5-2022

Teaching Programmable Microcontrollers to Novice Users in a College of Agriculture: Effects on Attitude, Self-Efficacy, and Knowledge

Grant T. Hood
University of Arkansas, Fayetteville

Teaching Programmable Microcontrollers to Novice Users in a College of Agriculture: Effects on Attitude, Self-Efficacy, and Knowledge

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Agricultural and Extension Education

by

Grant T. Hood
Purdue University
Bachelor of Science in Agricultural Education, 2018

May 2022
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

____
Donald M. Johnson, Ph.D.
Thesis Advisor

____
George W. Wardlow, Ph.D.
Committee Member

____
Chris Estepp, Ph.D.
Committee Member

Abstract

This thesis consists of two articles that examined an instructional treatment based on the use of Arduino UNO R3 programmable microcontrollers in a fundamentals of agriculture systems technology course at the University of Arkansas. The first article examined students' breadboarding and programming self-efficacy and knowledge of Arduino. The treatment consisted of a three-class-period instructional treatment, starting with a pretest before instruction to measure students' baseline interest, knowledge, and self-efficacy of breadboarding and programming Arduino. This was followed with a short 30-minute instructional video explaining basic Arduino programming and breadboarding. Next a hands-on laboratory activity requiring students to breadboard and program an LED circuit was conducted. The activity was graded and rubrics were returned to the students before they took the posttest. Students' mean scores for breadboarding and programming self-efficacy and Arduino knowledge were higher after the instructional treatment, while the observed mean for interest slightly declined.

The second article examined the rubric scores from the hands-on laboratory activity and evaluated where students most commonly made errors breadboarding and programming. Rubric scores on Arduino breadboarding were 58.5% and programming 23.5%, leading us to conclude that students needed more instruction on Arduino programming and in breadboarding simple electronic circuits. The single most common error made when programming was the lack of writing simple comments at the end of each line of the program sketch to describe what the command is doing. The second most common error in programming was not writing the command to correctly identify a digital pin as an output. For breadboarding, the two most

common errors were that students were unable to correctly “forward-bias” an LED and wire a single 240ohm resistor in series in the circuit.

Both articles produced findings worth implementing into a future redesigned study where novice agriculture students are introduced to basic electronics circuitry followed by Arduino programming. Readers should design instruction that provides students with the opportunity for mastery experiences like breadboarding and programming success during instruction prior to an individual hands-on task. The instructional treatment should be extended in time to allow students more opportunity to process new knowledge. The hands-on activity should be simplified to include only one LED circuit, and the reference sheet should show more complete examples of programming. Students should be encouraged to work together on the hands-on activity rather than being left to work individually.

Acknowledgements

Special thanks to Dr. Donald Johnson, for without him this thesis would not have been possible. Thank you to Dr. Christopher Estepp and Dr. George Wardlow, my thesis committee and teaching mentors, I have learned so much from you all. To the rest of the AECT department, thank you for your continual support and encouragement to get me through this degree.

I would also like to thank Dr. Sarah LaRose of Purdue University for leading me to the University of Arkansas. Despite my struggles in your classroom you still saw my potential and for that I am forever grateful.

Furthermore, I would like to thank my grandparents Larry and Beth Dawson, for constantly calling and checking on me while I have been away. I would also like to thank Karli Stringer, for being such a great friend and always being there for me during my time in Arkansas. To my neighbor Jim Hockemeyer, for his support and commitment to my success in this degree. To James and Abigail Fairchild, Jake Wicker, Mason Gordon, Henry Akwah, and the students of the University of Arkansas who have made this campus feel like a home to me over the past two years, thank you.

Finally, I would like to thank my parents, Todd and Heidi Hood, my brother Garth Hood, and my Grandma Connie Hood, for their continual love and support of me.

To everyone else along the way that called to check-in and see how my time was going here in Arkansas, thank you for being a part of my success.

Table of Contents

List of Figures

List of Tables

Chapter One: Introduction

The agricultural and food industry is currently undergoing a transformation where technology and software systems are driving many automated tasks and production lines, reducing the cost of production of agricultural goods and increasing profits (Suprem et al., 2012). However, many agriculturalists believe that the adoption of technology comes at a higher purchasing costs when in reality this is not the case (Titovskaya, et al., 2019). Many of these technologies have been around long enough that they are relatively inexpensive and user friendly, the cost is only an issue to those that are unskilled and inexperienced.

One example of these technologies are microcontrollers, found in many agricultural applications such as food packaging systems (Suprem et al., 2012), tractors and machinery (Goering et al., 2003), field robotics (Suprem et al., 2013), variable rate technology (VRT), and smart irrigation control systems (Goap et al., 2008). Programmable microcontrollers have transformed the agriculture industry and added to the diverse skillset that is necessary of today's agricultural workforce. This calls for the curriculum to be modified to include microcontroller programming and basic electronics circuit building to better prepare agricultural systems technology students to work in the industry where these technologies are prevalent (Titovskaya, et al., 2019). College of agriculture students need exposure to microcontroller programming during an undergraduate degree program to better prepare them for a technology-driven future. Microcontroller programming is vocational education and should be taught as an important skill, not only for the future of technology but also as a way to help students think in a more logical computational thinking order (Santosa & Waluyanti, 2019).

Successful instruction in programming can produce adequately prepared students to work in the agriculture technology industry (Weidenbeck, 2005). Thus, it is important to understand the level of interest that novice college of agriculture students have about programming and breadboarding simple microcontroller circuits, as well as, students' self-efficacy and knowledge about programming, in order to effectively teach students these challenging skills.

Problem Statement

Microprocessors are ubiquitous in consumer products and in industrial and agricultural applications. Therefore, agriculture students need a basic introduction to microcontrollers and programming language to ensure that agricultural graduates possess skills relevant to the agricultural workforce. However, there is a lack of research regarding programming among novice agriculture students and their understanding of microcontrollers, end-user programming, and electronics circuit building skills. Widenbeck (2005) said, "To support students in the introductory programming course, whether majors or non-majors, we need to understand the cognitive and social-cognitive factors that affect their success in learning" (p.13). There is a lack of research on novice programming among agriculture students and their understanding of microcontrollers, end-user programing, and electronics circuit building skills. Identifying the relationship between breadboarding and programing errors may help to unlock the most effective modes of instruction (Booth et al., 2016).

Originally designed in Italy, in 2005 as a more affordable option for programming students, the Arduino is an open-source microcontroller developed with novice programmers in mind. Replacing the older, more expensive, and less powerful Parallax "basic stamp" microcontrollers (Popiel, 2015). Arduino is one of the most commonly used microcontrollers in the classroom and in the agriculture industry, due to their affordability, ease of use, and open source programing environment (Titovskaya, et al., 2019). With over 25 different platforms, and sold under a Creative Commons license, Arduino is the choice for novice programmers (DesPortes and DiSalvo, 2019).

By teaching novice agriculture students Arduino programming, we are introducing students to skills that make them valuable to the workforce. Not only does programming diversify graduates' career skills, but it also fosters the development of students' computational thinking. Understanding students' self-efficacy, interest, knowledge, and common errors in a hands-on Arduino activity will guide educators in developing instruction for novice programming students.

Purpose

The purpose of this study was designed to better understand how novice agriculture students learn embedded computing and programming, and to understand where common errors were made when completing an instructional activity. Understanding the effects of an instructional treatment of novice agriculture students' interest, self-efficacy and knowledge in Arduino programming will help guide educators in developing curriculum that better suits the needs of novice programmers. The results of this study will be used to aide in the development of further novice instructional materials and practices for Arduino and similar technologies in a college of agriculture.

Objectives

This research was designed with three objectives in mind. The first objective was to determine the effects of an instructional treatment on novice agriculture students' interest, self-efficacy, knowledge, on Arduino. The second objective of this study was to examine the relationships between the above variables and student rubric scores on an Arduino activity. The third objective of the study was to identify common mistakes made by novice students breadboarding and programming of an Arduino task.

Limitations

This study focused on only one class of novice programing students in a university college of agriculture fundamentals of agricultural systems technology course. Therefore, the results of this study have limitations that make it relevant only to novice programmers, more specifically novice agriculture students.

Key Definitions

Arduino UNO R3: Common, affordable, easy to use programable microcontroller on a printed circuit board (Arduino, 2022).

Breadboarding: Using a solderless circuit board, wires, and resistors along with sensors and other components to construct temporary or prototype electronic circuits (Sedas et al., 2021).

Microcontroller: A slower, smaller, and more affordable industry version of the microprocessor which can be used with a variety of different sensors and electronic controls (O'Rourke, 2005).

Microprocessor: A computer processor located on an integrated-circuit chip (Merriam-Webster, 2022).

Novice programmer: A student that is learning programming for the first time (Sim et al., 2021).

Sketch: Term used to describe the program of the Arduino IDE, written in a language similar to C++ (Badamasi, 2014).

References

Badamasi, Y. A. (2014, September). The working principle of an Arduino. In 2014 11th international conference on electronics, computer and computation (ICECCO) (pp. 1-4). IEEE.

Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Prentice-Hall.

Booth, T., Stumpf, S., Bird, J., & Jones, S. (2016, May). Crossed wires: Investigating the problems of end-user developers in a physical computing task. In *Proceedings of the 2016 CHI conference on human factors in computing systems* (pp. 3485-3497).

Des Portes, K., & DiSalvo, B. (2019). Trials and Tribulations of Novices Working with the Arduino. In Proceedings of the 2019 ACM Conference on International Computing Education Research (pp. 219-227).

Epstein, D., & Klerman, J.A. (2013). When is a program ready for rigorous impact evaluation? The role of a falsifiable logic model. Evaluation Review, 36(5), 375-401. doi: 10.1177/0193841X12474275

Goap, A., Sharma, D., Shukla, A. K., & Krishna, C. R. (2018). An IoT based smart irrigation management system using machine learning and open source technologies. *Computers and Electronics in Agriculture, 155*(2018), 41-49. https://doi.org/10.1016/j.compag.2018.09.040

Goering, C. E., Stone, M. L., Smith, D. W., & Turnquist, P. K. (2003). *Off-road vehicle engineering principles*. American Society of Agricultural Engineers.

K. Sreeram, R. S. Kumar, S. V. Bhagavath, K. Muthumeenakshi and S. Radha, "Smart farming—A prototype for field monitoring and automation in agriculture," 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2017, pp. 2189-2193, doi: 10.1109/WiSPNET.2017.8300148.

Merriam-Webster (2022)

O'Rourke, N., Hatcher, L., & Stepanski, E. J. (2005). *Using SAS for univariate and multivariate statistics.* SAS Institute, Inc.

Popiel, G. (2015). Introduction to the Arduino Microprocessor. *Qst*, *99*, 30-32. Retrieved from https://www.proquest.com/magazines/introduction-arduino-microprocessor /docview/1730617252/se-2?accountid=8361

Santosa, E. S. B., & Waluyanti, S. (2019, November). Teaching Microcontrollers using Arduino Nano Based Quadcopter. In *Journal of Physics: Conference Series* (Vol. 1413, No. 1, p. 012003). IOP Publishing.

Sedas, R. M., Peppler, K., & Thompson, N. (2021). Breadboards and Paper Circuits: Differences in Advanced Circuitry Learning and PCB Layout Design. In *Proceedings of the 15th International Conference of the Learning Sciences-ICLS 2021*. International Society of the Learning Sciences.

Sim, T. Y., Teow, M., & Lau, S. L. (2021, November). Unplugged Computational Thinking Activities Framework Development for Novice Programmer. In *2021 IEEE International Conference on Computing (ICOCO)* (pp. 297-302). IEEE.

Suprem, A., Mahalik, N., & Kim, K. (2013). A review on application of technology systems, standards and interfaces for agriculture and food sector. *Computer Standards & Interfaces*, *35*(4), 355-364.

Titovskaya, N. V., Titovskaya, T. S., & Titovskii, S. N. (2019). The prospects of microprocessor application in the Agriculture Digitalization. IOP Conference Series: Earth and Environmental Science, 315(3), 032011. https://doi.org/10.1088/1755-1315/315/3/032011

Wiedenbeck, S. (2005, October). Factors affecting the success of non-majors in learning to program. In Proceedings of the first international workshop on Computing education research (pp. 13-24).

Chapter 2: Effects of an Instructional Treatment on Novice Programmers in an Agricultural Systems Technology Course

Introduction

In the world of agricultural education, a new skill set is being added to the agricultural toolbox of skills; microprocessor technology. Agriculture, food, and natural resources (AFNR) systems are shifting in method and design at an unprecedented rate (Dennis, Aguilera, & Satin, 2009). Inclusion of new and emerging technologies within AFNR curricula is essential to empowering learners for future success (King et al., 2019). Colleges across the United States have long faced the challenge of maintaining up-to-date curriculum amidst a quickly evolving agricultural industry which must progress in order to sustain the increasing world-wide demand for agricultural products (National Research Council, 2009). One new technology appearing in agriculture is microcontrollers. Microcontrollers are being widely used in agricultural applications as embedded computing systems for the advancement of technological practices. Example applications include tractors (Goering et al., 2003), smart irrigation systems (Goap et al., 2018), field robots (Suprem et al., 2013), and variable-rate applicators (Schumann, 2010).

Simply put, microcontrollers are integrated circuit devices that contain a microprocessor, memory, and peripherals which can receive inputs and control outputs within a system (Keim, 2019). As these modern technologies continue to achieve a greater presence in the agricultural industry, undergraduate agricultural students should develop a basic understanding of these technologies as they prepare for a career in a field dominated by microprocessors and embedded computing.

Within the world of microprocessors is a vast variety of controllers and software. Arduino, one type of microprocessor, is a programmable, open-source microcontroller and software program that is widely used in all ages and levels of education (Al-Abad, 2017). Arduino is user-friendly compared to other systems; however, barriers still exist as novice users experience unfamiliar concepts and techniques involved in learning computer programming (Thomas et al., 2011).

Mercier (2015) stated that college graduates should be prepared for the disciplines within the agricultural and food science fields in which they study. Further, recent studies like Stripling and Ricketts (2016) have called for more research to provide a stronger support for the development of a scientific workforce. As a result, the purpose of this study was to determine the effects of an instructional treatment on interest, self-efficacy, and knowledge of novice Arduino users in a college of agriculture. The results of this study will be used to guide and refine future teaching and learning experiences using this new and emerging, important technology in colleges of agriculture.

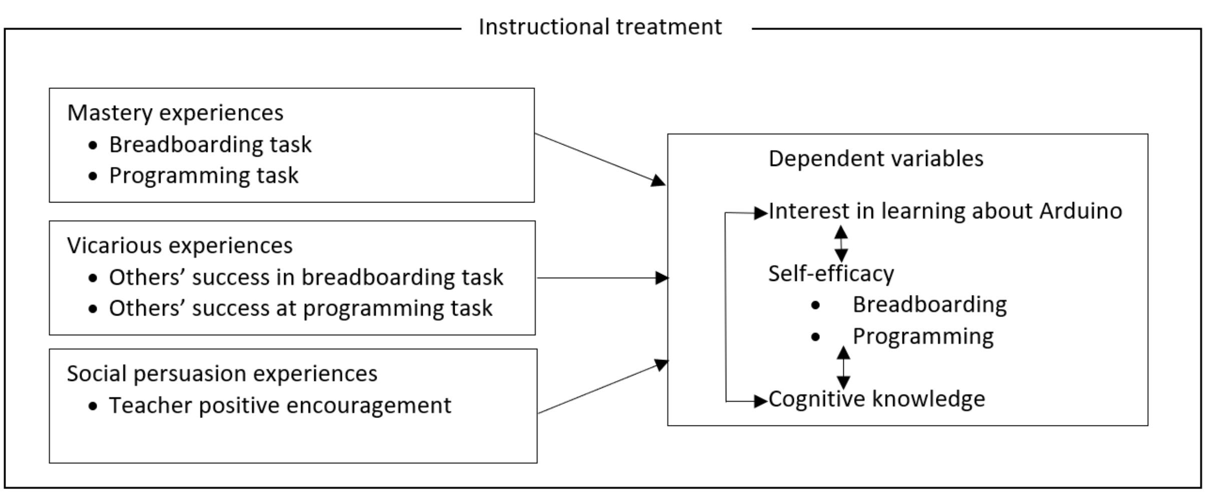
Theoretical Framework

This study was guided by Bandura's (1986) self-efficacy theory. According to Bandura, self-efficacy is an individual's assessment of their own ability to successfully achieve a desired outcome when engaged in a task or activity. Self-efficacy is affected by mastery experiences, vicarious experiences, and social persuasion (McKim and Velez, 2016). The most powerful effect on self-efficacy comes from mastery experiences, which occur when an individual has personal success in accomplishing a task. Vicarious experiences, on the other hand, occur when an individual observes others like them successfully accomplish a task. Finally, social persuasion experiences are when a trusted individual expresses confidence the individual can successfully accomplish a task.

Figure 1 illustrates how Bandura's (1986) self-efficacy theory was applied for this study. Following classroom instruction, students had the opportunity to engage in mastery experiences with breadboarding and programming within the context of the hands-on activity. Students also had vicarious experiences as they watched classmates achieve success in the breadboarding and programming tasks; these successes were announced by the instructor. Finally, students developed social persuasion experiences as the instructor made encouraging comments as the students worked. Example comments included, "Great job on breadboarding - many of you have your circuits correctly breadboarded," and "You're getting the hang of programming - many are just a step away from having it correct." Previous research has found positive relationships between self-efficacy, interest, and learning in academic subjects (Lee et al., 2014). Therefore, this model assumed positive intercorrelations between the dependent variables; interest, self-efficacy, and knowledge of Arduino.

Figure 1

Bandura's (1986) Self-efficacy Theory as Applied to the Arduino Study



According to McLaughlin's et al. (2005) theory of student content engagement, which states that instruction should be developed with the content and activity in mind to increase students' motivation, ability to process new knowledge, and decrease stress resulting in an overall increased engagement in learning. It is known that learning must start with some form of prior knowledge that already exists in the student, this is known as constructivism. Learning takes place when a student begins to question his or her past experiences and starts to build new knowledge (Doolittle & Camp, 1999). Student motivation can be impacted by the level of difficulty of the content and the activity. The short, three-class session design of this study is guided by the occasion for processing new information by providing students with an engaging curriculum, and challenging hands-on activity. The results of this study will aide in the design of future programming and breadboarding instruction and hands-on activities (McLaughlin et al., 2005).

Purpose and Objectives

The purpose of this study was to understand the effects of an instructional treatment on novice agriculture students' interest, self-efficacy and knowledge of Arduino programming to guide educators in developing curriculum that better suits the needs of novice programmers. This study has two objectives; the first objective was to determine the effects of an instructional treatment on novice agriculture students' interest, self-efficacy, and knowledge, about circuit breadboarding and Arduino programming. The second objective of this study was to examine the relationships between these variables and student rubric scores on an Arduino activity.

Two null hypotheses were stated for statistical testing of H_{01}: an instructional treatment (lecture and laboratory activity) will have no significant ($p < .05$) effect on novice agriculture students' interest in learning about Arduinos, breadboarding self-efficacy, programming self-efficacy, or knowledge of breadboarding and programming as measured by pretests and posttests, and H_{02}: there will be no significant ($p < .05$) relationship between students' interest in learning about Arduinos, breadboarding self-efficacy, programming self-efficacy, knowledge of breadboarding and programming and students' rubric scores for breadboarding and programming.

Methodology

Students enrolled in the University of Arkansas College of Agricultural, Food, and Life Sciences undergraduate students served as the population for this study. The accessible sample consisted of all students (*n* = 41) enrolled in the Fall 2021 fundamentals of agricultural systems technology course. Students consenting to participate (*n* = 28) were randomly divided into two separate groups A and B. This study was conducted during the second week of the Fall 2021 academic semester and extended over three days of instruction.

This study consisted of two instruments based on an interest inventory developed by Gable and Roberts (1983) and the programming self-efficacy instrument developed by Kittur (2020). The instrument consisted of four major sections to measure interest in Arduino (13 items; pretest *a* = .88; posttest *a* = .93) such as, "I liked learning about Arduino" and "Learning about Arduino gave me skills I will use in life". A five-point Linkert-type scale was used, where a 1 was strongly disagree and a 5 was strongly agree. Part II measured students' confidence levels in Arduino programming; 8 items (pretest *a* = .94; posttest *a* = .88). Example questions included, "Write Arduino sketch statements that use correct syntax" and "Explain the basic logical structure of an Arduino sketch". This was also measured using a five-point Linkert-type response scale, where 1 was Very Unconfident and 5 was Very Confident. Part III measured students' confidence in building Arduino Circuits on a breadboard, 9 items (pretest *a* = .93; posttest *a*

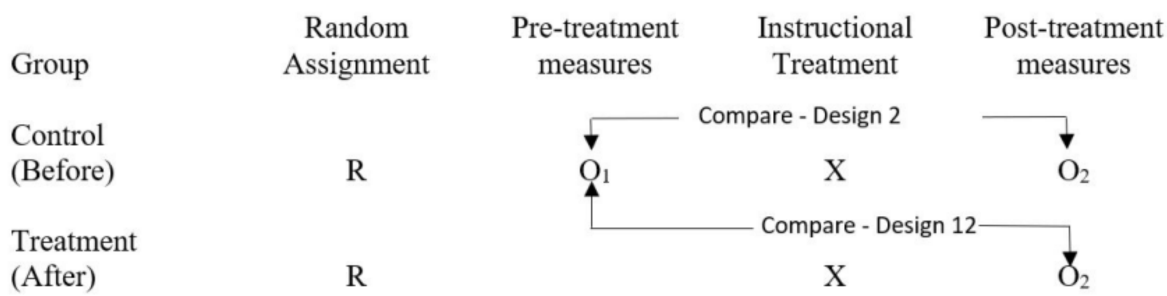
=.94). With questions like, “Connect a specific analog pin from the Arduino UNO to a breadboard” and “forward-bias a light-emitting diode (LED)”, using the same Linkert-type 1-5 scale as above. For part IV, background information confirmed each student’s year in school, gender, and any type of programming training prior to the class. The final question asked students whether they had any prior hands-on experience with programming. The self-efficacy scales were created using recommendations from Bandura (2006).

The knowledge pre/posttest was developed to measure students’ knowledge about Arduino programming and breadboarding. This instrument consisted of 15 multiple choice questions, 14 of which measured students’ cognitive knowledge of Arduino’s and the 15th question confirmed their novice status by asking students to describe their confidence level by choosing from one of the following: “I am not at all confident that my answers are correct, I am fairly confident that my answers are correct, and I am extremely confident that my answers are correct”. The same cognitive knowledge pretest was administered again as a posttest with the questions randomized the second time. KR-21 reliabilities were .43 for the pretest and .64 for the posttest. The low reliability on the pretest suggested that students guessed as a way of completing the test (Paek, 2014). A final question of the pretest confirmed that 93.8% of students (all but one) were “not at all confident” their answers were correct.

As shown in figure 2, this study used a separate-sample pretest-posttest group design from (Campbell & Stanley, 1963; design 12) to compare the results of pretest and posttest knowledge across groups. A one-group pretest-posttest design was used to measure differences before and after instruction (Campbell & Stanley; design 2). External validity is controlled using Design 12 except for history, maturation, and the interaction of history and maturation. Since the study was conducted within a weeks’ time, history was ruled out as a threat, as were maturation and the interaction between the two (Campbell & Stanley; design 12). Design 2, while a weaker design, acted as an internal replication within the control group.

Figure 2

Campbell & Stanley (1963) research design 2 and design 12 as used in this study



For the third instrument, a rubric was developed to evaluate students' breadboarding and programming skills on the hands-on activity. The rubric consisted of 27 possible points, 10-points for breadboarding, 15 points for programing, and two points for the correspondence between the breadboarded and the programed circuit. The first 10 points of the rubric included items such as "circuit is wired to a digital pin" and "resistor is connected in series with LED." The second part of the rubric contained 15 points for the programming and assigned one point to each line of the programming sketch such as "digitalWrite for Blue LED 500ms." The final sections of the rubric contained two points and determined if the sketch corresponded with the breadboarding. The researcher graded each activity using this rubric; while a second researcher randomly chose five completed activities and graded them using the same rubric. The Cohen's kappa coefficient for the rubric were .87 and 1.0, respectively, indicating near perfect and perfect agreement (Cohen, 1960).

Overview

Twenty-eight students agreed to participate in the study, which was conducted during the second week of a fundamentals of agricultural systems technology course, which met Mondays, Wednesdays, and Fridays for 50-minute class periods. The first-class meeting met virtually on Zoom and consisted of a pre-recorded video lecture. The second-Class meeting of this study took place in person, in a computer lab where students had access to a computer with the Arduino IDE software already downloaded and ready for use. On the third and final class meeting students, met in their traditional classroom to debrief and discuss the mastery activity.

Monday

Both groups met virtually on Zoom and were given the same pre-recorded 30 minute instructional lesson on basic Arduino breadboarding and programing, which introduced students to terms and definitions, and how to use programmable microcontrollers. The treatment group A was given the 15 question pretest to measure baseline self-efficacy, interest, and knowledge of Arduino breadboarding and programming. In an effort to maintain consistency, participants of group B took a placebo test before instruction. After instruction, students were prepared to meet on Wednesday, where they were told that they would complete a mastery activity on what they had learned from the instruction.

Wednesday

For the second-class meeting, students met in a campus computer lab where they had access to their own computer with the Arduino IDE programing software downloaded. Due to restrictions on the number of people allowed in the lab at one time, students met at one of two succinct lab sessions selected by the student's availability, each lasting 40 minutes. Upon entry, students found an activity sheet, reference sheet, and bag of supplies at their workstation. These were all the necessary components to successfully breadboard and program the activity using only their knowledge from the previous video lesson, the IDE programming help menu, and the provided reference sheet that gave examples of basic Arduino breadboarding and programming. The activity asked students to construct circuits on a breadboard with two separate light-emitting diodes (LED), one blue and one red. Students were asked to blink the blue LED on and off twice, followed by the red LED blink on and off once. Students had to program a digitalWRITE code specifying which digital output pin they would like to provide with power as well as declaring a digital pin as an output with a pinMode_ command. Each supply bag contained: one Arduino Uno programmable microcontroller with breadboard secured to a mounting plate, one blue LED, one red LED, two resistors, six breadboarding wires, and a USB 2.0 programming cable. The instructor in the room provided words of encouragement throughout the activity, announcing students' progress as they accomplished different levels of the activity. Students were not allowed to use outside resources or talk to their neighbors during the activity. Students were asked to sit every other chair to prevent any talking or cheating.

Upon completion of the second day, the researcher evaluated each participant's completed circuit board using the breadboarding and programming rubric. Grades were recorded and intact, completed circuit boards were collected.

Friday

On third day students met again in a traditional classroom where they were debriefed on the activity and graded rubrics were handed back to students. The instructor reviewed the activity with the students and showed an example of what a successfully breadboarded and programmed activity looked like. To measure students' interest, knowledge, and self-efficacy again after instruction the same pretest was administered as a posttest, with the questions reordered to maintain accuracy.

Data Analysis

For Design 12, a one-way MANOVA was used to test for significant ($p < .05$) differences between groups R_A and R_B on measures O_1 (for R_A) and O_2 (for R_B). A significant MANOVA was followed by univariate ANOVAs to identify the dependent variables where the groups differed (O'Rourke et al., 2005). For Design 2, paired t-tests, with the Bonferroni correction (Field & Miles, 2012), were used to test for significant ($p < .0125$) differences between O_1 and O_2 for group R_A only. The dependent variables were interest in Arduino, breadboarding self-efficacy, programming self-efficacy, and Arduino breadboarding and programming knowledge. We also examined the intercorrelations between the independent and dependent variables.

Results

Twenty-six students from the fundamentals of agriculture systems technology course during the fall semester of 2021 at the University of Arkansas completed all parts of the study; (n = 26) two students in group R_A completed the pretest, but missed a later class period. Pretest scores for these two students were included in the analysis for Design 12 only, leaving 16 students in group R_A for O_1 and 12 students for group R_B for O_2. In Design 2, only 14 students in group R_A completed all parts of the study and were included in the results. The majority of students (n = 28) indicated that they were freshmen (23.1%) or sophomores (30.8%), with an equal representation of male and females, and100% of students confirmed their novice status with Arduinos prior to the study.

Students from both groups completed a knowledge, self-efficacy, and interest posttest, and students from the control group (group A) completed the same knowledge, interest, and self-efficacy pretest prior to instruction. The results of students' pre and posttest, as well as their rubric scores of the mastery activity, were evaluated and compared. Results from student’s pretest and posttest scores were compared as well as students’ rubric scores to determine which breadboarding and programming tasks students were successful in, and if student success had any effect on self-efficacy. The rubric scores were also analyzed to determine if there was any relationship between breadboarding and programming errors.

Design 12: Separate-Sample Pretest-Posttest Design

Table 1 shows the observed pretest and posttest means for students’ interest in learning about Arduino, breadboarding self-efficacy, programming self-efficacy, and knowledge of Arduino. Means for breadboarding self-efficacy, programming self-efficacy, and Arduino knowledge were higher after the instructional treatment (group R_B) than before (group R_A), while the mean for interest in Arduino decreased slightly.

Table 1

Observed Pretest and Posttest Scores for Interest, Breadboarding Self-Efficacy, Programming Self-Efficacy, and Knowledge

	Group A (pretest)		Group B (posttest)	
Measure	*M*	*SD*	*M*	*SD*
Interest in learning about Arduino[a]	3.46	0.45	3.36	0.74
Breadboarding self-efficacy[b]	1.75	0.69	2.80	0.97
Programming self-efficacy[b]	1.96	.078	2.30	0.84
Knowledge about Arduino[c]	38.8%	17.3%	68.9%	19.1%

[a] Measured on a 1 - 5 scale where 1 = strongly disagree and 5 = strongly agree. [b] Measured on a 1 - 5 scale where 1 = very unconfident and 5 = very confident. [c] Percent correct on a 14-item test.

A one-way MANOVA, between-groups design was used to tested the null hypothesis of no effect of the instructional treatment on any dependent variable. Results showed a significant difference between groups for one or more dependent variables, Wilkes’ Lambda = 0.43, p <

.001. Resulting univariate ANOVA's indicated significant increases in after-treatment scores (group R_B) for breadboarding self-efficacy [$F(1, 25) = 9.99$, $p = .004$], and knowledge [$F(1, 25) = 16.60$, $p < .001$]. Breadboarding self-efficacy ($\eta^2 = 0.29$) noted a large effect after the instructional treatment (Cohen, 1988), and knowledge of Arduino ($\eta^2 = .40$). No significant differences were observed on student interest [$F(1,25) = 0.18$, $p = .67$] or programming self-efficacy [$F(1,25) = 0.80$, $p = .38$].

Design 2: One Group Pretest-Posttest Design:

This design evaluated observed pretest and posttest scores within group A, on each of the four dependent measures for students ($n = 14$) who completed all components of the study. Table 2 shows the observed mean scores and how self-efficacy of breadboarding, programming, and knowledge all increased., while the mean score for students' interest in learning about Arduino slightly decreased from pretest-posttest.

Table 2

Pretest and Posttest Scores for Interest, Breadboarding Self-Efficacy, Programming Self-Efficacy, and Knowledge for Group A only

	Pretest		Posttest	
Measure	*M*	*SD*	*M*	*SD*
Interest in learning about Arduino[a]	3.46	0.45	3.35	3.35
Breadboarding self-efficacy[b]	1.75	0.69	3.00	3.00
Programming self-efficacy[b]	1.96	0.78	2.32	2.32
Knowledge about Arduino[c]	38.8%	17.3%	69.6%	17.4%

[a] Measured on a 1 - 5 scale where 1 = strongly disagree and 5 = strongly agree. [b] Measured on a 1 - 5 scale where 1 = very unconfident and 5 = very confident. [c] Percent correct on a 14-item test.

Four paired t-tests were conducted to test the null hypothesis of no effect of the instructional treatment on any dependent variable such as attitude towards Arduino, breadboarding & programming self-efficacy, and knowledge. These results indicated significantly higher scores for breadboarding self-efficacy and knowledge of Arduino. The *alpha* level was established at .0125 using the Bonferroni correction for each test to maintain an overall experiment rate of .05 (Field & Miles, 2012). Significant increases in the posttest scores were noted for breadboarding self-efficacy [$t(14) = 6.42$, $p < .001$] and knowledge [$t(15) = 5.92$, $p < .001$]. For the instructional treatment, Cohen's *d* for breadboarding self-efficacy ($d = 1.68$) and knowledge ($d = 1.48$) indicated large effects (Cohen, 1998). There was no significant difference between pretest-posttest scores on interest [$t(15) = -0.76$, $p = .46$] and programming self-efficacy [$t(14) = 1.60$, $p = .13$].

Rubric scores from all students' (n =26) mastery activities ranged from 20% to 100%, with a mean score of 58.5% (*SD* = 24.0%). Rubric scores on programming varied from 0% to 100%, with a mean score of 23.5% (*SD* = 36.0%). According to Davis' (1971) conventions, there were significant correlations ranging from moderate to very strong between rubric scores and the dependent variables, which are shown in Table 3. Breadboarding and programming were highly correlated (r = .83) and there were very strong correlations between programming (r = .73) and breadboarding (r =.77) self-efficacy and interest in learning more about Arduino. The breadboarding and programming achievement by rubric score were not significantly correlated and had significant, but lower correlations with interest in learning more about Arduino (r= .46 and .45, respectively).

Table 3

Intercorrelations between Rubric Scores, and Posttest Interest, Self-efficacy, and Cognitive Test Scores

Variable	X1	X2	X3	X4	X5	X6
Breadboarding rubric score (X1)	1.0	$.22^{NS}$	$.52^{**}$	$.52^{**}$	$.46^{*}$	$.59^{**}$
Programming rubric score (X2)		1.0	$.36^{NS}$	$.49^{*}$	$.45^{*}$	$.66^{***}$
Breadboarding self-efficacy (X3)			1.0	$.83^{***}$	$.77^{***}$	$.54^{**}$
Programming self-efficacy (X4)				1.0	$.73^{***}$	$.54^{**}$
Interest in learning about Arduino (X5)					1.0	$.53^{**}$
Knowledge about Arduino (X6)						1.0

^{NS}Not significant ($p > .05$). $^{*}p < .05$. $^{**}p < .01$. $^{***}p < .001$.Conclusions and Discussion

Conclusions and Recommendations

Both Design 12 and Design 2 produced similar results; a short-duration instructional treatment had significant, positive, and large effects on increasing novice college of agriculture students' overall Arduino knowledge and breadboarding self-efficacy, but no significant effects were observed on interest in Arduino or programming self-efficacy. Programming self-efficacy scores (23.5%) were lower than breadboarding rubric scores (58.5%). Based on these results, it might be assumed that increased self-efficacy in breadboarding could be a result of students' greater mastery and vicarious experiences for breadboarding. Alternatively, the lack of an increase in programming self-efficacy could be due to students' lower mastery and vicarious experiences for programming. This is consistent with theory (Bandura, 1986) and research (Erdil, 2019; Lee et al., 2014) on academic self-efficacy.

The significant, positive intercorrelations between the students' breadboarding and programming rubric scores and the interest, self-efficacy, and knowledge support previous research by (Thomas et al., 2011; Erdil, 2019; Lee et al., 2014; McLaughlin et al., 2005) and would suggest that further research should question how these interactions impact student learning. For example, programming (r^2 = .24) and breadboarding (r^2 = .27) rubric scores explained less than 30% of the variance in self-efficacy in each of the corresponding areas. Knowing that all participants of this study indicated that they were novice Arduino and programming users, what other extrinsic or intrinsic factors could be related to self-efficacy of the dependent variables? Breadboarding (r^2 = .59) and programming (r^2 = .53) self-efficacy were especially insightful at predicting students' overall interest in learning about Arduino.

Based on the rubric scores of this study it seems that most students were not adequately prepared to successfully breadboard and program a simple Arduino mastery activity; when given a short 30-minute introduction to programming and breadboarding video lesson, a reference sheet, and all the necessary components to complete the task. This was especially true for the programming part of the activity thus supporting Lee et al. (2011) and Thomas et al. (2011). Therefore, the instructional treatment should be lengthened in time and restructured to include increased instructional scaffolding (Wood et al., 1976), like simplified breadboarding and programming tasks, an example of a partially constructed program, and the use of cooperative learning, or other methods. The level of difficulty of this instruction should be decreased for novice programmers to allow the brain deeper processing and repetition of new knowledge, so that students can organize this new knowledge in a way that allows for them to easily recall the skills when needed (McLaughlin et al., 2005). Future research is needed to determine if a lengthened and redesigned instructional treatment would have an effect in successes of breadboarding and programming Arduino activities or an increase in interest in Arduino or self-efficacy on breadboarding and especially in programming as suggested by both Bandura's (1986) theory and by Erdil (2019) & Lee et al., (2014) previous research.

References

Al-Abad, M. (2017). A review of embedded systems education in the Arduino age: lessons learned and future directions. *International Journal of Engineering Pedagogy, 7*(2), 79-93. https://doi.org/10.3991/ijepv7i2.6845

Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Prentice-Hall.

Bandura, A. (2006). Guide for constructing self-efficacy scales. In F. Pajares & T. Urdan (Eds.), *Self-efficacy beliefs of adolescents* (pp. 307-337). Information Age Publishing.

Bruner J. S. (1978). The role of dialogue in language acquisition. In A. Sinclair, R. J. Jarvella & W. J. Levell (eds). *The child's conception of language*, Springer-Verlag.

Campbell, D. T., & Stanley, J. C. (1963). *Experimental and quasi-experimental designs for research*. Houghton Mifflin Company.

Cohen, J. (1960). A coefficient for agreement of nominal scales. *Journal of Educational and Psychological Measurement, 20*(1), 37-46. https://doi.org/10.1177/001316446002000104

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nded.). Lawrence Erlbaum Associates, Publishers.

Davis, J. A. (1971), *Elementary survey analysis*. Prentice-Hall.

Dennis, C., Aguilera, J. M., & Satin, M. (2009). Technologies shaping the future. In C. A. da Silva, D. Baker, A. W. Shepherd, and C. Jenane (Eds.), Agro-industries for development (pp. 92-135). Cambridge, MA: CAB International and The Food and Agriculture Organization of the United Nations.

Doolittle, P. E., & Camp, W. G. (1999). Constructivism: The Career and Technical Education Perspective. *Journal of Vocational and Technical Education*, 16(1).

Erdil, N. O. (2019, June), *Motivating students for learning using scaffolding and a variety of assignments and activities* [Paper presentation]. ASEE Annual Conference & Exposition, Tampa, FL, United States. https://peer.asee.org/33121

Field, A., & Miles, J. (2012). *Discovering statistics using SAS*. Sage.

Gable, R. K. & Roberts, A. D. (1983). An instrument to measure attitude toward school subjects. *Educational and Psychological Measurement, 43*(1), 289-293. https://doi.org/10.1177/001316448304300139

Goap, A., Sharma, D., Shukla, A. K., & Krishna, C. R. (2018). An IoT based smart irrigation management system using machine learning and open source technologies. *Computers and Electronics in Agriculture*, *155*(2018), 41-49. https://doi.org/10.1016/j.compag.2018.09.040

Goering, C. E., Stone, M. L., Smith, D. W., & Turnquist, P. K. (2003). *Off-road vehicle engineering principles*. American Society of Agricultural Engineers.

Keim, R. (2019). *What is a microcontroller? The defining characteristics and architecture of a common component*. https://www.allaboutcircuits.com/technical-articles/what-is-a-microcontroller-introduction-component-characteristics-component/

King, L., McKim, A., Raven, M., & Pauley, C. (2019). New and emerging technologies: Teacher needs, adoption, methods, and student engagement. *Journal of Agricultural Education*, *60*(3). https://doi.org/10.5032/jae.2019.03277

Kittur, J. (2020). Measuring the programming self-efficacy of electrical and electronics engineering students. *IEEE Transactions on Education*, *63*(3), 216-223.

Lee, W., Lee, M-J, & Bong, M. (2014). Testing interest and self-efficacy as predictors of academic self-regulation and achievement. *Contemporary Educational Psychology*, *39*(2014), 86-99. http://dx.doi.org/10.1016/j.cedpsych.2014.02.002

McLaughlin, M., McGrath, D. J., Burian-Fitzgerald, M. A., Lanahan, L., Scotchmer, M., Enyeart, C., & Salganik, L. (2005, April 12). Student Content Engagement as a Construct for the Measurement of Effective Classroom Instruction and Teacher Knowledge. Washington, DC; American Institutes for Research.

National Research Council. (2009). *Transforming Agricultural Education for a changing world*. National Academies Press.

Paek, I. (2015). An investigation of the impact of guessing on coefficient α and reliability. *Applied Psychological Measurements*, *39*(4), 264-277. doi: 10.1177/0146621614559516

Schuman, A. W. (2010). Precise placement and variable rate fertilizer technologies for horticultural crops. *HortTechnology*, *20*(1), 34-40.

Suprem, A., Mahalik, N., & Kim, K. (2013). A review of application of technology systems, standards and interfaces for agriculture and food sector. *Computer Standards and Interfaces, 35*(2013), 355-364. http://dx.doi.org/10.1016/j.csi.2012.09.002

Thomas, M. K., Ge, X., & Greene, B. A. (2011). Fostering 21st Century skill development by engaging students in authentic game design projects in a high school computer programming class. *Journal of Educational Computing Research*, *44*(4), 391–408. https://doi.org/10.2190/EC.44.4.b

Wood, D., Bruner, J. S., Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology, 17*(1976), 89-100.

Chapter 3: Novice Arduino Users in an Introduction to Agriculture Systems Technology Course: Common Breadboarding and Programming Errors

Introduction

In recent years, agriculture has begun the adoption of new technologies in an effort to improve efficiency and automate processes for sustaining the agricultural demands of the world (Ji-Chun, 2010). One of the new uses of technology currently being integrated into agriculture is microcontrollers, which are used as embedded computing systems in technological practices (Goering et al., 2003; Goap et al., 2018; Suprem et al., 2013; Schumann, 2010). In these practices, microcontrollers serve as circuit devices that can receive inputs and control outputs for improved systems (Keim, 2019). Specifically, Arduino, a type of microprocessor, is one that is a programmable, open-source microcontroller and software program that is widely accepted and used in various industries, ages, and skill levels (Al-Abad, 2017). This type of system is relatively user-friendly when considering the types of microprocessors. Nevertheless, there are some common errors among novice users in learning the breadboarding and programming capabilities of Arduino systems.

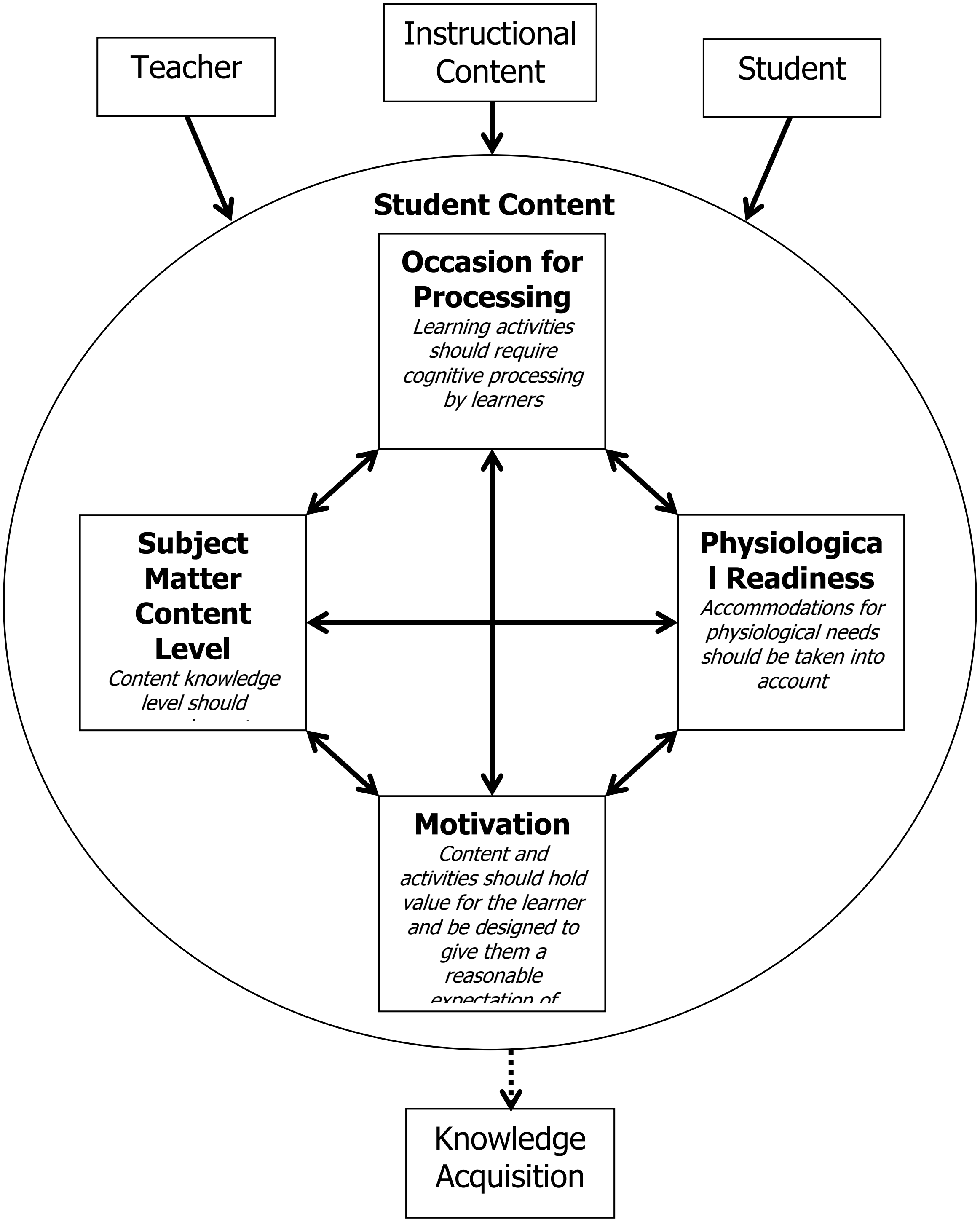
There are few studies that evaluate the struggles of learning programming and electronics, especially in agriculture. Some studies like one from DesPortes (2019) examined the barriers of the learning process in a qualitative study in order to offer new insights on this topic. In a world increasingly impacted by technology, agricultural careers are no exception to the demand for improved technology education. However, when the needs of students are not understood, curricula involving programming and electronics can fail students in preparing them for a quickly evolving industry (National Research Council, 2009). The following study is a quantitative analysis which evaluated the common mistakes that novice programmers made after a brief introductory lesson in an effort to provide literature for the improvement of Arduino programming curricula for novice agriculture programmers.

Theoretical Framework

This study was guided by McLaughlin's et al. (2005) theory of student content engagement as a measure of effective classroom instruction. The theory can be divided into four categories: subject matter content level, occasion for processing, physiological readiness, and motivation (shown below in Figure 3). These four categories will vary on impact between each other. Some may have tremendous benefits to instruction while also creating negative effects in another area of engagement. The ability to design, process and complete the content within a given instructional environment will vary widely among students and teachers. Student Content Engagement was used as a means of choosing a subject matter that is both interesting and important for agriculture students to learn. Furthermore, it is important to connect subject matter with instruction through means of student interaction by an embedded activity.

Figure 3

Conceptual model of student content engagement



Note. Estepp, C. M., & Roberts, T. G. (2010). Using student content engagement to improve college teaching [Abstract]. NACTA Journal, 54(supplement 1), 61.

Subject matter content level states that students learn based on their ability to question previous knowledge already owned (McLaughlin et al., 2005). Subject matter must be able to trigger a connection to past student experiences and knowledge in order for learning to take place. It is common for students to challenge ideas based on how they view the world, leading to the development of new knowledge. According to McLaughlin et al. (2005), “what is new material today is prior knowledge tomorrow” (p. 8). Effective instruction should be designed to be slightly above students’ prior knowledge and include multiple opportunities to make connections with new knowledge and skills. This study takes students’ basic understanding of agriculture and leads them to microcontrollers that often go beneath the surface of everyday skills such as machinery and controls.

According to McLaughlin et al., (2005), occasion for processing is how the brain receives, stores, and takes in new and old information from the student’s learning environment. It is safe to assume that the amount of information the brain can intake at a given time is limited from person to person. The flow of information is a two-way street between our sense and memory, controlled by the brain. Learning is the process of developing these links between memory, senses, and the processing that is done by our brain. Deep processing has been found to be the most impactful time for learning to occur by means of categorization, elaboration, mnemonics (for retrieval and coding for retrieval) etc. The main difference in learning comes from the depth of processing that occurs. This processing can come during “down” time when the brain has time to go back and review what has been learned or by multiple opportunities to receive information over a length of time, usually two to three days to develop associations. Thus, the use of an instructional lesson on day one followed by a related instructional activity two days later to provide ample time for the brain to “deep” process a new skill before having the chance to refer back to this stored knowledge (McLaughlin et al., 2005).

Physiological readiness according to McLaughlin’s et al. (2005) refers to a student’s ability to learn at any given moment in time. This readiness is affected by two key factors: attention and stress. Students must be attentive in order for learning to take place. Attention affects how much working memory is available to the student in order to be used for the development of new knowledge. The second factor, stress, can be both good or bad stress, and must remain at an optimum level in order for learning to occur. Disabilities, ADHD, nutrition, and sleep are all factors that can have an impact on student learning, leading one to design instruction to meet the needs of all types of students. Giving students two opportunities to learn new knowledge by instruction and activity is an excellent way to maintain the optimum level of stress by equally providing them with the tools to be successful as well as giving them enough of a challenge to create some stress (McLaughlin et al., 2005).

Based on McLaughlin’s et al. (2005) theory, motivation is what drives a student to engage in a specific learning opportunity. Each student brings their own influences on motivation, both intrinsic and extrinsic factors, that will have good and bad effect. While students bring a constant level of motivation to the classroom, motivation is driven by the instructional activity of the moment. A student’s attitude towards participation is a direct reflection of a student’s expectancy of success towards a specific task, or the level of challenge will affect a student’s motivation whether to continue learning or not. The Arduino breadboarding and programming activity was designed to appeal to students’ overall interest, while engaging them in a task-based learning activity to develop simple electric circuitry and programming skills (McLaughlin et al., 2005)

Purpose and Objectives

The purpose of this study was to understand where novice agriculture programming students frequently faced challenges, or errors in programming and breadboarding simple circuits. The objective of this study was to identify common mistakes made by novice students when breadboarding and programming an Arduino task based on student rubric scores.

Methodology

Students enrolled in the University Of Arkansas College Of Agricultural, Food, and Life Sciences undergraduate students served as the population for this study. The accessible sample consisted of all students (*n* = 25) enrolled in the fall 2021 semester fundamentals of agricultural systems technology course who completed the hands-on activity during the given time and were graded according to the rubric. This study was part of another larger study, which focused on students' self-efficacy, knowledge, and interest on Arduino.

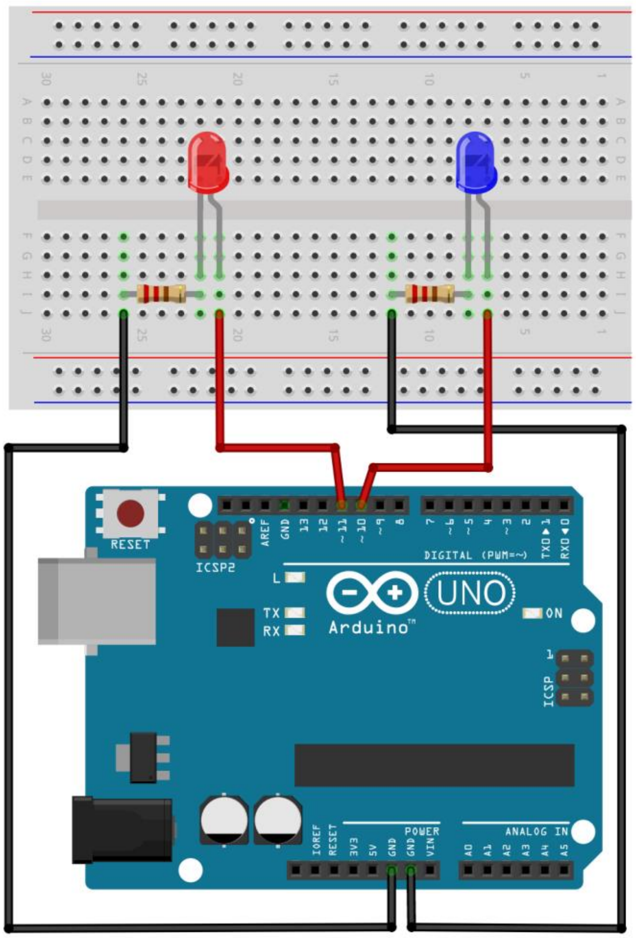
The instrument developed for this study was a 27 point rubric to evaluate students' breadboarding and programming skills on the hands-on Arduino breadboarding and programming activity. The rubric consisted of three major sections; 10-points for breadboarding, 15 points for programing, and two points for the correspondence between breadboard circuitry and programming. The first 10 points of the rubric included items such as "circuit is wired to a digital pin" and "resistor is connected in series with LED". The second part of the rubric consisted of 15 points from the programming portion of the activity, one point was assigned to each line of the programming sketch such as "digitalWrite for Blue LED 500ms". The last section of the rubric consisted of two points that determined if the sketch corresponded with the breadboarding. The researcher graded each activity using this rubric; while a second researcher chose five randomly completed activities and graded them using the same rubric. The Cohen's kappa coefficient values for the rubric were .87 for breadboarding and 1.0 for programming, indicating near perfect and perfect agreement (Cohen, 1960).

Arduino UNO R3 programmable microcontrollers and breadboards were used for this activity. Students were given an introduction to programming and breadboarding virtual lesson explaining basic programming language and simple electronics circuit building. After this 30-minute virtual instruction, students met again two days later in a computer lab where they were given an Arduino UNO R3 microcontroller & breadboard, two LED lights, one red and one blue, two current limiting resistors, more than enough solderless breadboard jumper wires, and a programming cable with access to a computer with the Arduino IDE programming software. An activity sheet asked students to breadboard and program two simple LED circuits with differing colors to blink on and off in a specified sequence. An Arduino reference sheet was provided with an example of a single red LED circuit and generic Arduino programming format statements. Students were not allowed to use any outside resources, including the internet or neighbors, to help them complete the task. Students were given 50-minutes to complete this activity and students were spaced every other chair to prevent cheating.

Shown in Figure 4 is the simplest way to breadboard the hands-on Arduino activity, although this was not the only way students could have successfully completed the task. Upon completion of the breadboarding and programming activity, students were able to plug in their microcontrollers to test if they correctly completed the task. At the end of the 40 minute lab session microcontrollers and breadboards were collected and names were placed on them. A rubric was developed to score each student’s breadboarding and programming abilities. The rubric was developed based on the breadboarding and programming lines that each of the two LED circuits that were required. Such items evaluated if the “circuit is wired to a digital pin”, “resistor is connected in series with LED”, and “circuit is wired to ground pin” with these items being worth 1 point each (either they did or did not complete the line item,) no partial credit was given.

Figure 4

A simple way to build the Arduino LED light activity



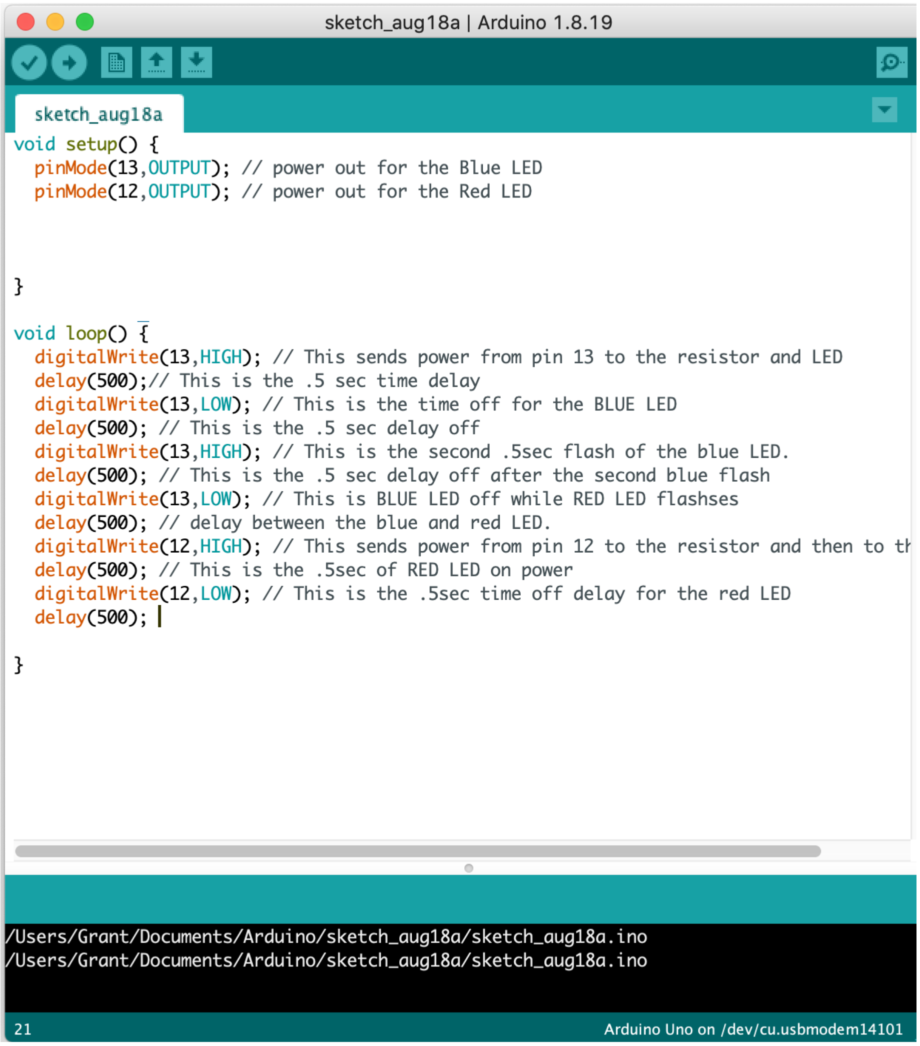
This simple breadboarding activity required students to connect the digital pin of the Arduino to the breadboard using jumper wires. One of the most important factors in building a successful circuit is that the LED must be wired in a forward-bias way. This means that the output from the digital pin needs to connect to the anode (+) leg of the LED and the ground pin needs to be attached to the cathode (-) of the LED. This was clearly shown in the reference sheet.

The resistor could be wired to either side of the LED, as long as it was wired in series into each circuit.

The programming section of the rubric evaluated the students' abilities to correctly write the program. Each line of the program or "sketch" specified a different command to the Arduino. Students first had to declare an output for each LED circuit in the "setup" function of Arduino IDE, choosing any of the numbered digital or analog pins, for this activity the digital pin needed to be chosen. In the "loop" section of the sketch, students specified either HIGH or LOW voltage to a specified pin number and a delay written in milliseconds. Figure 5 shows an example of a completed program sketch that students should have written in order to complete make the LED circuits function properly.

Figure 5

Screenshot of Completed Arduino Activity



The final part of the rubric determined the ability of the students to reference the chosen pin in the sketch. Breadboarding requires students to choose any of the 14 digital pins on the circuit board to initiate each of the LED circuits. Since students were allowed to choose which output pin of the Arduino they used, students were also required to reference in the sketch this same pin. For example, if a student chose to wire a circuit to pin 12 on the Arduino, in the sketch in order for the circuit to work, it must first be identified in the setup part of IDE by writing "pinMode(13,OUTPUT);" declaring that digital pin 13 has been identified to send out power.

Data Analysis

Frequency distributions of students' rubric scores were categorized into breadboarding and programming mean scores. To identify where students commonly made errors the scores were further broken down into three sections; breadboarding, programming, and compatibility between breadboarding and programming.

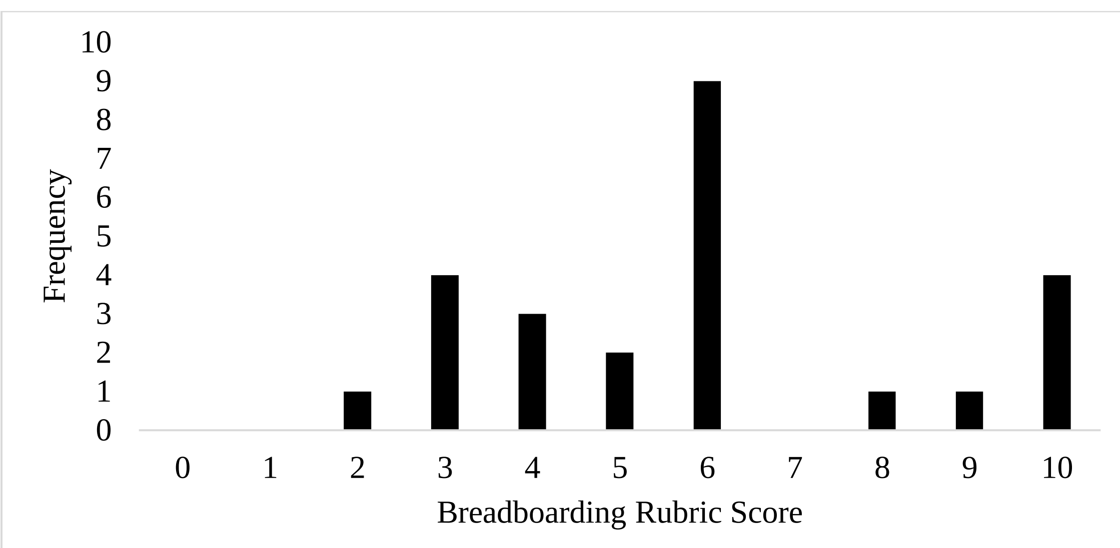
Results

Arduino microcontroller and breadboards were collected at the conclusion of the 40 minute class session, regardless of if the student was finished with the activity or not. Students scored a mean of 36.1% (*SD* = 24.8%) on the overall rubric encompassing the breadboarding, programming, and compatibility of the activity. It is worth mentioning that only three students (11.5%) were able to demonstrate correctly functioning circuits by the end of the session.

Students scored a mean of 58.4% (*SD* = 25.0%) on the breadboarding portion of the rubric. As shown in Figure 6 below, the frequency of students' breadboarding rubric scores ranged from a two to a ten, with a score of six being the most common.

Figure 6

Breadboarding Rubric Scores Out of 10 Points Possible



Of these students, 66.7% correctly originated both LED circuits from digital pins of the Arduino. Furthermore, 55.6% correctly terminated both the red and blue LED circuit back to GND pins. Also, 59.3% correctly wired resistors in series with both LED circuits. However, only 22.2% connected both LED anodes to the positive side of the circuit and only 25.9% connected the cathodes to the negative side of the circuit. Table 4 shown below, shows two of the most common errors observed in the blue LED circuit were; 78.0% (n = 18) failed to connect a GND pin to the cathode (-) of the blue LED and 78.0% (n = 17) did not connect the anode (+) side of the LED to a digital pin on the Arduino using a jumper wire.

Table 4

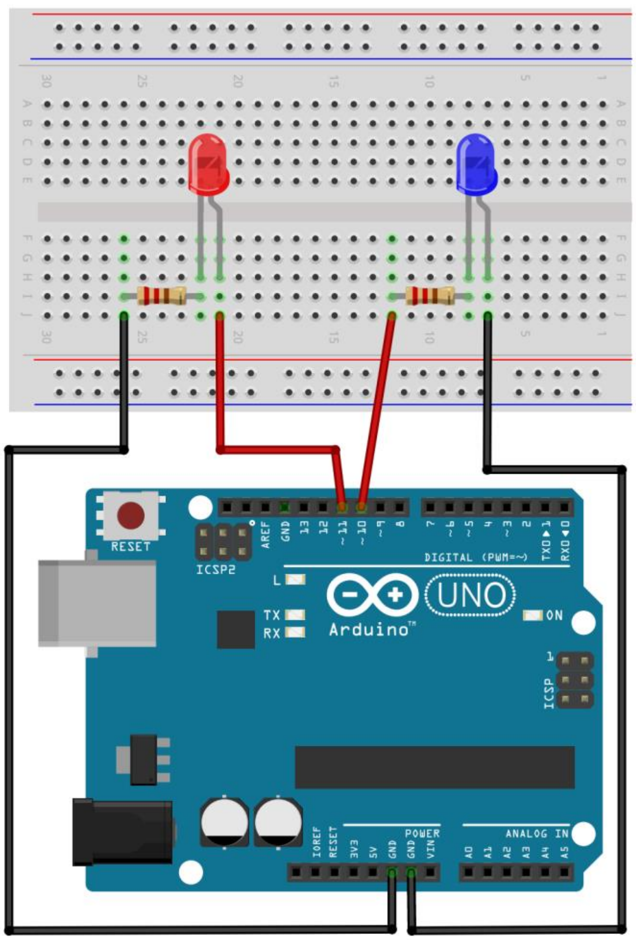
Percentage of Students' Errors Breadboarding Blue and Red LED Circuits

Rubric Criteria	*n*	Percent Error (%)
Blue LED Circuit		
Circuit is wired to a digital pin	8	32.0
DigitalPin connects to anode (+) of LED	18	72.0
GND pin connects to cathode (-) of LED	17	78.0
Resistor is connected in series with LED	11	44.0
Circuit is wired to GND pin	11	44.0
Red LED Circuit		
Circuit is wired to a digital pin	3	12.0
DigitalPin connects to anode (+) of LED	15	60.0
GND pin connects to cathode (-) of LED	15	60.0
Resistor is connected in series with LED	4	16.0
Circuit is wired to GND pin	2	8.0

The first two common errors from table 4 are shown below in figure 5. Notice how the GND pin is connected the anode (+) side of the blue LED (indicated by the bend in the longer wire of the LED) of the circuit with the black jumper wire on the Arduino, thus the current does not flow in a forward-bias direction from a digital pin output through the LED circuit and back to ground. The resistor can be connected in series on either side of the LED circuit, this one happened to be on the anode (+) side of the LED circuit.

Figure 7

Incorrectly Breadboarded Blue LED Circuit.



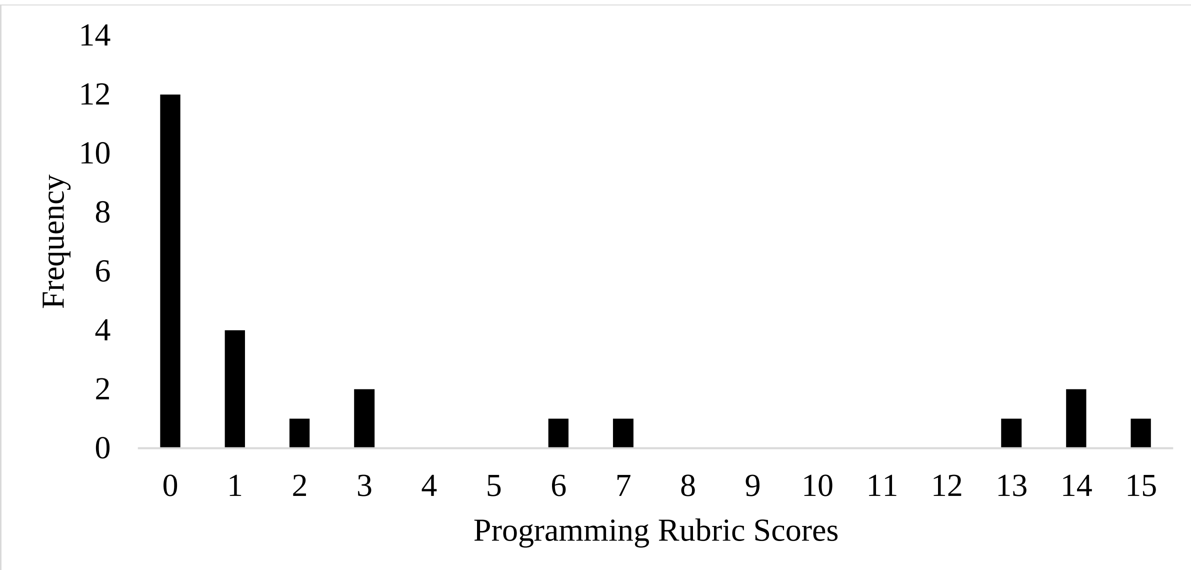
Note. You can see that the red wire of the blue LED circuit connects to the cathode (-) of the LED and to a digital pin of the Arduino.

The two least common errors occurred in the red LED circuit; the first of which indicated that only 8.0% (n = 2) of students were unable to properly ground the red LED circuit, and 12.0% (n = 3) students did not wire the red LED circuit to a digital pin. Instead, they chose to terminate circuits at analog pins or other numbered pins of the Arduino and did not complete the circuit.

Students scored a mean of 21.6% (*SD* = 34.0%) on the programming or "sketch" portion of the rubric. As shown in Figure 8, programming rubric scores ranged from zero to 15, with zero being the most common score.

Figure 8

Programming Rubric Scores Out of 15 Points Possible



In the program, only 22.2% of students declared both digital pins as outputs in the setup function of IDE and less than one in five wrote the correct program statements to cause the LED circuits to turn on (18.5%), to turn off (18.5%), or to have the correct delay between events (18.5%). Table 5 shows that students' most frequent errors when programming was that 88% (*n* = 22) of the participants were not writing comments at the end of each line of the program to describe what that command represents. Three other common errors were: 84% (*n* = 21) of students were not writing commands in the loop part of the sketch such as "DigitalWrite for BLUE LED HIGH," 84.0% (*n* = 21) "Delay for Red LED 500ms" and 84.0% (*n* = 21)

"DigitalWrite for BLUE LED LOW." Three of the least common lines of the program for errors to be found were the very first line in the loop; "DigitalWrite for BLUE LED HIGH" and the "DigitalWrite for RED LED HIGH" both representing 72.0% (*n* = 18) students' errors.

Table 5

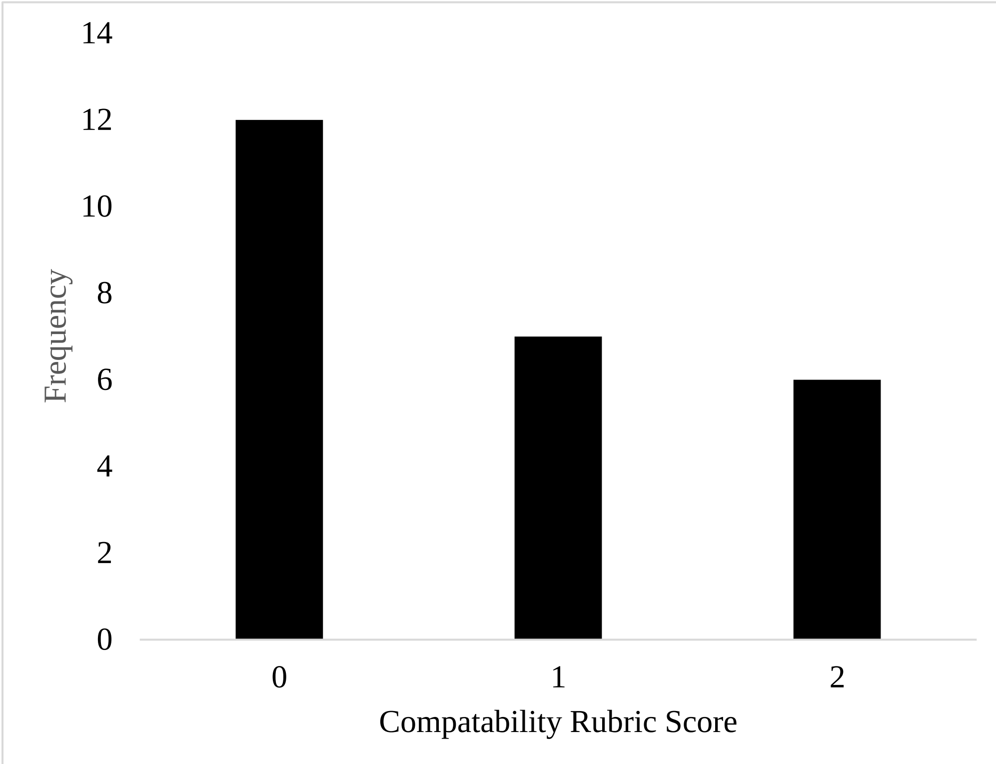
Percentage of Students' Errors Programming

Statement	*n*	Percent Error (%)
Void Setup		
pinMode for Blue LED pin_OUTPUT	19	76.0
pinMode for RED LED pin_OUTPUT	16	64.0
Void Loop		
DigitalWrite for BLUE LED HIGH	18	72.0
Delay for Blue LED 500ms	20	80.0
DigitalWrite for BLUE LED LOW	19	76.0
Delay for Blue LED 500ms	20	80.0
DigitalWrite for BLUE LED HIGH	21	84.0
Delay for Blue LED 500ms	20	80.0
DigitalWrite for BLUE LED LOW	21	84.0
Delay for Blue LED 500ms	20	80.0
DigitalWrite for RED LED HIGH	18	72.0
Delay for Red LED 500ms	19	76.0
DigitalWrite for RED LED LOW	20	80.0
Delay for Red LED 500ms	21	84.0
Comments after each line to describe function	22	88.0

Completed projects were also evaluated to determine the compatibility between physical breadboarding and the program. Shown below in Figure 9, almost half of all students scored a zero on the compatibility section of the rubric.

Figure 9

Breadboard and Program Compatibility Results Out of Two Points Possible



As for the students' ability to correctly reference the correct pin in the sketch matched with the pin that was chosen breadboarding, only 25.9% of them were found to be compatible. Shown below in Table 6 shows that it was found that 72.0% (*n* = 18) of students did not identify the correct pin number in their sketch for the blue LED, while the red LED circuit was more likely to be correctly done with less errors throughout the activity. As for the red LED circuit, 52.0% of students were still unable to reference the correct pin in the sketch.

Table 6

Program to Breadboarded Digital Output Pin

Statement	*n*	Percent Error (%)
Blue LED pinMode ____, Output corresponds to the breadboard pin chosen	18	72.0
Red LED pinMode ____, Output corresponds to the breadboard pin chosen	13	52.0

Out of 25 students, only three were successful at completing the activity and making the LED circuits blink correctly. These three students were the only ones to write descriptive comments after each line of programming stating what that line of programming language represented. Many of the pin numbers referenced in the program were found to be randomly selected and did not relate to what the students actually breadboarded. Some even mixed the two different colored circuits up with each other, while others failed to reference any digital pin at all as an output and left this section of the program blank.

Conclusion and Recommendations

Based on these errors, students were more likely to breadboard the red LED circuit correctly rather than the blue. This included connecting jumper wires to the correct GND and Digital pins of the Arduino, but not necessarily to a correctly forward bias the LED in circuit with the resistor wired in series. The reference sheet did clearly show an example of a single red LED circuit properly breadboarded. In accordance with the example red LED circuit, students were also more likely to declare a digital pin as an output in the sketch for the red rather than the blue circuit. Students were able to successfully breadboard circuits to digital pins, but they were unable to correctly reference this pin, or any setup or loop commands in the setup function of the Arduino IDE. In accordance with McLaughlin’s et al. (2005) and Doolittle & Camp, 1999) students needed more opportunities to construct knowledge bridges between breadboarding and programming, and more time to process this new skill.

Given the abbreviated nature of the instructional video (30-minutes) and activity (40-minutes), novice agriculture students performed marginally well on the breadboarding task, and poorly on the programming task. Consistent with McLaughlin et al. (2005) Theory of Student Content Engagement, the results of this study suggest that students need more instructional time on breadboarding simple electronics circuits, specifically focusing on curriculum that effectively challenges students' prior knowledge on circuits and builds new knowledge on basic electrical circuitry (grounding and current flow). In addition, more instruction is needed on the conceptual logic and practical aspects of writing in Arduino's open source programming language, which is similar to C++. Students may benefit from a guided instructional activity where they write programming alongside of the instructor.

Many of the errors observed while evaluating the projects were that students simply were unable to distinguish which pins on the Arduino are digital and which are analog. It seemed that students were unable to effectively recall information from the first class session that took place two days before the activity.

The instructional video should be redesigned to incorporate more opportunities to connect with a student's prior knowledge and allow more opportunities for deeper processing by means of categorization, elaboration or rehearsal during the breadboarding and programming activity, repetitive measures should be commonly used for developing a better understanding of the programming language according to the theory of student content engagement (McLaughlin et al., 2005). It is also noteworthy that students need more instruction on the relationship between the breadboarding and programming, the curriculum could be redesigned to slowly introduce students to circuitry and allow for learning to take place, and after mastery of basic electronic circuits programming then could be introduced as a subject to bridge students' past and future knowledge (McLaughlin et al., 2005).

References

Al-Abad, M. (2017). A review of embedded systems education in the Arduino age: lessons learned and future directions. International Journal of Engineering Pedagogy, 7(2), 79-93. https://doi.org/10.3991/ijepv7i2.6845

Des Portes, K., & DiSalvo, B. (2019, July). Trials and Tribulations of Novices Working with the Arduino. In Proceedings of the 2019 ACM Conference on International Computing Education Research (pp. 219-227).

Estepp, C. M., & Roberts, T. G. (2010). Using student content engagement to improve college teaching [Abstract]. *NACTA Journal, 54*(supplement 1), 61.

Goap, A., Sharma, D., Shukla, A. K., & Krishna, C. R. (2018). An IoT based smart irrigation management system using machine learning and open source technologies. Computers and Electronics in Agriculture, 155(2018), 41-49. https://doi.org/10.1016/j.compag.2018.09.040

Goering, C. E., Stone, M. L., Smith, D. W., & Turnquist, P. K. (2003). Off-road vehicle engineering principles. American Society of Agricultural Engineers.

Ji-Chun Zhao, Jun-feng Zhang, Yu Feng and Jian-xin Guo, "The study and application of the IOT technology in agriculture," *2010 3rd International Conference on Computer Science and Information Technology*, 2010, pp. 462-465, doi: 10.1109/ICCSIT.2010.5565120.

Keim, R. (2019). What is a microcontroller? The defining characteristics and architecture of a common component. https://www.allaboutcircuits.com/technical-articles/what-is-a-microcontroller-introduction-component-characteristics-component/

King, L., McKim, A., Raven, M., & Pauley, C. (2019). New and emerging technologies: Teacher needs, adoption, methods, and student engagement. *Journal of Agricultural Education*, *60*(3). https://doi.org/10.5032/jae.2019.03277

McLaughlin, M., McGrath, D. J., Burian-Fitzgerald, M. A., Lanahan, L., Scotchmer, M., Enyeart, C., & Salganik, L. (2005, April 12). Student Content Engagement as a Construct for the Measurement of Effective Classroom Instruction and Teacher Knowledge. Washington, DC; American Institutes for Research.

National Research Council. (2009). *Transforming Agricultural Education for a changing world*. National Academies Press.

Schuman, A. W. (2010). Precise placement and variable rate fertilizer technologies for horticultural crops. HortTechnology, 20(1), 34-40.

Suprem, A., Mahalik, N., & Kim, K. (2013). A review of application of technology systems, standards and interfaces for agriculture and food sector. Computer Standards and Interfaces, 35(2013), 355-364. http://dx.doi.org/10.1016/j.csi.2012.09.002

Chapter Four: Conclusions

The two articles of this thesis examined the use of Arduino UNO R3 programmable microcontrollers in a fundamentals of agricultural systems technology course at a university in the fall of 2021. The participants in this study took a pretest and posttest over Arduino interest, knowledge, and breadboarding and programming self-efficacy. An instructional treatment was designed and implemented over the course of one week of instruction. Students were first introduced to the subject of programming and breadboarding and how it relates to agriculture through the use of a 30-minute instructional video, which taught students the basics of programing and breadboarding. The second session of class gave them an opportunity to put skills to use by breadboarding and programming a simple red and blue LED circuit. Finally, on the third day, students met and were given graded rubrics from the activity and self-efficacy and knowledge were measured again with the posttest.

The first article results show that students had significantly ($p < .05$) higher breadboarding self-efficacy and Arduino knowledge scores after the instructional treatment, while there were no significant ($p > .05$) differences in interest in Arduino or programming self-efficacy after the instruction. There were significant ($p < .05$) positive correlations between breadboarding and programming rubric scores and breadboarding self-efficacy, programming self-efficacy, interest in learning about Arduino, and Arduino knowledge scores. This was consistent with Bandura’s (1986) self-efficacy theory. Higher student performance on the breadboarding and programming tasks were associated with higher levels of self-efficacy, knowledge, and interest.

In the second article students scored a mean of 58.5% on breadboarding compared to a mean of 23.5% on Arduino programming. The most common student errors were failing to correctly “forward-bias” the LED in a circuit, failing to correctly initiate a LED circuit from a digital pin, and failing to terminate the LED circuit at a GND pin. It is worth mentioning that students struggled to create the blue LED circuit in entirety likely because the activity reference sheet only provided an example of a single correctly breadboarded circuit. Students appeared largely unable to conceptualize how a second circuit should be added to the breadboard. Other common errors were made in the programming portion of the rubric, such as only 22.2% of students being able to declare digital pins as outputs. Less than one in five students correctly programmed circuits to blink the LED’s on and off with the correct time delays. More than 75% of students were unable to reference the correct digital pin in the Arduino IDE program as to the pin that was chosen when breadboarding circuits. The three students who successfully breadboarded and programmed the LED activity were the only students to include comments after each line of programming. This may indicate that students who failed both to write correct programs and to write comments for each line of program did not understand how to correctly execute the programming task.

The results of this study have implications for teaching Arduino programming and breadboarding to novice college of agriculture students. The increase in breadboarding self-efficacy was likely a result of students’ greater mastery and vicarious experiences in breadboarding; conversely, we did not see the same increase programming self-efficacy, likely due to students’ lesser mastery and vicarious experiences. part of the activity. Thus, the instructional treatment should be redesigned to include more time for guided programming and the repetition of breadboarding and programming concepts (Wood et al., 1976). It might be useful to simplify breadboarding and programming tasks and show examples of completed circuits and programs. Further research is needed to determine if a lengthened and redesigned instructional treatment would create increased success in breadboarding and programming simple Arduino projects as suggested by Bandura's theory (1986), Erdil (2019) and Lee et al., (2014). This is also consistent with McLaughlin's et al. (2005) theory of student content engagement.

In accordance with McLaughlin's et al. (2005), theory of student content engagement, this study provided ample challenges for students to question prior knowledge by learning Arduino programming and breadboarding. Based on the rubric scores, one might conclude that the content level was slightly too challenging for students. Specifically, students need more instruction on how to "forward-bias" LED's in circuits and how to declare digital pins as outputs in the Arduino IDE program. Special emphasis should be placed on the relationship between the microcontroller input/output pins to the programming language, and how the breadboard is used to create the circuit. The results from both of these studies will be incorporated into a redesigned instructional treatment and tested to determine if a longer, more in-depth treatment will increase students' success in learning basic Arduino breadboarding and programming.

References

Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Prentice-Hall.

Erdil, N. O. (2019, June), *Motivating students for learning using scaffolding and a variety of assignments and activities* [Paper presentation]. ASEE Annual Conference & Exposition, Tampa, FL, United States. https://peer.asee.org/33121

Lee, W., Lee, M-J, & Bong, M. (2014). Testing interest and self-efficacy as predictors of academic self-regulation and achievement. *Contemporary Educational Psychology, 39*(2014), 86-99. http://dx.doi.org/10.1016/j.cedpsych.2014.02.002

McLaughlin, M., McGrath, D. J., Salganik, laura, Enyeart, C., Scotchmer, M., Lanahan, L., & Burian-Fitzgerald, M. A. (2005). *Student Content Engagement as a Construct for the Measurement of Effective Classroom Instruction and Teacher Knowledge*. American Institutes for Research.

Appendices

Appendix A

UNIVERSITY OF ARKANSAS

To:	Donald M Johnson AGRI 00205
From:	Justin R Chimka, Chair IRB Expedited Review
Date:	07/20/2021
Action:	**Exemption Granted**
Action Date:	07/20/2021
Protocol #:	2107344251
Study Title:	Effects of an Arduino Lesson and Activity on Novice Programmers' Self-Efficacy, Interest, and Knowledge

The above-referenced protocol has been determined to be exempt.

If you wish to make any modifications in the approved protocol that may affect the level of risk to your participants, you must seek approval prior to implementing those changes. All modifications must provide sufficient detail to assess the impact of the change.

If you have any questions or need any assistance from the IRB, please contact the IRB Coordinator at 109 MLKG Building, 5-2208, or irb@uark.edu.

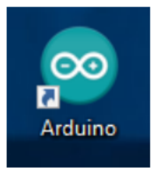
Appendix B

In-Class Activity: Breadboarding Circuits and Writing an Arduino UNO Sketch

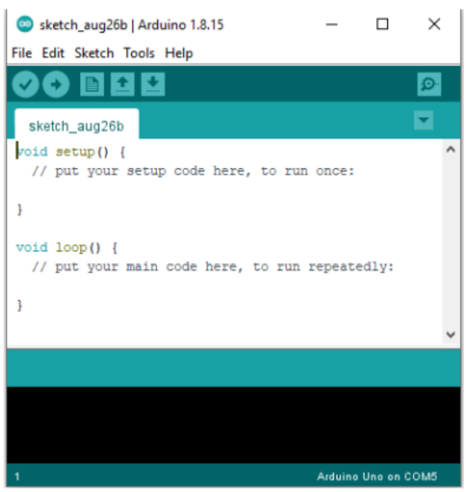
For this task, you are provided with an Arduino UNO, the Arduino IDI program, a USB 2 cable, a breadboard, connector wires, two 240*Ω* resistors, and one blue and one red light-emitting diode (LED).

Complete the following:

1. Use these components to "breadboard" two resistor/LED circuits. Each circuit should operate off of a different Arduino digital pin. One circuit should operate the Blue LED and the other circuit should operate the Red LED.
2. Once you have "breadboarded" your two circuits, connect the Arduino UNO to the computer using the Type 2 USB cable, and then open the Arduino IDE. The Arduino IDE icon looks like this:



3. The Arduino IDE should open a new sketch that will look like this:



If this is not what you see, open a new sketch by clicking "File" on the menu bar and selecting "New."

4. Write an Arduino sketch (program) to cause the following to occur:

   The Blue LED should blink twice (ON, OFF, ON, OFF), then the Red LED should blink once (ON, OFF), with 0.5 seconds between each event. This sequence of blinks should repeat as long as your Arduino sketch operates. This sequence is shown below:

BLUE (ON, 0.5 sec.) → BLUE (OFF, 0.5 sec.) → BLUE (ON, 0.5 sec.) → BLUE (OFF, 0.5 sec.) → RED (ON, 0.5 sec.) → RED (OFF, 0.5 sec.) → REPEAT

5. Document the <u>purpose</u> of <u>each line</u> of your program using "comments" in your program.

6. Verify/Compile your sketch.

   A. If there are errors, debug your sketch and repeat step 6.

   B. If there are no errors, upload your sketch to the Arduino UNO and verify the LEDs blink as specified in step 4 of this activity.

      - If the LEDs do not blink, or do not blink as specified in step 4, evaluate you sketch and/or your breadboarded circuits, correct the error(s), and repeat step 6.
      - If the LEDs blink as specified in step 4, or if time is up and you have not corrected the error(s), proceed to step 7.

7 Record the time (from the clock at the front of the room) when <u>one</u> of these events occur:

   A. You <u>successfully</u> complete the assignment: _______________ (time)

   OR

   B. You quit working on the assignment before successfully completing it:

   _______________ (time)

8. Copy and paste your Arduino sketch into the bottom of this page of this activity sheet.

9. Save this document as "Arduino1.doc" and submit the assignment electronically using the Blackboard submission link.

10. Write your first and last name on the index card provided, and leave the card and your wired Arduino and breadboard at your workstation.

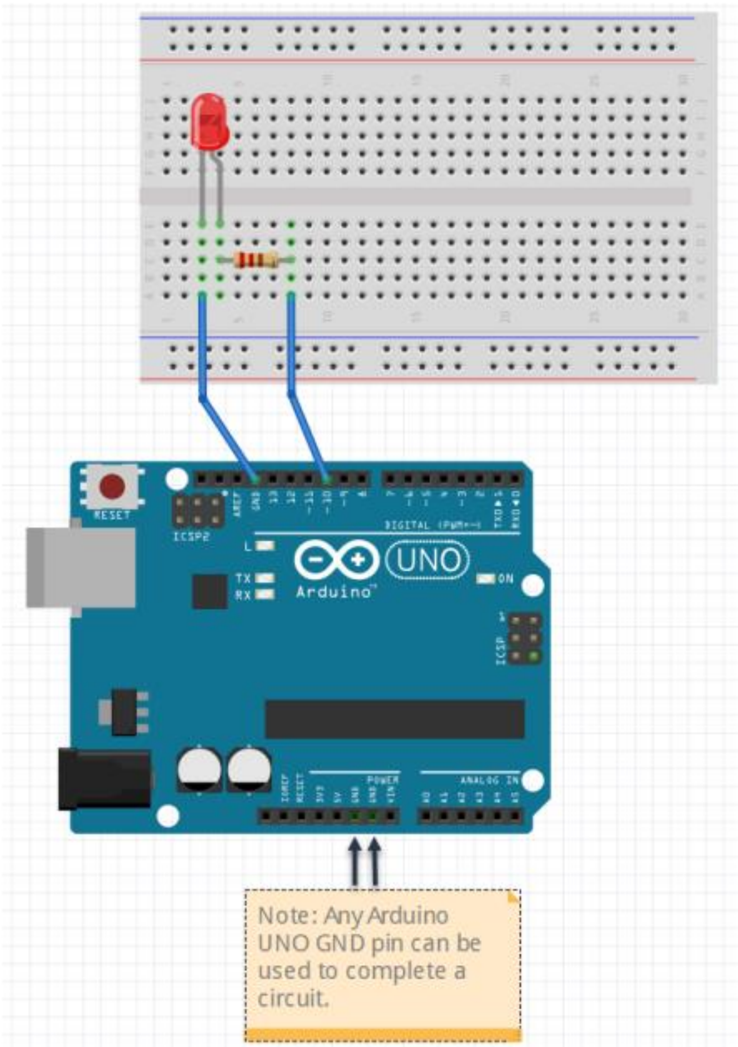
COPY and PASTE your Arduino Sketch below. Be sure to get the ENTIRE program.

(Do this whether or not you were able to successfully complete the activity.)

Appendix C

Arduino In-Class Activity
Student Reference Sheet

Breadboarding: The example below shows a circuit breadboarded from the Arduino UNO's Digital pin #10, through the resistor and LED, and back to ground (GND) on the Arduino UNO. This circuit is electrically complete.



Writing the Sketch: The generic format for the Arduino Sketch statements you may need to use in this activity are provided below:

pinMode(pin_number, OUTPUT or INPUT);

digitalWrite(pin_number, HIGH or LOW);

delay(milliseconds);

Appendix D

Student: ______________________________Group______

Arduino Breadboard Rubric:

Blue	
Circuit is wired to a digital pin	_____ /1
DigitalPin connects to anode (+) of LED	_____ /1
GND pin connects to cathode (-) of LED	_____ /1
Resistor is connected in series with LED	_____ /1
Circuit is wired to GND pin	_____ / 1
RED	
Circuit is wired to a digital pin	_____ /1
DigitalPin connects to anode (+) of LED	_____ /1
GND pin connects to cathode (-) of LED	_____ /1
Resistor is connected in series with LED	_____ /1
Circuit is wired to GND pin	_____ / 1
Total:	**_____ /10**

Arduino Sketch Rubric:

Setup	
pinMode for Blue LED pin_OUTPUT	_____ / 1
pinMode for RED LED pin_OUTPUT	_____ / 1
Loop	
DigitalWrite for BLUE LED HIGH	_____ / 1
Delay for Blue LED 500ms	_____ / 1
DigitalWrite for BLUE LED LOW	_____ / 1
Delay for Blue LED 500ms	_____ / 1
DigitalWrite for BLUE LED HIGH	_____ / 1
Delay for Blue LED 500ms	_____ / 1
DigitalWrite for BLUE LED LOW	_____ / 1
Delay for Blue LED 500ms	_____ / 1
DigitalWrite for RED LED HIGH	_____ / 1
Delay for Red LED 500ms	_____ / 1
DigitalWrite for RED LED LOW	_____ / 1
Delay for Red LED 500ms	_____ / 1
Comments after each line to describe function	_____ / 1
Total:	**_____ / 15**

Sketch/Breadboard Compatibility:

Blue LED pinMode ____, Output corresponds to the breadboard pin chosen	_____ / 1
Red LED pinMode ____, Output corresponds to the breadboard pin chosen	_____ / 1
Total:	**_____ / 2**

Total Project Points_____/27

Appendix E

Arduino Interest, Self-Efficacy, and Knowledge After Instruction

Informed Consent

We are conducting this study to determine the effectiveness of an Arduino lesson and activity on students' self-efficacy, interest, and knowledge of Arduino programming. There is no foreseeable risk as a result of participating in study. Your participation in this study is completely voluntary and declining to participate or discontinuing participation in this study at any time will not result in any penalty or loss of benefits to which you are otherwise entitled.

If you consent to participate in this survey:

1. Remove this page and keep it for your records.

2. Print your first and last name in the blanks at the top of the next page.

3. Complete the attached Interest and Self-Efficacy Survey.

Also, complete the required (graded) Arduino Knowledge Posttest

If you have any questions concerning this study, please feel free to contact either of the individuals listed below:

Dr. Don Johnson, University Professor
Agricultural Education, Communications and Technology
AFLS E-111B
Phone: 479-575-2039
email: dmjohnso@uark.edu

Ms. Iroshi (Ro) Windwalker, IRB Coordinator
Research Compliance
Office: 109 MLKG
email: irb@uark.edu
Phone: 479-575-2208

This survey contains four sections. In the first section, we want to know how you feel about learning Arduino now that you have completed the lesson and hands-on in-class activity.

In the second section, we want to know how confident you are that you can complete specific tasks related to writing Arduino sketches (programs) after having completing the lesson and hands-on in-class activity.

In the third section, we want to know how confident you are that you can complete specific tasks related to breadboarding circuits from the Arduino UNO after having completing the lesson and hands-on activity.

In the final section, we would like to know a little about you and your previous experiences related to Arduino, Arduino programming and similar activities.

Please respond honestly - there are no wrong or right answers.

Please print your first and last name in the blank below and then begin with Part I of the survey.

Name: ____(first and last).

Part I: Attitude Toward Learning About Arduino

Please read each of these statements and rate your level of agreement with each statement by circling the appropriate number to the right of the statement, where 1 = "Strongly Disagree," 2 = "Disagree," 3 = "Neither Agree nor Disagree," 4 = "Agree," and 5 = "Strongly Agree."

Statement	Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
I liked learning about Arduino.	1	2	3	4	5
Arduino fascinated me.	1	2	3	4	5
Learning about Arduino will be of great value to me.	1	2	3	4	5
Learning about Arduino held my interest.	1	2	3	4	5
Arduino was the most annoying topic in this course.	1	2	3	4	5
I would like to learn more about Arduino.	1	2	3	4	5
I think Arduino was boring.	1	2	3	4	5
Learning about Arduino gave me skills I will use in life.	1	2	3	4	5
I am glad that I had the chance to learn about Arduino.	1	2	3	4	5
Learning about Arduino was beneficial to me.	1	2	3	4	5
Learning about Arduino helped me to develop good reasoning ability.	1	2	3	4	5
I was interested in learning about Arduino.	1	2	3	4	5
Learning about Arduino was difficult.	1	2	3	4	5

(Please continue to Part II on the next page)

Part II: Level of Confidence in Writing Arduino Sketches (Programs)

Please read each statement and rate your current level of confidence for that statement by circling the most appropriate number, where 1 = "Very Unconfident," 2 = "Only Slightly Confident," 3 = "Somewhat Confident," 4 = "Moderately Confident," and 5 = "Very Confident."

How confident are you of your ability to?	Very Unconfident	Only Slightly Confident	Somewhat Confident	Moderately Confident	Very Confident
Write Arduino sketch statements that use correct syntax (format).	1	2	3	4	5
Place Arduino statements in the correct section of the sketch (e.g. in the void setup *or* void loop *section*).	1	2	3	4	5
Explain the basic logical structure of an Arduino sketch.	1	2	3	4	5
Write a short Arduino sketch to do a task that is familiar to you.	1	2	3	4	5
Write a short Arduino sketch to do a task that is <u>not</u> familiar to you.	1	2	3	4	5
Write a short Arduino sketch that someone else can understand and use.	1	2	3	4	5
De-bug (find errors and correct) a short Arduino sketch that you wrote, and make it work.	1	2	3	4	5
Mentally trace through a short Arduino sketch written by someone else and explain what will happen when the sketch runs.	1	2	3	4	5
Find ways of overcoming the problem if you get stuck while writing a short Arduino sketch.	1	2	3	4	5
Quickly come up with a suitable strategy for writing an Arduino sketch to do a specific task.	1	2	3	4	5
Manage your time efficiently if you had a deadline for writing an Arduino sketch.	1	2	3	4	5
Find a way to concentrate on your Arduino sketch even, even when there are many distractions around you.	1	2	3	4	5
Find ways of motivating yourself to complete an Arduino sketch, even if the project is of no interest to you.	1	2	3	4	5

(Please continue to Part III on the next page)

Part III: Level of Confidence in Building Arduino Circuits on a Breadboard.

Please indicate how confident you currently are that you can perform each of the following tasks by circling the most appropriate number, where 1 = "Very Unconfident," 2 = "Only Slightly Confident," 3 = "Somewhat Confident," 4 = "Moderately Confident," and 5 = "Very Confident."

	Not at All Confident	Only Slightly Confident	Somewhat Confident	Moderately Confident	Very Confident
Connect a specific analog pin from the Arduino UNO to a breadboard.	1	2	3	4	5
Connect a specific digital pin form the Arduino UNO to a breadboard.	1	2	3	4	5
Connect a ground (GND) pin form the Arduino UNO to a breadboard.	1	2	3	4	5
Connect components (e.g. LEDS and resistors) and wires to the breadboard to build a complete electronic circuit.	1	2	3	4	5
Trace the flow of current through a circuit built on a breadboard.	1	2	3	4	5
Forward-bias a light-emitting diode (LED).	1	2	3	4	5
Calculate the proper size (ohms) resistor to use to protect an LED from excess current.	1	2	3	4	5
Follow a simple electronic circuit schematic to build the circuit on a breadboard.	1	2	3	4	5

(Please continue to Part IV on the next page)

Part IV. Background Information

Directions: Please respond to the following questions in the manner indicated.

1. What is your current academic classification? (circle your response)

   A. Freshman

   B. Sophomore

   C. Junior

   D. Senior

   E. Graduate Student

2. What is your gender? ________________ (fill in the blank)

3. Before studying Arduino programming in this class, did you have any experience doing any type of computer programming? (circle your response)

   NO

   YES

   → *If YES, please describe your previous experience(s):*

4. Did you have any hands-on experience with Arduino before we studied it in ASTM 1613? (circle your response)

   NO

   YES

Thank you!

Appendix F

Name: ____

Arduino Pre/Posttest

Directions: Write your first and last name in the blank above. Select the one best answer to each item by **circling** the **letter** of the answer.

1. An Arduino is a(n) ___. (1 pt.)

   A. Programmable microcontroller

   B. Read-only memory chip

   C. External hard-drive

   D. Basic input/output system (BIOS)

2. Arduino UNO ___ pins can only have a value of either 0 volts or 5 volts. (1 pt.)

   A. Analog

   B Digital

   C. Ground (GND)

   D. Power

3. Which of the following statements about Arduino UNO pins is true? (1 pt.)

   A. Analog pins can only be configured as inputs

   B All digital pins can be configured as either inputs or outputs

   C. Digital pins can only be configured as outputs

   D. All analog pins can be configured as either inputs or outputs

4. A resistor is placed in series with an LED in order to __ (1 pt.)

   A. Increase the voltage at the LED

   B. Protect the LED from a short-circuit

   C. Limit current in the LED circuit to a safe level

   D. Protect the LED from a voltage surge

5. When an LED is placed in a circuit with its long lead (anode) toward the positive terminal and its short (cathode) lead toward the negative terminal , the LED is ___. (1 pt.)

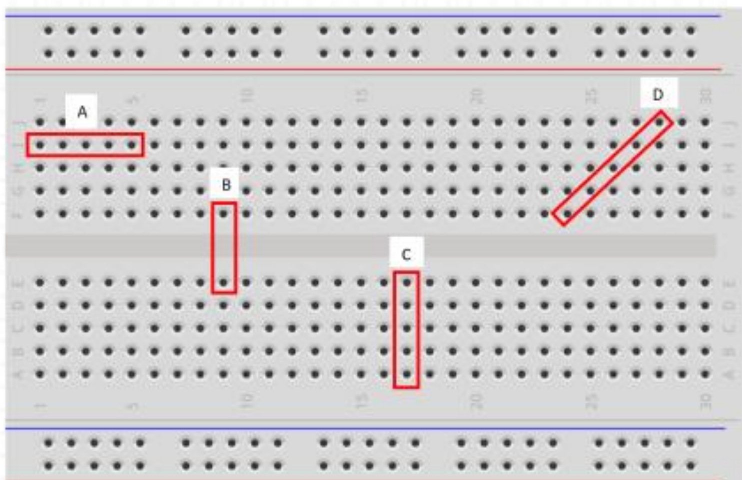
   A. Un-biased

   B. Dual-biased

   C. Reverse-biased

   D. Forward-biased

6. In the breadboard below, which red rectangle (A - D) outlines slots that are electrically connected to each other? (1 pt.)



   A. Rectangle A

   B. Rectangle B

   C. Rectangle C

   D. Rectangle D

7. Which of the following statements would be correct to include in the *void setup* function of an Arduino sketch?

   A. delay(1500);

   B. pinMode(4, OUTPUT);

   C. digitalWrite(A4, HIGH);

   D. digitalWrite(A4, LOW);

8. In an Arduino sketch (program) which of the following statements shows proper syntax to initialize pin 3 as an output?

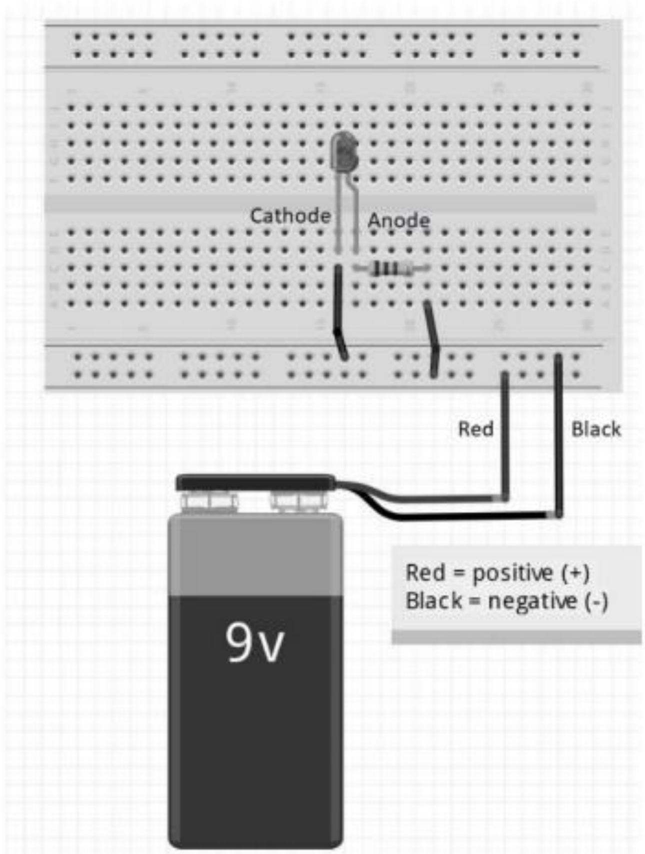
A. pinMode(3, OUTPUT);

B. Pinmode(3, Output);

C. pinmode(3, OUTPUT);

D. PinMode(3, output);

9. You have "breadboarded" the circuit below. With the battery connected as shown, which of the following is true? (1 pt.)



A. The LED will light up because there is a complete circuit

B. The LED will <u>not</u> light up because there is <u>not</u> a complete circuit

C. The LED will <u>not</u> light up because the LED is reverse-biased

D. The LED will <u>not</u> light up because the LED is forward-biased

10. In an Arduino sketch's *void loop* function, which of the following would cause an LED attached to pin 5 to blink on and off once per second? (1 pt.)

A.
```
digitalWrite(5, HIGH);
delay(500);
digitalWrite(5, LOW);
delay(500);
```

B.
```
digitalWrite(5, High);
delay(500);
digitalWrite(5, Low);
delay(500);
```

C.
```
digitalWrite(5, HIGH);
delay(0.5);
digitalWrite(5, LOW);
delay(0.5);
```

D.
```
digitalWrite(5, High);
delay(0.5);
digitalWrite(5, Low);
delay(0.5);
```

11. In the function below which of the following is true about the text to the right of the two forward slashes (//)? (1 pt.)

```
analogRead(A0); //Read the input pin
```

A. This is a "function" the program will read and execute

B. This is a "comment" the program will neither read nor execute

C. This is a "sub-routine" the program will read and execute

D. This is a "priority function" the program will execute before the "analogRead" function

12. What will happen after an ArduinoUNO executes the last statement (line) in the sketch (program) shown below?

```
void loop{}  {
digitalWrite(7, HIGH);
delay(1000);
digitalWrite(7, LOW);
delay(1000);
}
```

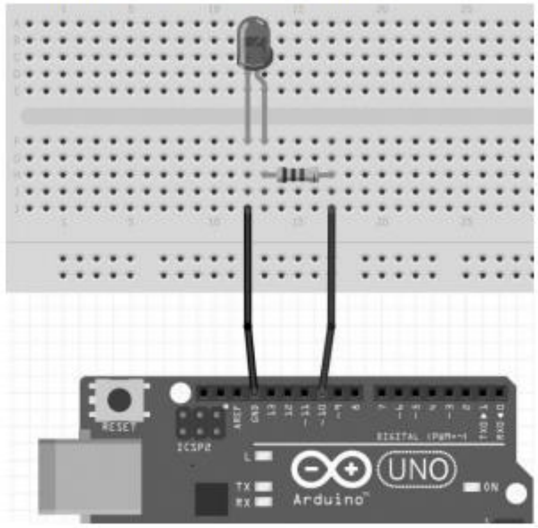
A. It will pause and await further program statements

B. It will execute all of the statements again, starting with the first statement

C. It will go into "power save" mode

D. It will switch "off"

13. In the circuit below, the Arduino UNO is to turn "on" the LED. Which of the following statements would be required in the *void loop* function of the Arduino sketch (program)?



A. digitalWrite(GND, HIGH);

B. pinMode(10, OUTPUT);

C. pinMode(GRD, Output);

D. digitalWrite(10, HIGH);

14. You just compiled an Arduino sketch (program) and received the message shown at the bottom of the screen below. What does this message indicate?

A. The token ring adapter is not connected

B. There is an error in the sketch

C. The sketch has been uploaded and is ready to run

D. There is no connection between the Arduino UNO and the computer

15. Which of the following statements best describes your level of confidence that your answers on this posttest are correct?

A. I am not at all confident that my answers are correct

B. I am fairly confident that my answers are correct

C. I am extremely confident that my answers are correct