

Development of Sensing and Programming Activities for Engineering Technology Pathways Using a Virtual Arduino Simulation Platform

Dr. Murat Kuzlu, Old Dominion University

Murat Kuzlu joined the Department of Engineering Technology, Old Dominion University (ODU) in 2018 as an Assistant Professor. He received his B.Sc., M.Sc., and Ph.D. degrees in Electronics and Telecommunications Engineering. He worked as a senior researcher at TUBITAK (Scientific and Technological Research Council of Turkey) between 2006 and 2011. Before joining ODU, he was a Research Assistant Professor at Virginia Tech's Advanced Research Institute. His research interests include cyber-physical systems, smart cities, smart grids, artificial intelligence, engineering education and next-generation wireless networks.

Dr. Vukica M. Jovanovic, Old Dominion University

Dr. Vukica Jovanovic is a Chair of Department of Engineering Technology and Associate Professor and Batten Endowed Fellow in Mechanical Engineering Technology Program. She holds a Ph.D. from Purdue University in Mechanical Engineering Technology, focuses on Digital Manufactur

Dr. Otilia Popescu, Old Dominion University

Dr. Otilia Popescu received the Engineering Diploma and M.S. degree from the Polytechnic Institute of Bucharest, Romania, and the PhD degree from Rutgers University, all in Electrical and Computer Engineering. Her research interests are in the general areas of communication systems, control theory, signal processing and engineering education. She is currently an Associate Professor in the Department of Engineering Technology, at Old Dominion University in Norfolk, Virginia, and serves as the Program Director for the Electrical Engineering Technology Program. In the past she has worked for the University of Texas at Dallas, University of Texas at San Antonio, Rutgers University, and Politehnica University of Bucharest. She is a senior member of the IEEE.

Salih Sarp, Old Dominion University

Salih Sarp is a Ph.D. student in the Electrical and Computer Engineering department at Old Dominion University, USA. Currently, he is developing AI applications and sensor fusion models. Previously, he received his BS degree in Electronics and Communicati

Development of Sensing and Programming Activities for Engineering Technology Pathways using a Virtual Arduino Simulation Platform

Abstract

The Arduino platform has long been an efficient tool in teaching electrical engineering technology, electrical engineering, mechatronics, and computer science concepts in schools and universities and to introduce new learners to programming and microcontrollers. Numerous Arduino projects are widely available through the open-source community. They can help students have hands-on experience in building circuits and programming electronics with a wide variety of topics that can make learning electrical prototyping fun. However, the educational fields of electrical engineering and electrical engineering technology need continuous updating to keep up with the continuous evolution of the computer system. Although the traditional Arduino platform has proved to be an effective tool for teaching electrical engineering concepts for many years since its release, another tool has recently been introduced within the open-source community, a virtual simulation software for the Arduino platform that may provide more educational benefits. This paper discusses how using Arduino simulators can potentially engage K-12 students in a learning experience, which can further lead them to computer science, mechatronics, electrical engineering, and electrical engineering technology career pathways. The paper also presents sample Arduino projects showcased through readily available Arduino simulators.

Introduction

As time progresses, so does the development of technologies, which affect the world around us [1]. Nowadays, computers are found in almost every facet of a person's daily life, ranging from sensors used in various devices [2] to personal devices and home devices, as well as advanced computing that supports various daily activities. Computers can be found on a personal level in the form of a smartphone, which can allow people to communicate with each other across the world. They can also be found at home in the form of a personal computer or a desktop, which can run programs such as video games or even watch videos on the internet [3]. As the world becomes increasingly digital, it is important for individuals to understand how computers work and how they can be used to enhance various aspects of our lives. On an enterprise level, some computers are used as servers to allow for the access of information to and from a database. Finally, computers are even used in advanced research facilities to perform computations of complex mathematical formulas at incredibly high speeds. While there are many different types of computers, they ultimately share the same goal of carrying out complex tasks much faster and more efficiently than humans. Therefore, understanding how computers interact with the physical environment using microcontrollers is beneficial for students of multiple majors, ranging from those that deal with the hardware side of things, such as electrical majors in engineering or engineering technology, to majors that use microcontrollers to interact with mechanisms such as mechatronics, as well as for students in majors that are only dealing with the software side of things, such as computer science.

Open-Source Microcontrollers

A microcontroller is a small computer on a single VLSI integrated circuit (IC) chip with one more processor core. It has internal memory and options to attach to the various inputs and outputs through the peripheral devices [4,5]. A microcontroller is a small standalone form-factor computer with a processor, a computer on a chip [6], and memory. It is used for monitoring or controlling tasks [7]. Microcontrollers are found in embedded systems such as cars, microwaves, or TV remotes. They are used to execute instructions defined by software, usually to automate a specific task or to collect data using sensors [8]. Another common example of a microcontroller can be observed within thermostats, taking data on temperatures using an analog sensor and then responding by controlling the temperature of the house [9, 10]. The Internet of Things (IoT) is an industry where microcontrollers play an important role [11]. A part of IoT consists of wireless sensor networks to collect and transfer large amounts of data in real-time.

Microcontrollers provide a practical solution for computational performance with low power consumption [12]. The computer that is discussed in this paper is a development board developed by a group of engineers in Italy called Arduino, which is like that of a microcontroller, but contains much more features to make it beginner friendly. As stated by their designers, Arduino designs manufactures, and supports electronic devices and software, allowing people worldwide to easily access advanced technologies that interact with the physical world. The Arduino is a programmable board that consists of a microcontroller, as discussed previously, along with analog and digital inputs and outputs (I/O) [13]. An example Arduino board is the Arduino Uno, which consists of an ATmega328 microcontroller, 14 digital input/output pins, six (6) analog input pins, a USB port to connect to a computer, and more. A diagram of the Arduino Uno is shown in Figure 1 for reference.

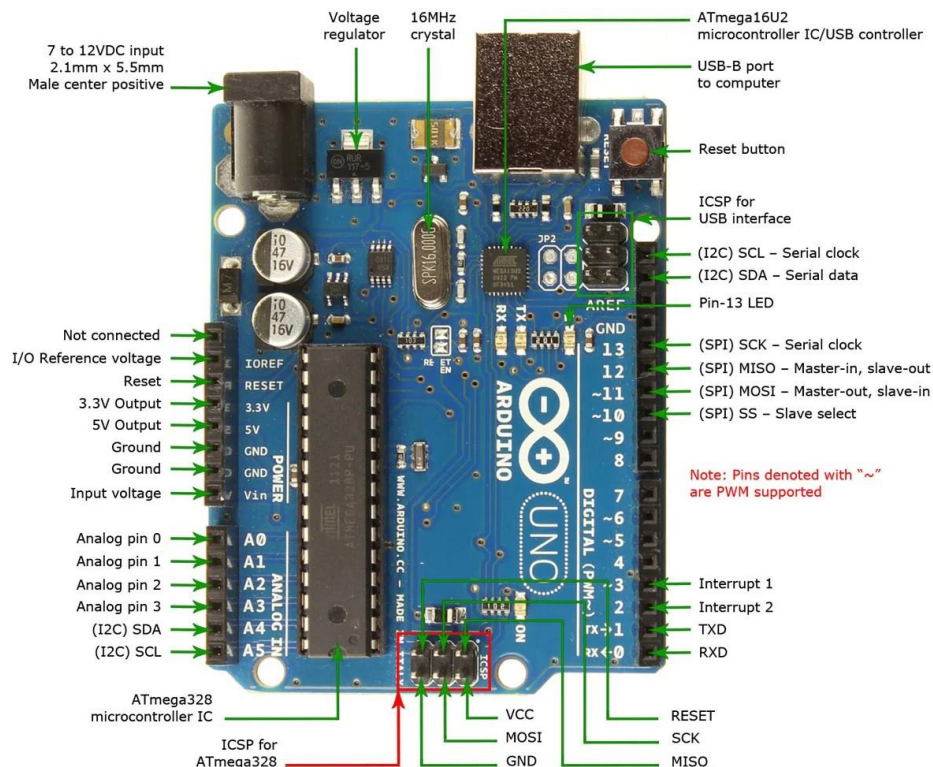


Figure 1: Arduino Uno Diagram from Jameco Electronics [14]

In the realm of electrical engineering technology and electronics, the Arduino platform plays an important role. To give context, the education of electronics faces the challenge of the rapid development of technologies of today [15, 16]. Electronics are constantly being integrated with more applications as time passes, which then increases the importance and dependence on electronics. The importance of electronics should then also justify a quality education for students who will be responsible for the state of electronics in the future to come. This paper will discuss the use of the virtual Arduino simulation concept to enhance and enrich electrical engineering teaching with the integration of various I/O and sensors. The Arduino platform is an important tool in electrical engineering and electronics education because it provides a user with an easily accessible board capable of running many types of electronics projects along with the guidance of a large active community [16, 17]. The Arduino platform also has the advantage of low-cost boards, compatibility with Windows, MacOS, and Linux operating systems, using a simple IDE, and is open source [13]. The low cost of Arduino boards and the compatibility with many operating systems make it accessible to most students and even institutions that intend on integrating the platform within their curriculum. The simple IDE is helpful for those who are new to programming as it only contains the text editor where code is written and a few buttons to compile, upload, or change the settings of the board [18].

The Arduino Simulator

Although the Arduino platform is known to be relatively accessible and low-cost, there exists an alternative that is potentially more beneficial in an educational environment: an Arduino simulator. Arduino simulators are software designed to imitate the hardware of physical Arduino boards. While Arduino simulation software cannot run projects in the physical world, it has advantages over traditional Arduino boards, such as increased device reliability, increased safety from electrical components, even lower costs, and opportunities for distance learning [19].

Physical Arduino boards are no exception to the eventual wear and tear of electronic devices with usage over time. Components on the board, such as transistors and diodes that generate heat, are prone to damage or even failure [20]. With an Arduino simulator, the board is simulated digitally, resulting in no damage to any components. Many Arduino projects require the use of breadboards, sensors, resistors, and other components to build circuits. It would require a student to have a basic knowledge of circuits and some safety precautions when dealing with electrical components. An Arduino simulator ensures the safety of students from overheating components and live wires while still offering the experience of building circuits. Arduino boards are usually priced no more than 60 dollars per board, which is relatively cost-effective given the amount of learning experience that can be gained from Arduino projects.

Similarly, some Arduino simulators, such as TinkerCAD or SimulIDE, offer the same experience free of charge. These simulators also come with various components, such as LEDs, sensors, and resistors, to be used indefinitely for Arduino projects. This makes the Arduino platform more accessible by lowering the barrier of entry by not having to purchase a physical board or the external components required in a given project.

Arduino simulators are also an effective learning tool, even with distance learning and online classrooms. Ever since the COVID-19 pandemic, the world has seen an increase in remote operations from companies to schools and universities [21]. In an online electrical engineering course, Arduino simulators have the advantage over physical boards as they can be accessed anywhere if the student can download the software.

Ultimately, an Arduino simulator is software that can provide the circuit building and programming experience offered by traditional Arduino boards but in a more cost-effective, safer, more reliable, and more accessible manner which makes it an effective learning tool in an electrical engineering program.

Methods and Sensing and Programming Activities

This section discusses the features of a popular Arduino simulation software called TinkerCAD and sample projects that could be used within an electrical engineering program.

TinkerCAD Circuits

TinkerCAD is a free web application for 3D model design, circuit design, and programming in general [22]. In TinkerCAD Circuits, a user is presented with a top-down view of the entire circuit, which displays the components used in the ongoing project and the wiring used to connect all the components.

A menu on the right side of the simulator provides a wide variety of components, such as resistors, LEDs, buttons, capacitors, and sensors, that a user may want to use in their project. These components can be accessed by a simple drag and drop to the circuit display mentioned previously. A sample view of the TinkerCAD simulator, including the display and component menu (Figure 2).

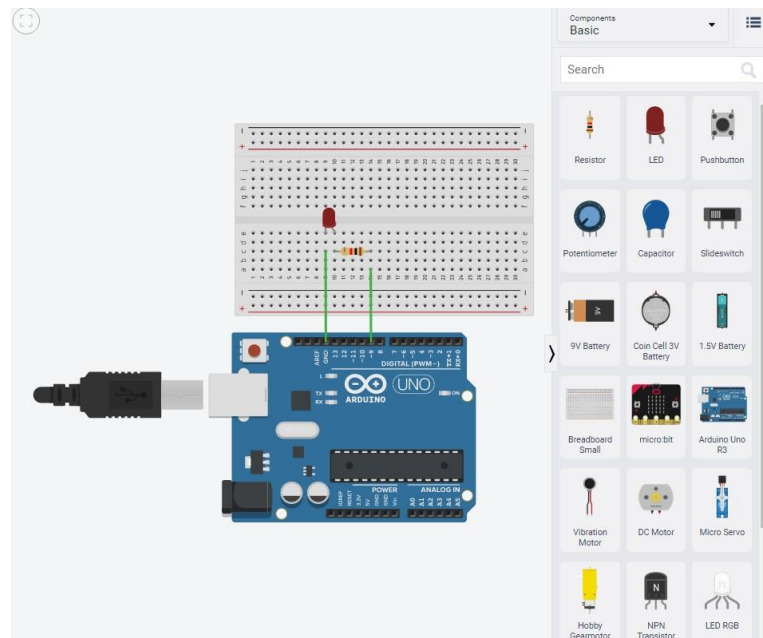


Figure 2: TinkerCAD Circuits User Interface.

In addition, this Arduino simulator also has an interface for programming the Arduino, including a traditional text editor and an option for block coding (Figure 3). A serial monitor also showcases any output designated in the source code. This paper will discuss the use of TinkerCAD Circuits when simulating the Arduino platform.

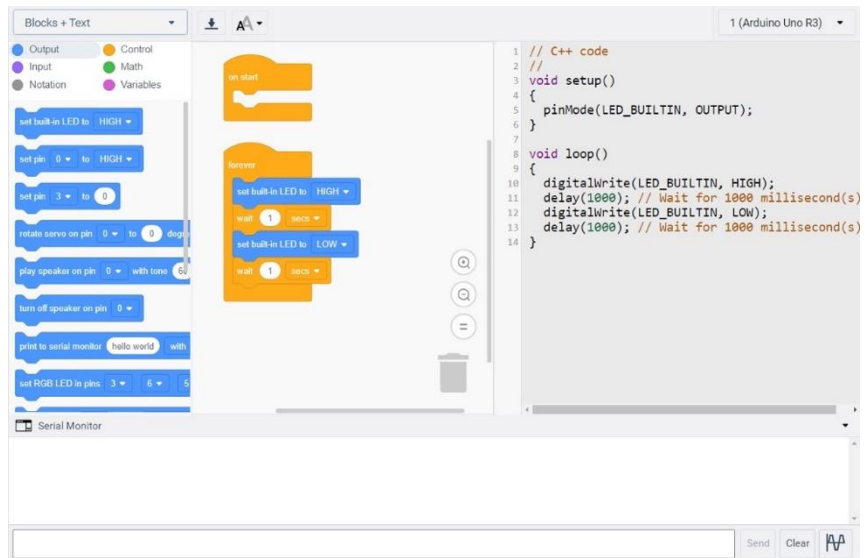


Figure 3: Text editor, block coder, and serial monitor

RGB LED Project

In this project, an RGB LED is programmed to display red, green, and blue light. This is done by connecting the digital pins 9, 10, and 11 to 220 Ω resistors and then connecting to the red, blue, and green pins on the RGB LED, as shown in Figure 4. The cathode portion of the LED is then connected to the GND slot on the Arduino to complete the circuit. The schematic diagram for the circuit on TinkerCAD Circuits is shown below, along with the source code in the project.

According to the source code, pin 11 is defined as red, pin 10 is defined as green, and pin nine (9) is defined as blue, and they are all set as an output on the Arduino. In the loop section of the code, red is set to turn on and delay for 1000 milliseconds or 1 second and repeats this process with green and then blue.

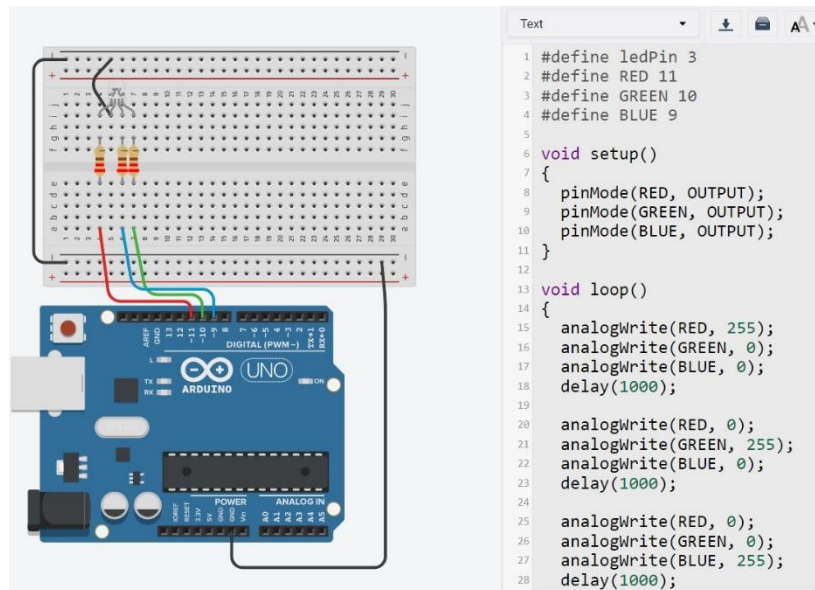


Figure 4: RGB LED schematic with source code

Temperature Sensor Project

In this project, a TMP36 temperature sensor is used to measure the temperature of the surrounding environment. The sensor is connected to the Arduino with the Vout pin going to the analog input slot A0. The schematic diagram and the source code are shown in Figure 5. In the code, the A0 analog pin is set as an input. The program then takes the value from the A0 pin, converts it to Celsius, and prints the value onto the serial monitor. With the TinkerCAD simulator, the user is able to set the temperature of the simulation manually by hovering over the TMP36 sensor and choosing a desired temperature.

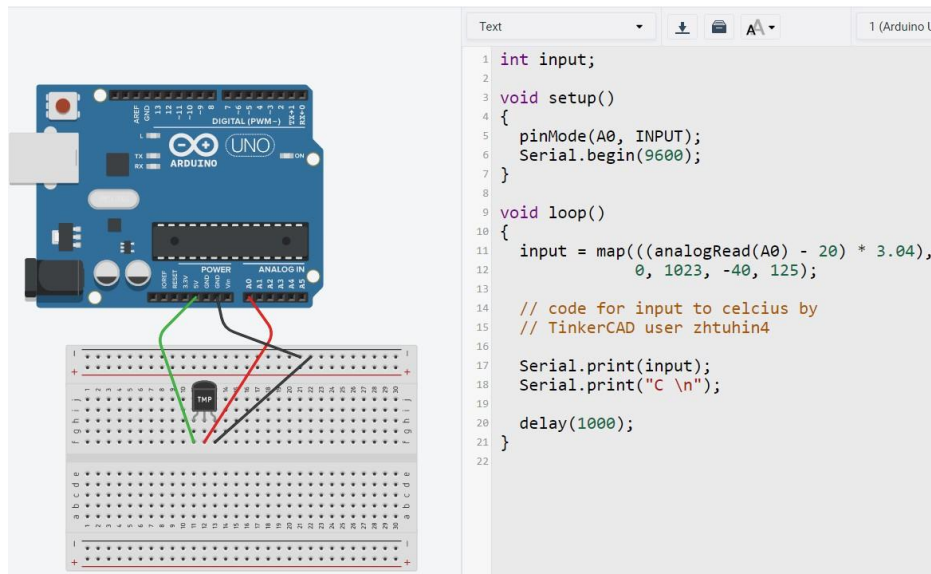


Figure 5: Temperature sensor schematic with source code

Push-button and LED Project

In this project, a push-button is used to read user input and turns on an LED whenever the button is pushed down. Firstly, the push-button is connected to the 3.3V terminal and ground of the Arduino to power the button and then is connected to pin seven (7) to be read as input. Next, a standard LED is connected to pin six (6) and then to the ground. During the simulation, the LED should light up when the button is pressed, and the serial monitor should output a message confirming that the button was pressed. The schematic and source code for this project (Figure 6).

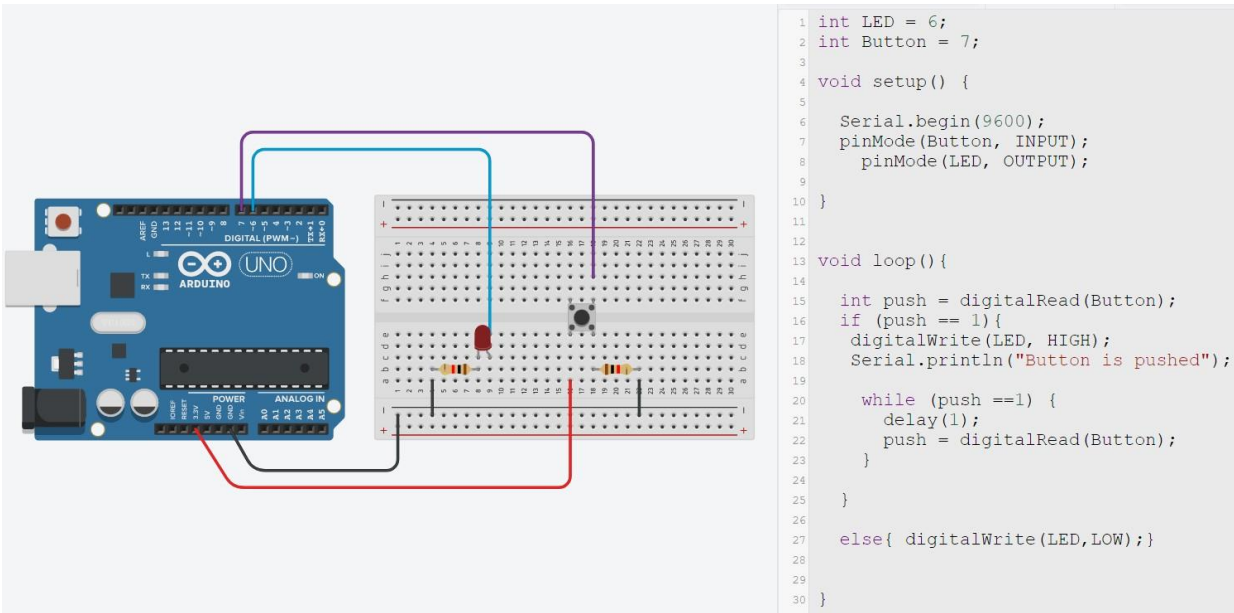


Figure 6: Push-button and LED Project schematic and source code

In the code, pin 7 for the push-button is set as an input, and pin 6 for the LED is set as an output. Within the loop, the value from the button is read and used in an if/else statement. If the button is pushed, then the program will send a HIGH signal to the LED and then print out to the serial monitor that the button is pushed. While a statement is used to prevent the program from executing the button output too many times if pushed once, an effect known as bouncing.

Keypad Project

In this project, a 4x4 keypad is connected to the Arduino using digital pins 2-9 (Figure 7). During the simulation, the serial monitor will display the corresponding character when any key on the keypad is pressed. The rows and columns are defined and mapped to the keypad in the source code below (Figure 8). The characters on the keypad are then defined in a 4x4 matrix. Next, the connections are mapped to the corresponding pins on the Arduino, and finally, an if statement is used to output the corresponding key when pressed.

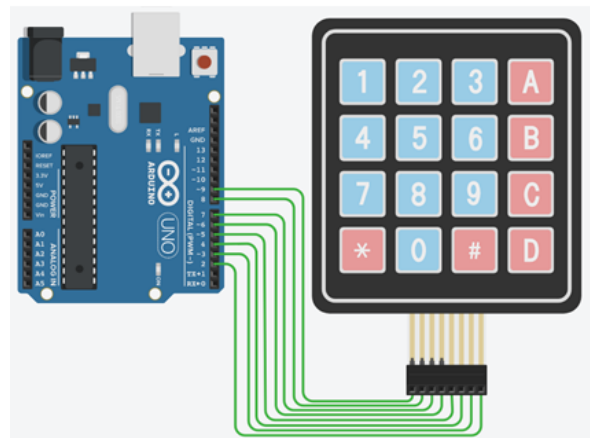


Figure 7: Keypad project schematic


```
#include <Keypad.h>
const byte ROWS= 4;
const byte COLS= 4;

//Key Mapping
char keymap[ROWS][COLS]=
{
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};

// connecting to pins
byte rowPins[ROWS] = {9,8,7,6}; //Rows 0 to 3
byte colPins[COLS]= {5,4,3,2}; //Columns 0 to 3

//initializes an instance of the Keypad class
Keypad myKeypad= Keypad(makeKeymap(keymap),
rowPins, colPins, ROWS, COLS);

void setup()
{
Serial.begin(9600);
}

void loop()
{
char keypressed = myKeypad.getKey();
if (keypressed != NO_KEY)
{
Serial.println(keypressed);
}
}
```

Figure 8: Keypad project source code

Results

This section showcases the output of the sample projects included in the paper above. It includes any visual observations in the circuit and any data presented in the serial monitor.

In the RGB LED project, the LED is observed to light up as red for a second, then switching to green for one second, and finally switching to blue for one second. This process is seen to be repeated indefinitely until the simulation is manually stopped. The observations for each of the occurrences are shown in Figure 9. Since this project had no numerical data to output, the serial monitor was not included.

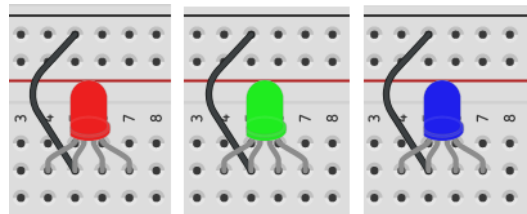


Figure 9: RGB LED observations

In the temperature sensor project, the temperature was randomly set during the simulation to imitate the fluctuations of temperature in the real world. Below are the results displayed on the serial monitor (Figure 10). TinkerCAD Circuits also provides a plot of the temperature in Celsius in addition to the serial monitor, which helps visualize the data collected over time.

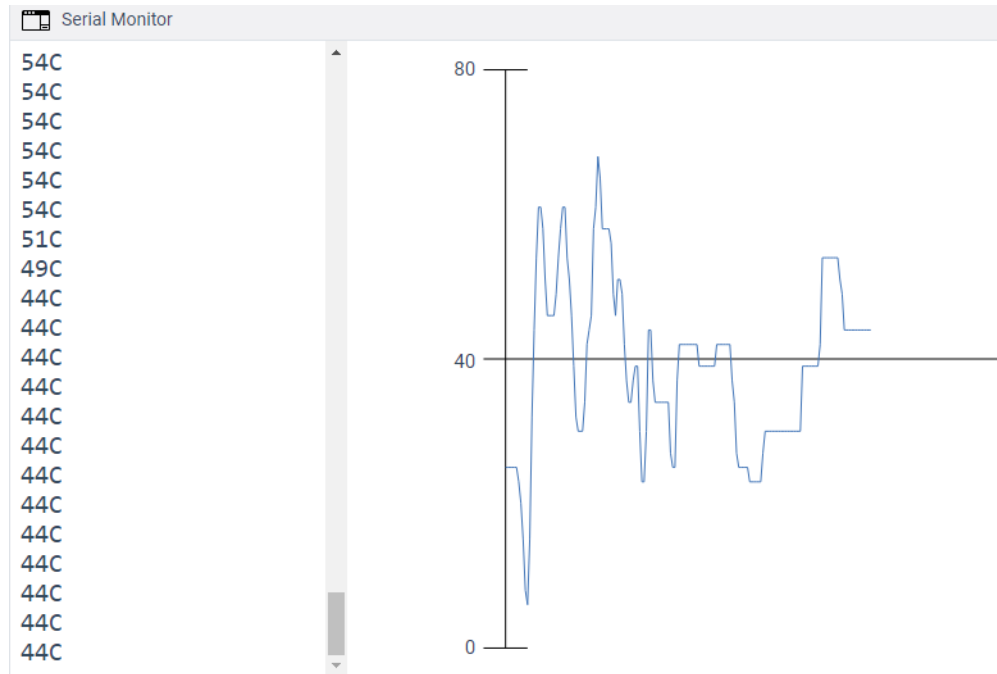


Figure 10: Temperature Sensor serial monitor and plot

In the push-button and LED simulation, the button was pushed manually once. Below are the results of the LED and the serial monitor after the button was pushed (Figure 11). Without the code to counteract the bouncing effect, the serial monitor would have many instances of the text, even though the button was only pushed once.

The LED is observed to be turned on because of the HIGH signal sent from the Arduino, and the serial monitor has outputted text confirming that the button was pushed.

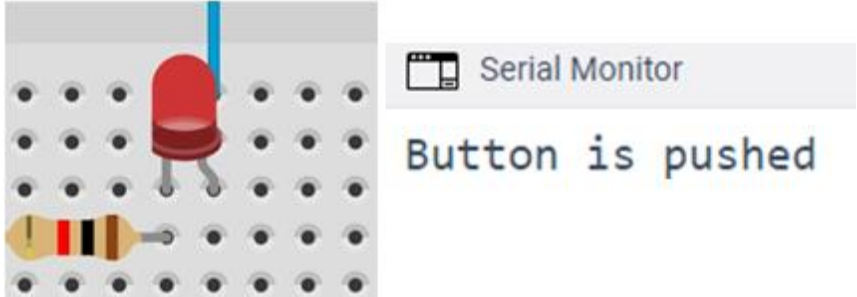


Figure 11: Push-button and LED output and serial monitor

In the keypad project during the simulation, the keypad was pushed in the order: *9 D B * A*. In the serial monitor shown below, the output matches the keys that were pressed in the corresponding order (Figure 12).

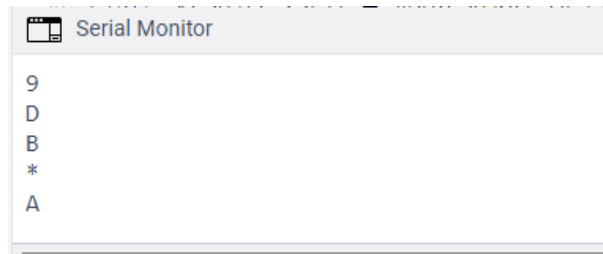


Figure 12: 4x4 Keypad output

Discussion

This section further discusses the results of the sample projects mentioned above and how the use of a virtual Arduino simulator will benefit electrical engineering courses. In the RGB LED project, a student will be able to learn circuit concepts such as using a resistor to lower the electrical current and know how the connections from the Arduino affect the different pins on the LED. From this project, the student can also gain programming experience by observing the effect of the RGB LED by changing parameters such as brightness or delay. In the temperature sensor project, a student can get experience in using the serial monitor to observe the output data during the simulation and how it relates to the plot. The student will also learn about the use of sensors, which is how electronics collect analog data from the physical world. In the push-button and LED project, the student learns about a simple button for user input as well as the bouncing effect. They can also learn about the cathode and anode terminals of a normal LED. In the keypad project, the student learns about the use of a matrix within programming to easily map different values in rows and columns.

Conclusion

As mentioned previously in the paper, computers play an important role in society in making our lives easier. The microcontroller is an important computer that automates tasks and collects data in the real world at a low cost, making electrical prototyping available for anyone, even at home. The rapid advancement of computer technology and the importance of computers, in general, have also raised concerns about the quality of education. Arduino is often seen as an excellent supplement to computer science, electrical engineering, and electrical engineering technology programs. As mentioned previously, it offers a platform for learning basic circuit design and programming on an affordable computer similar to a microcontroller. However, the use of virtual software to simulate the Arduino platform promises to be an even better fit for computer science, electrical engineering, and electrical engineering technology courses. Compared to traditional Arduino boards, the simulators are usually free to use, more reliable, accessible, safer, and adaptable to inevitable needs for remote work. Also, users can work with multiple Arduino boards. With the sample projects discussed in this paper, it is shown that a virtual Arduino simulator is capable of the potential teaching experience that is offered with the traditional Arduino platform. As the demands for a more programming and embedded savvy workforce continue to grow, virtual Arduino simulation should continue to remain a useful tool for K-12 teachers to expose students to computer science, electrical engineering, and electrical engineering technology, especially for the ones with limited resources and non-admin rights to the classroom computers.

Acknowledgment

The project team wishes to acknowledge funding from the U.S. Department of Education, Office of Career, Technical, and Adult Education, Division of Academic and Technical Education, Innovation and Modernization Program, award # V051F190072 “Computer Science Principles and Cybersecurity Pathway for Career and Technical Education”.

References

- [1] E. Lally, *At home with computers*. Routledge, 2020.
- [2] M. Javaid, A. Haleem, S. Rab, R. P. Singh, and R. Suman, “Smart performance of virtual simulation experiments through Arduino tinkercad circuits,” *Sensors International*, vol. 2, no. 100121, pp. 1–10, 2021.
- [3] D. Morley and C. S. Parker, *Understanding computers: Today and tomorrow, comprehensive*. Cengage Learning, 2014.
- [4] H.M.D. Toong, Microprocessors. *Scientific American*, 237(3), pp.146-161, 1977
- [5] L. D. Wittie, “Microprocessors and microcomputers,” *Encyclopedia of Computer Science*, vol. January, no. 2003, pp. 1161–1169, 2003.
- [6] J. H. Davies, *MSP430 microcontroller basics*. Elsevier, 2008.
- [7] G. Gridling and B. Weiss, “Introduction to microcontrollers,” *Vienna University of Technology Institute of Computer Engineering Embedded Computing Systems Group*, 2007.
- [8] X. F. Carelse, “An introduction to the industrial applications of microcontrollers,” *Physica Scripta*, vol. 2002, no. T97, p. 148, 2002.
- [9] M. M. C. J. G. B. B. Zungeru, A. M., “Design and simulation of an automatic room heater control system,” *Heliyon*, vol. 4, no. 6, p. e00655, 2018.
- [10] R. Kumar, “Design implementation of intelligent digital home thermostat using npx low power 32bits lpc2148 microcontroller,” *International Journal of Advanced Engineering Research And Science*, vol. 1, no. 2, pp. 57–60, 2014.
- [11] A. M. K. S. M. K. B. S. Javed, F., “Internet of things (IoT) operating systems support, networking technologies, applications, and challenges: A comparative review,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 2062–2100, 2018.
- [12] R. Chéour, S. Khriji, M. Abid, and O. Kanoun, “Microcontrollers for IoT: Optimizations, computing paradigms, and future directions,” in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, 2020, pp. 1–7.
- [13] R. Singh, A. Gehlot, B. Singh, and S. Choudhury, “Introduction to Arduino,” in *Arduino-Based Embedded Systems*, 1st ed. CRC Press, 2018, pp. 3–6.
- [14] “Arduino uno pinout diagram.” [Online]. Available: <https://www.jameco.com/Jameco/workshop/CircuitNotes/CN-arduino-uno.html>
- [15] J. J. Rodriguez-Andina, L. Gomes, and S. Bogosyan, “Current trends in industrial electronics education,” *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3245–3252, 2010.
- [16] J. Hurtuk, M. Chovanec, and N. A´ dam, “The Arduino platform connected to education process,” *IEEE 21st International Conference on Intelligent Engineering Systems (INES)*, 2017, pp. 000 071–000 076.
- [17] A. A. Galadima, “Arduino as a learning tool,” in *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, 2014, pp. 1–4.
- [18] M. Fezari and A. Al Dahoud, “Integrated development environment “IDE” for Arduino,”

- WSN applications*, pp. 1–12, 2018.
- [19] P. F. Goncalves, J. Sa´, A. Coelho, and J. Dura˜es, “An Arduino Simulator in Classroom - a Case Study,” in *First International Computer Programming Education Conference (ICPEC 2020)*, 2020.
- [20] A. A. Almubarak, “The effects of heat on electronic components,” *Int. J. Eng. Res. Appl.*, vol. 7, no. 5, pp. 52–57, 2017.
- [21] I. Koksai, “The rise of online learning,” *Forbes*, 2020.
- [22] “From mind to design in minutes.” [Online]. Available: <https://www.tinkercad.com/>