In 1984, programming in assembly language was standard for embedded systems, and circuit boards were taped to vellum on a light board. I didn’t mind the assembly language programming but thought the circuit board layout process could be improved. Using one of my school’s new IBM PCs, which ran MS-DOS, I wrote a paint program that let me place circles and lines on the screen and print out photo-etched circuit boards on a laser transparency.

I couldn’t believe how much I’d learned in just six months of mentorship, and eagerly anticipated the engineering classes I’d take in college. Little did I know that the typical path to an engineering degree would fail to meet my expectations.

THE TYPICAL COLLEGE EXPERIENCE
In the fall of 1985, I enrolled at the University of Florida. Enrollment involved a drop/add process in which I—and 35,000 other students—walked around the gym in search of open course sections. I then filled in each section’s circle on a Scantron sheet and lined up to have the sheet scanned in for approval by the mainframe. Some classes required a signature, and thus, a trip to that department; other classes had long lists of prerequisites. As a freshman, I couldn’t take Circuits 1 or Digital Computer Architecture until I’d been accepted into the Department of Electrical Engineering.

Two years later and with all the prerequisite courses completed, I was fully admitted to the department. I signed up for what I expected to be interesting classes, only to discover that they focused on book work and theory. I wouldn’t get to make anything or work on a project until I was in graduate school—another three years away.

At this point, my goal became to finish school quickly and find a job in which I would actually get to build things. The situation’s irony hit home: you could graduate with a bachelor’s degree in electrical engineering and not know how to solder. You definitely wouldn’t know that when you solder, you should breathe out to avoid inhaling the fumes. Furthermore, my classmates didn’t share the same passion for wiring circuit boards and writing code nor did they experience the pure excitement that came from creating a solution to a problem.

Similar challenges exist in computer science: you can graduate without appreciating the importance of writing unit tests or, at a minimum, having a strong opinion about the best integrated development environment (IDE) or the use of tabs versus spaces. (For those not familiar with the tabs-versus-spaces argument, this YouTube video puts it into humorous perspective: goo.gl/pFYufX.)

LIFE AFTER COLLEGE
As chief technology officer of numerous start-ups, I did my share of interviews. A standard question I posed to potential hires was, What’s your favorite editor? The only wrong answer was Microsoft Word, because they’d obviously misunderstood the question. If they said NetBeans, my personal favorite, then their odds of getting the job increased. If they answered Emacs or vi, I knew I’d found a programmer with strong Linux skills who could also do system administration work. (For details on the editor war, see en.wikipedia.org/wiki/Editor_war.) Having a favorite editor or IDE signifies experience and was a stronger indicator of success than an outstanding grade-point average.

Sadly, getting an engineering job straight out of college at a company willing to invest in job-specific training is no longer the norm. Start-ups are looking for hires with specific skills that can be put to use the first day on the job.

OUR FUTURE: THE NEXT GENERATION OF STUDENTS
During my formative years—the early 80s—I worked on my bike and built forts and skateboard ramps. Playing video games involved a trip to the mall with a pocket full of quarters. In contrast, today’s typical first-grader has a room full of Legos and has mastered complex video games, having spent countless hours watching YouTube videos to learn the tricks. The next generation of students—having been exposed early in life to technology and problem solving—wants to tackle new challenges but is less interested in being told how. Our educational institutions have been slow to change and embrace these students’ true potential.
I’ve seen firsthand the effect of technology competitions on young people’s lives. For 20 years, I’ve been a mentor in FIRST robotics (www.firstinspires.org), which holds robotics competitions for 400,000 K-12 students worldwide. For example, in the FIRST LEGO League, teams build a Lego EV3 robot with a range of sensors and LabVIEW block programming. FIRST Tech Challenge teams use an Android-based robotics control system—which includes an augmented reality software development kit that uses the phone’s camera and object recognition—to track a robot’s position and field orientation. And FIRST Robotics Competition teams have six weeks to build a robot that then competes in tournaments. VEX Robotics offers a similar competitive robotics program with 10,000 teams worldwide (www.roboticseducation.org). The Science Olympiad hosts 7,300 middle- and high-school teams at 33 different events, 9 of which focus on building or programming (www.soinec.org). These programs all start with a common goal: to get middle- and high-school students hooked on the thrill of competition by presenting them with a challenge and specifications that define the problem. After high school, there are fewer such STEM-related competitions available.

As this column’s new editor, I see an opportunity to promote the work of students who’ve found their passion. More importantly, I hope to raise awareness about college-level engineering competitions—and increase the number of electrical engineering graduates who know how to solder and computer science graduates who have a favorite editor.

SCOOTER WILLIS is the director of computational biology at the Avera Cancer Institute and founder of STEM HQ. Contact him at willishf@gmail.com.

2017 B. Ramakrishna Rau Award
Call for Nominations

Honoring contributions to the computer microarchitecture field

New Deadline: 1 May 2017

Established in memory of Dr. B. (Bob) Ramakrishna Rau, the award recognizes his distinguished career in promoting and expanding the use of innovative computer microarchitecture techniques, including his innovation in compiler technology, his leadership in academic and industrial computer architecture, and his extremely high personal and ethical standards.

WHO IS ELIGIBLE? The candidate will have made an outstanding innovative contribution or contributions to microarchitecture and use of novel microarchitectural techniques or compiler/architecture interfacing. It is hoped, but not required, that the winner will have also contributed to the computer microarchitecture community through teaching, mentoring, or community service.

AWARD: Certificate and a $2,000 honorarium
PRESENTATION: Annually presented at the ACM/IEEE International Symposium on Microarchitecture
NOMINATION SUBMISSION: This award requires 3 endorsements. Nominations are being accepted electronically: www.computer.org/web/awards/rau.
CONTACT US: Send any award-related questions to awards@computer.org.

www.computer.org/web/awards/rau