Science Education International
36(4)_2025,429-440
https://doi.org/10.33828/sei.v36.i4.6

DESIGN AND DEVELOPMENT PAPER

Design and Development of Scratch–Arduino Educational Tools for Teaching Electricity Concepts

Ioannis Balouktsis*, Gerasimos Kekkeris

Department of Primary Level Education, Democritus University of Thrace, Greece

***Corresponding Author:** ibaloukt@eled.duth.gr

ABSTRACT

This study aimed to design and develop a series of digital educational tools that facilitate the teaching and learning of electrical concepts among middle-school students. The tools were created using Scratch 3, which is a visual programming environment developed by MIT. Physical computing is integrated using the Arduino platform. They focused on fundamental electricity concepts, including Ohm's law, two- and three-way "aller-retour" switch circuits for lighting control, and ON–OFF switch circuits for operating two or three LEDs. In addition, two interactive education assessment tools were developed to support and evaluate the learning outcomes. The integration of digital simulations with real-world circuits enables bidirectional interactions between physical and virtual environments. Students can control real electrical circuits through simulations, while influencing the digital environment through physical input. This approach fosters a deeper understanding of the electric current, voltage, resistance, and circuit logic. Designed for students aged 11–15 years, these tools combine visual representation with hands-on exploration to enhance student engagement and conceptual comprehension. An interactive and experiential learning environment contributes to effective knowledge acquisition and retention in the field of electricity.

KEY WORDS: Arduino, computational thinking, electricity education, interactive simulations, physical computing, Scratch 3.

INTRODUCTION

Digital technology has brought about significant changes in the way students explore and understand scientific phenomena. In physics education – and particularly in the teaching of electricity – core concepts such as electric current, voltage, and resistance often remain abstract and "invisible," making them difficult for students to grasp (McDermott and Shaffer, 1992; Shipstone, 1984). To address these challenges, interactive simulations offer dynamic visualizations of such concepts, facilitating learning and enhancing student engagement (De Jong, 2006; Rutten et al., 2012).

One of the most widely used environments for creating educational applications is Scratch, a block-based programming platform developed by MIT (Resnick et al., 2009). Early work on Scratch emphasized its potential as a low-threshold programming environment that supports creativity, experimentation, and learning through making (Maloney et al., 2004). Scratch embodies the principles of constructivism and constructionist learning (Papert, 1980). According to this approach, students learn to design, create, and share projects based on their personal interests and experiences. Through this process, students become active creators of knowledge and develop computational thinking as they model and simulate scientific phenomena (Brennan and Resnick, 2012; Grover and Pea, 2013; Lye and Koh, 2014).

However, purely digital simulations lack the tangible and experiential elements that characterize real experiments. The emerging field of physical computing bridges digital environments with real electronic systems, thereby offering a method to overcome this limitation (Blikstein, 2013). Several studies have demonstrated that Arduino-based and physical computing activities enhance students' engagement, problem-solving skills, and understanding of STEM concepts in secondary education (Barrera et al., 2019; Govender et al., 2021; Martín-Gutiérrez et al., 2021). Connecting Scratch with Arduino (using, e.g., the s3OneGpio extension) allows students to simultaneously interact with virtual and physical objects, thereby gaining a deeper and more integrated understanding of the phenomenon.

This paper presents a suite of digital tools for teaching key electrical concepts such as the relationships between voltage, current, and resistance, as well as the effect of switching configurations on current flow. The suite includes five main educational tools and two assessment tools.

- A simulation of Ohm's Law
- Two- and three-way switch circuits (aller-retour)
- ON–OFF switch circuits (series and parallel) for controlling two or three LEDs, and
- Two interactive self-assessment quizzes.

All tools exhibited a high level of design and functionality, both computationally and graphically. Moreover, through the Arduino platform and its Scratch extension, the tools incorporate physical computing, enabling bidirectional interactions between the physical circuits and simulations.

These tools are intended primarily for students aged 11–15 years and aim to promote active inquiry-based learning through visualization, interaction, and immediate feedback. The bidirectional integration of physical computing enables the physical circuit to control the simulation, and vice versa, creating a fully interactive learning environment. This environment combines programming, experimentation, and exploration, fostering students' active participation and understanding of electrical concepts, while providing a comprehensive educational experience that integrates theory and practice into a coherent learning framework.

THEORETICAL FRAMEWORK

Research on computer-based learning environments has demonstrated their significant contribution to conceptual change and active student engagement in the learning process (De Jong and van Joolingen, 1998; Linn and Eylon, 2011). Simulations allow learners to explore causal relationships and test hypotheses that are difficult to implement in conventional laboratories (De Jong 2006; Wieman, 2008). When such tools are applied under guided instruction, including phases of prediction, observation, and feedback, they strengthen metacognitive processes and lead to substantial learning gains (Rutten et al., 2012).

Teaching electricity presents challenges because the phenomena are "invisible" and students often hold persistent alternative conceptions (McDermott and Shaffer, 1992; Engelhardt and Beichner, 2004). For example, learners might believe that current is "used up" or that voltage "flows" through wires (Shipstone, 1984). Digital simulations can address these misconceptions by making the relationships between current, voltage, and resistance visible through immediate graphical feedback (Adams, 2010; Perkins et al., 2006).

Furthermore, combining virtual and real experiments is particularly effective (Olympiou and Zacharia, 2012; Wörner et al., 2022). Recent systematic reviews and meta-analyses highlight that combining virtual simulations with real or physical experiments leads to significantly improved conceptual understanding, inquiry skills, and learner engagement in science education (Ainsworth, 2022; Banda & Nzabahimana, 2021). This dual approach strengthens the connection between theory and practice as student's transition from screen to physical circuits, thus improving both conceptual understanding and hands-on learning.

Theories of constructionist learning (Papert, 1980) and computational thinking (Wing, 2006) emphasize the importance of active knowledge creation through the combined use of digital and physical artifacts. Scratch platform supports this philosophy, allowing students to focus on logic and creativity rather than code syntax (Kafai and Burke, 2009; Kafai and Burke, 2015). Integrating Scratch with Arduino increases learners' motivation and understanding of the system (Ntourou et al., 2021; Sarı and Yıldırım, 2022; Marín-Marín et al., 2024).

Despite this promise, few studies have examined true bidirectional integration, in which digital and physical environments interact in real-time and influence each other. In addition to achieving a high level of computational and graphical design, this study proposes a comprehensive approach that incorporates two-way interaction, aiming to further improve the educational experience and students' understanding of learning environments that merge digital and physical components.

METHODOLOGY

These tools were developed based on Scratch 3 to create an intuitive, visually dynamic, and pedagogically grounded environment. Each tool includes interactive elements, graphics, motion, and sound, following a Design-Based Research methodology (Design-based Research Collective, 2003). Each tool was designed to make abstract concepts of electricity visible, tangible, and meaningful to the students.

In this developmental phase, the present study focuses exclusively on the design and development of Scratch–Arduino educational tools and does not include an empirical classroom implementation or the assessment of student learning outcomes. The effectiveness of the tools in supporting students' concretization of abstract electrical concepts will be examined in the next phase of the project through a combination of (a) rubric-based analysis of students' written work during guided learning activities, (b) short pre- and post-tests aimed at capturing conceptual change, and (c) concept-mapping tasks designed to reveal shifts in students' knowledge structures. These procedures will allow for a systematic and multi-dimensional evaluation of the extent to which learners translate abstract theoretical constructs into concrete and meaningful representations once the tools are deployed in authentic classroom environments.

The design and programming processes followed five distinct but interconnected phases. User Interface (UI) design, Code and variable structure, Integration of the physical layer using Arduino, Feedback mechanisms, and Scalability and reusability.

UI Design

The interfaces were designed with a playful yet functional aesthetic to be understandable for students aged 11–15 years. The sprites represent real electrical components (batteries, resistors, switches, LEDs, and measuring instruments) with realistic colors and clear layouts. Each simulation includes active elements that respond to user actions through event and control blocks, providing direct interaction and visual understanding of the phenomena.

Code and Variable Structure

The code was organized around key variables that were updated in real time based on student actions or microcontroller inputs. These values control graphical phenomena, such as the electron flow speed or LED brightness, transforming abstract physical relationships into an immediate visual experience. Thus, learning becomes conceptual and experiential with a clear causal link between voltage, current, and resistance.

Arduino Integration (Physical Layer)

Scratch–Arduino connection is achieved through Scratch s3OneGpio extension, enabling bidirectional data flow. Digital signals from Scratch activate physical outputs (e.g., LEDs), whereas inputs from buttons or sensors modify the values in Scratch. This closed loop between the physical and digital environments creates a unified learning space in which students see that their actions produce real physical effects, strengthening the connection between theory and practice.

Feedback Mechanisms

Feedback is primarily conceptual and functional. The simulations dynamically render the operation of instruments and circuits through visual and auditory cues. For example, students can observe that the ammeter (in series) carries the entire current, the voltmeter (in parallel) carries very little current, and the ohmmeter is active only when the circuit is open. These variations, depicted visually, allow learners to conclude observations and reflections, thus enhancing metacognitive awareness (Hattie and Timperley, 2007; Shute, 2008).

Scalability and Reusability

The tools were developed as reusable templates, allowing teachers to modify variables, add new sprites, or incorporate additional circuits. Scratch 3 was chosen because of its open-source nature, cross-platform compatibility, and alignment with pedagogical approaches that foster creativity and inquiry-based learning (Resnick et al. 2009).

Ohm's Law Simulation

The development of a circuit simulation tool for teaching Ohm's law enables students to explore the relationship between the voltage (V), current (I), and resistance (R). Students can vary the resistance of the rheostat, which is measured using an ohmmeter in both analog and digital form (in Ω). They can also adjust the DC power supply voltage using buttons labeled (+1V), (+0.1V), (−1V), and (−0.1V). The current intensity was automatically calculated and displayed analogically via moving gauge needles and digitally on screens or indicators. When the circuit is powered, electron flow becomes visible along the conductors and instruments (voltmeter and ammeter). When the circuit is unpowered or set to zero voltage, the ohmmeter function is activated, allowing measurement of the resistance of the rheostat, while a small electron flow is shown through the rheostat and ohmmeter. Figure 1 shows two interface windows of the digital tool, where in case (a) the circuit is not powered, and in case (b) it is powered.

The speed of electron flow varies according to the current intensity, which affects the readings of the instruments. This interactive approach engages students by allowing them to manipulate voltage and resistance through a Scratch interface and by immediately observing the consequences. Thus, learning becomes hands-on and active, rather than passive. Guided inquiry approaches supported by interactive simulations have been shown to improve students' understanding of Ohm's law and related electrical relationships (Korsunsky & Agarwal, 2013). The software effectively transforms the theoretical formula I = V/R into an exploratory "game," maintaining student interest through direct visual feedback: needle movements, LED brightness changes, and dynamic current flow help students correlate actions with outcomes. These visual analogies make abstract quantities concrete, aiding beginners in safely and realistically grasping key electrical principles.

The inclusion of physical computing through an Arduino UNO bridges virtual and physical experiments, allowing students to observe and control real hardware in sync with simulations (Figure 2). In the first configuration, the circuit consists of three LEDs connected to Arduino digital pins 3, 5, and 10, which are controlled through simulation using Pulse-Width Modulation. Each LED represents an electrical component.

- LED on pin 3 → Current (I)
- LED on pin 5 → Voltage (V)
- LED on pin 10 → Resistance (R)

The brightness of each LED changes proportionally to its variable value, creating visual synchronization between the physical circuit and the digital display. For instance, the

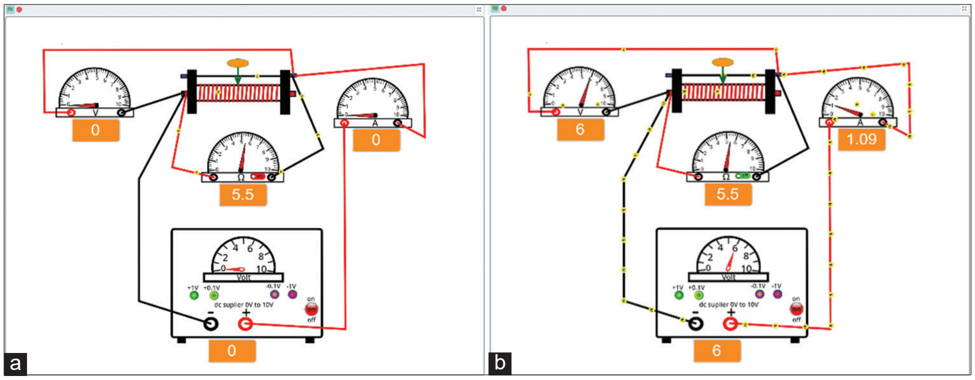


Figure 1: Scratch interface for Ohm's law simulation (visualization of relationships between voltage, current, and resistance with animated electron flow). (a) The supply voltage was 0 V and the ohmmeter measured the resistance. (b) Voltage-powered circuit

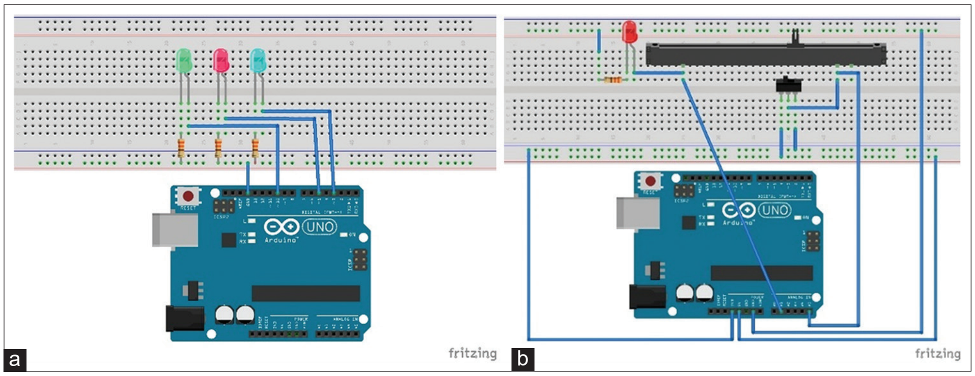


Figure 2: Hardware for the physical computing implementation of Ohm's law simulation: (a) hardware controlled by the simulation, and (b) hardware controlling the simulation

"current" LED glows brighter as simulated current increases, while the "resistance" LED dims as R increases. This multisensory feedback connects the screen observations to physical phenomena.

In the second configuration, hardware controls the simulation. The student adjusted a real rheostat (potentiometer) connected to the analog input A1; its value was transmitted to Scratch, dynamically altering the virtual resistance in the simulation. Another voltage selector switch, connected between 5V and 3.3V through analog input A5, allows for quick switching of the supply voltage. Changes 3.3V to 5V are reflected instantly in Scratch simulation, influencing both the virtual voltmeter and the current display. This dual setup, hardware ↔ software, highlights the educational power of physical computing, in which experimentation, coding, and observation merge into one experience.

Through these activities, students manipulate physical variables such as voltage and resistance, and observe digital responses such as current changes, LED brightness, and instrument readings. In doing so, they understood how technology bridges theory and reality while developing skills in experimentation, programming, and data interpretation, and integrating physics, technology, and computer science into a unified.

Two- and Three-Way (Aller-Retour) Switch Circuits

The development of simulation tools for aller-retour circuits with two and three switches allows students to explore the structure of the lighting circuits used to control a lamp or load from multiple locations (such as stairways or hallway lights). Digital tools simulate household lighting circuits, where illumination can be switched on or off from different independent locations: two single-pole double-throw (SPDT) switches for a two-way circuit, or two SPDT switches plus one double-pole double-throw (DPDT) intermediate switch for a three-way circuit. These simulations help students understand how such circuits function and how the arrangement of switches affects the current flow, enhancing both their conceptual and practical knowledge.

A particularly pedagogically valuable feature is the simulation of electric current as moving electrons. This visual representation helps students grasp the concept of the current flowing through a closed circuit, making the invisible process visible and intuitive. Figure 3 shows two interface windows of the digital tool. In case (a), the position of the switches creates a closed circuit that powers the lamp, which lights up. In contrast, in case (b), the circuit is open, and the lamp remains off.

Figure 4 shows two interface windows of the digital tool, where in case (a) the position of the switches is such that the circuit is closed and powers the lamp, which lights up, while in case (b) the circuit is open, and the lamp does not light up.

The simulation allowed students to click on each switch and immediately observe its effect on the lamp (on or off) and the direction of current flow. Independent control, meaning that the lamp can be switched off by either switch, regardless of which one turns it on, makes the activity particularly instructive. This feature reinforces students' understanding of toggle-switch logic and the functioning of the intermediate (crossover) switches used in real electrical systems.

In the three-switch version, the students experimented with three switches that controlled a single lamp. By clicking on any switch, they see that the lamp responds in real time, illuminating when the circuit is closed, and turning off when open. This interactive experimentation promotes the comprehension of multi-switch configurations and the concept of parallel control points in household circuits.

Each switch sprite has two costumeumes corresponding to its two positions (up/down or on/off). Clicking on a switch changes its costume and triggers an event that updates the circuit state. The circuit sprite continuously monitors the state of all the switches and updates the lamp and background accordingly. "Electron" sprites are cloned and animated using glide commands to simulate continuous current flow; when the circuit opens, clones disappear, stopping the visual flow. This real-time animation helps students to link circuit logic to visual outcomes, fostering an understanding of current continuity

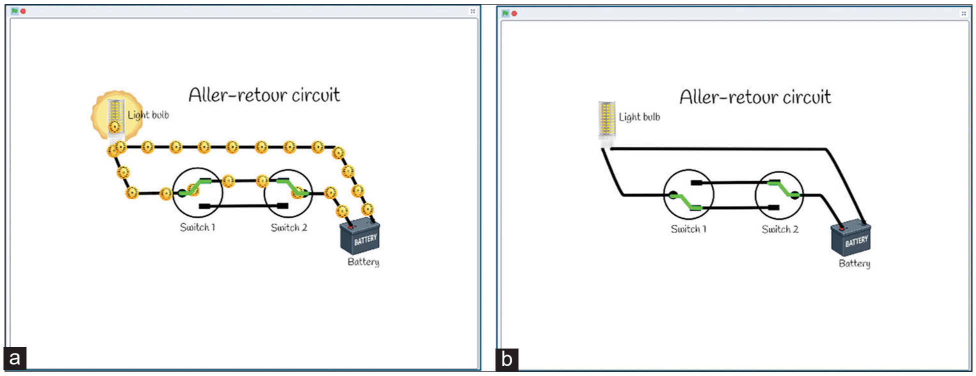


Figure 3: Scratch simulation of two-way (aller-retour) switch circuit. (a) Current path when the lamp is on. (b) No current flow when the lamp was turned off

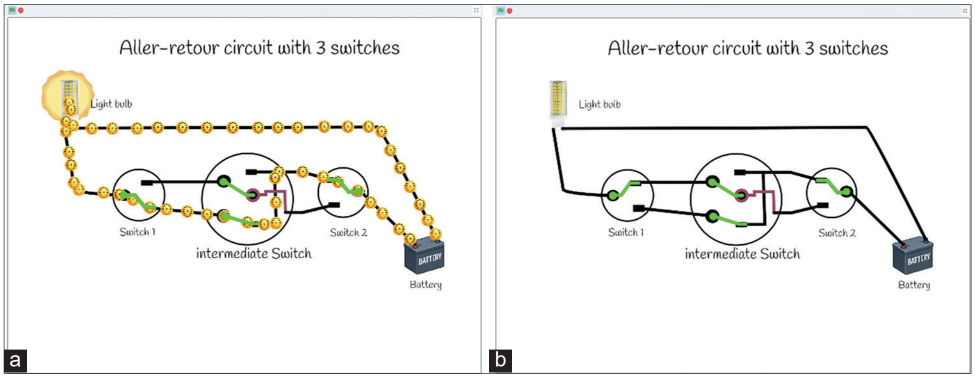


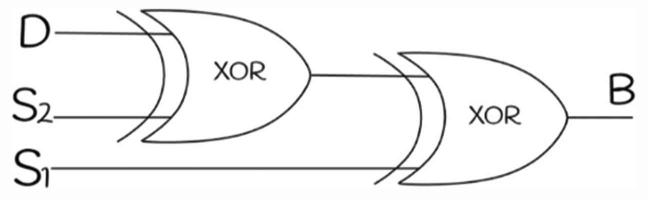
Figure 4: Scratch simulation of three-way (aller-retour) switch circuit. (a) Current path when the lamp was on. (b) Current path when the lamp was turned off

and logic switching.

The core of the program is the Boolean logic, which controls the lamp state. With the two SPDT switches, there are four possible position combinations (2^2). The lamp lights up when both switches are either in position “0” or both in position “1”, corresponding to an XNOR logic gate, $B = S_1$ XNOR S_2.

S_1=SPDT1	**S_2=SPDT$_2$**	**B=Bulb**
0	0	1
0	1	0
1	0	0
1	1	1

For the three-switch case (two SPDT switches and one DPDT switch), lamp state (B) follows the Boolean relationship $B = S_1$ XOR (S_2 XOR D), where D represents the DPDT switch. These expressions illustrate how the physical switch states are related to logical programming.

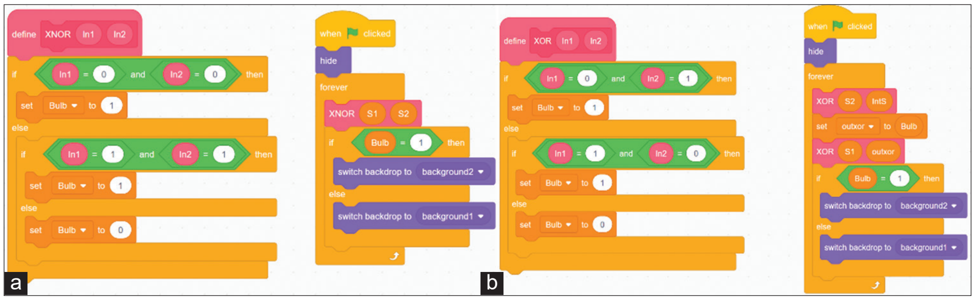


S_1=SPDT1	**S_2=SPDT2**	**D=DPDT**	**B=Bulb**
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Next to the table, the circuit diagram of the digital gates that implement the truth table is displayed.

As shown in Figure 5, the command structure for turning the lamp on and off is presented both in case 1 and in case 2, for better understanding and ease of implementation.

To combine the simulation with real-time hardware



```
define XNOR (In1) (In2)
if <<(In1) = (0)> and <(In2) = (0)>> then
    set [Bulb v] to (1)
else
    if <<(In1) = (1)> and <(In2) = (1)>> then
        set [Bulb v] to (1)
    else
        set [Bulb v] to (0)
```

```
when flag clicked
hide
forever
    XNOR (S1) (S2)
    if <(Bulb) = (1)> then
        switch backdrop to (background2 v)
    else
        switch backdrop to (background1 v)
```

```
define XOR (In1) (In2)
if <<(In1) = (0)> and <(In2) = (1)>> then
    set [Bulb v] to (1)
else
    if <<(In1) = (1)> and <(In2) = (0)>> then
        set [Bulb v] to (1)
    else
        set [Bulb v] to (0)
```

```
when flag clicked
hide
forever
    XOR (S2) (IntS)
    set [outxor v] to (Bulb)
    XOR (S1) (outxor)
    if <(Bulb) = (1)> then
        switch backdrop to (background2 v)
    else
        switch backdrop to (background1 v)
```

Figure 5: Structure of control commands for lamp switching: (a) two-switch circuit and (b) three-switch circuit

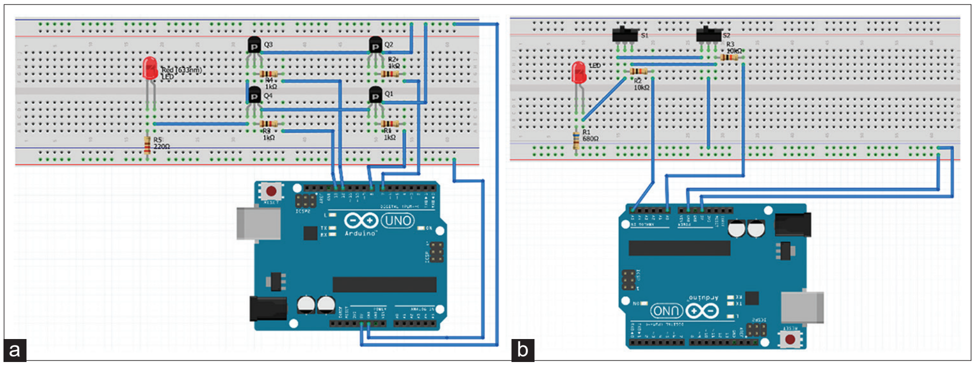


Figure 6: Hardware for the physical computing implementation of the two-way (aller-retour) circuit simulation: (a) hardware controlled by the simulation and (b) hardware controlling the simulation

interaction, aller-retour circuits were implemented using Arduino. Two complementary setups were developed in this study.

In the first setup (Figure 6a), the circuit includes one LED and four transistors (Q1–Q4). Transistors Q1 and Q2 act as SPDT1, whereas Q3 and Q4 represent SPDT2. The switching of the transistors was controlled using Arduino digital pins 7, 8, 12, and 13 under the command of Scratch simulation. The LED turns on or off in sync with the simulation, providing multisensory feedback that connects the on-screen observation with a real physical phenomenon.

In the second setup (Figure 6b), the hardware controlled the simulation. The student toggles two real switches connected to Arduino's analog inputs A0 and A5. The input values are transmitted to Scratch and dynamically alter the states of the two simulated SPDT switches.

This bidirectional feedback loop (hardware ↔ simulation) exemplifies the full potential of physical computing in educational practices, reinforcing the interplay between programming, experimentation, and observation.

ON–OFF Switch Circuits for LED Control

The development of simulation tools for ON–OFF switch circuits, connected either in series or in parallel, allows students to change the state of each switch (ON or OFF), thereby controlling the lighting of two or three LEDs. Depending on the switch combination, they can illuminate two or three LEDs, thereby understanding the conductive or nonconductive paths created by the opening and closing switches. In this way, learners discover how each LED's activation depends on the configuration of the switches, gaining a practical understanding of the series and parallel circuit behavior.

To visualize the current flow, the first simulation used small-arrow sprites placed along circuit lines. Each arrow sprite contains costumeumes with different arrow orientations, and the costumeume alternation produces a perception of motion that corresponds to the conventional current direction. These arrows remained hidden until the current was detected in the branch at which they appeared and animated. Each arrow sprite had its own independent script.

Figure 7 shows two interface windows of the digital tool with two LEDs, where in case (a) the position of the switches is such that the circuit is closed and powers the two LEDs, which light up, while in case (b) the circuit is open and both LEDs are off.

In the two-LED simulation (Figure 7a), when both switches are closed, the circuit is complete, and both LEDs light up. When any switch opens (Figure 7b), the circuit breaks, and the LEDs go out. Students can test all combinations and observe the effects of switching positions on current paths and LED behavior.

In the second simulation, two visualizations of the current flow were available. Arrow sprites moving along the wires (conventional current direction), or Particle sprites ("electrons") move in the opposite direction to conventional flow. Students can choose between these two modes of representation, reinforcing their conceptual understanding

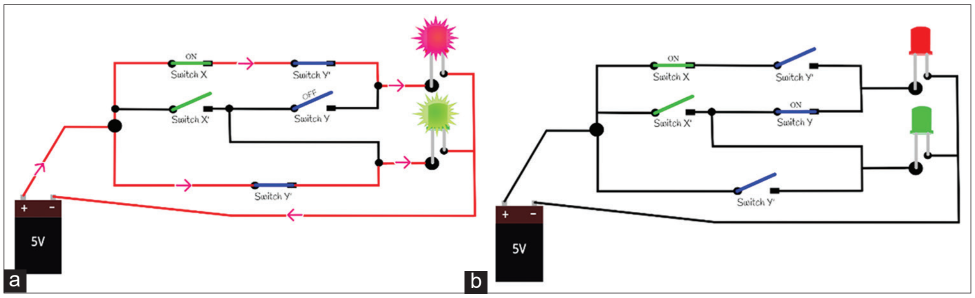


Figure 7: Scratch simulation for controlling two LEDs showing current paths when the LEDs are ON (a) and when all LEDs are OFF (b)

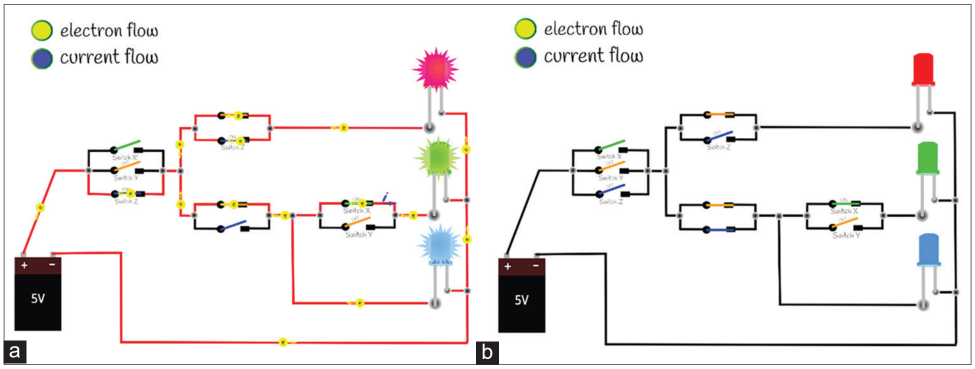


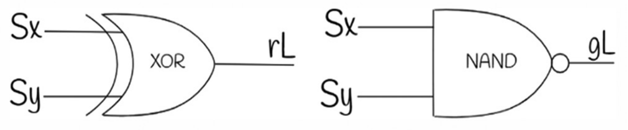
Figure 8: Scratch simulation for controlling three LEDs showing current paths when certain LEDs are ON (a) and when all LEDs are OFF (b)

of both the current conventions. These two projects form a progressive learning sequence: the first (two LEDs) introduces basic ideas, and the second (three LEDs) adds complexity to deeper comprehension and challenges. This gradual increase in difficulty benefits instructional design by allowing teachers to begin with simple cases and extend their exploration as students become more confident. Figure 8 shows two interface windows of the digital tool with three LEDs.

In the three-LED simulation, with three switches controlling three LEDs, students analyzed more complex interactions and logical relationships that determined each LED's state. An important part of the program structure, as in the aller-retour simulation, is the Boolean logic, which determines whether each LED is ON or OFF based on the active switch states. Among the two LED circuits, the red LED (rL) behaves as the XOR of the two switches, whereas the green LED (gL) behaves as the NAND of the two switches. Thus, rL = S_x XOR S_y and gL = S_x NAND S_y.

Sx	Sy	rL	gL
(OFF) 0	(OFF) 0	0	1
(OFF) 0	(ON) 1	1	1
(ON) 1	(OFF) 0	1	1
(ON) 1	(ON) 1	0	0

Figure 9 shows the instruction structure for turning on and off the two LEDs.



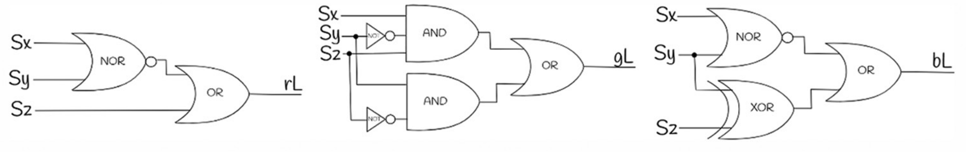
For the three LED circuits, the combinations were expanded to 2^3 = eight possibilities. Using Boolean algebra, the relationships between the three LEDs were derived as:

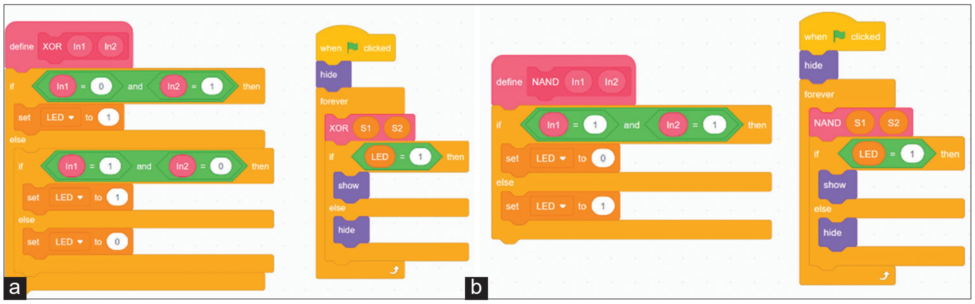
Sx	Sy	Sz	rL	gL	bL
(OFF) 0	(OFF) 0	(OFF) 0	1	0	1
(OFF) 0	(OFF) 0	(ON) 1	1	0	1
(OFF) 0	(ON) 1	(OFF) 0	0	1	1
(OFF) 0	(ON) 1	(ON) 1	1	0	0
(ON) 1	(OFF) 0	(OFF) 0	0	0	0
(ON) 1	(OFF) 0	(ON) 1	1	1	1
(ON) 1	(ON) 1	(OFF) 0	0	1	1
(ON) 1	(ON) 1	(ON) 1	1	0	0

rL = Sz OR (Sx NOR Sy),

gL= (Sy AND (NOT Sz)) OR (Sx AND (NOT Sy) AND Sz) και

bL = (Sx NOR Sy) OR (Sy XOR Sz)





a

```
define XOR In1 In2
if <(In1) = (0) and (In2) = (1)> then
    set LED to 1
else
    if <(In1) = (1) and (In2) = (0)> then
        set LED to 1
    else
        set LED to 0
```

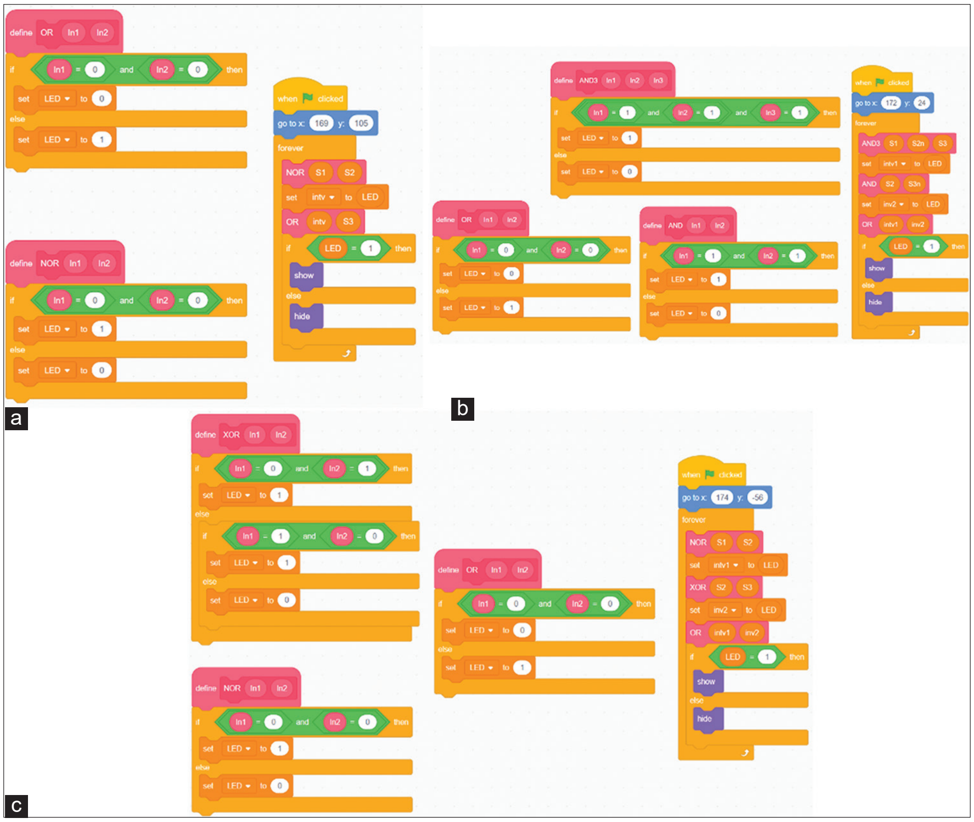
```
when clicked
hide
forever
    XOR S1 S2
    if <(LED) = (1)> then
        show
    else
        hide
```

b

```
define NAND In1 In2
if <(In1) = (1) and (In2) = (1)> then
    set LED to 0
else
    set LED to 1
```

```
when clicked
hide
forever
    NAND S1 S2
    if <(LED) = (1)> then
        show
    else
        hide
```

Figure 9: Command structure for turning LEDs on and off: (a) red LED and (b) green LED



a

```
define OR In1 In2
if <(In1) = (0) and (In2) = (0)> then
    set LED to 0
else
    set LED to 1
```

```
define NOR In1 In2
if <(In1) = (0) and (In2) = (0)> then
    set LED to 1
else
    set LED to 0
```

```
when clicked
go to x: 169 y: 105
forever
    NOR S1 S2
    set intv to LED
    OR intv S3
    if <(LED) = (1)> then
        show
    else
        hide
```

b

```
define AND3 In1 In2 In3
if <(In1) = (1) and (In2) = (1) and (In3) = (1)> then
    set LED to 1
else
    set LED to 0
```

```
define OR In1 In2
if <(In1) = (0) and (In2) = (0)> then
    set LED to 0
else
    set LED to 1
```

```
define AND In1 In2
if <(In1) = (1) and (In2) = (1)> then
    set LED to 1
else
    set LED to 0
```

```
when clicked
go to x: 172 y: 24
forever
    AND3 S1 S2n S3
    set intv1 to LED
    AND S2 S3n
    set inv2 to LED
    OR intv1 inv2
    if <(LED) = (1)> then
        show
    else
        hide
```

c

```
define XOR In1 In2
if <(In1) = (0) and (In2) = (1)> then
    set LED to 1
else
    if <(In1) = (1) and (In2) = (0)> then
        set LED to 1
    else
        set LED to 0
```

```
define NOR In1 In2
if <(In1) = (0) and (In2) = (0)> then
    set LED to 1
else
    set LED to 0
```

```
define OR In1 In2
if <(In1) = (0) and (In2) = (0)> then
    set LED to 0
else
    set LED to 1
```

```
when clicked
go to x: 174 y: -56
forever
    NOR S1 S2
    set intv1 to LED
    XOR S2 S3
    set inv2 to LED
    OR intv1 inv2
    if <(LED) = (1)> then
        show
    else
        hide
```

Figure 10: Command structure for turning LEDs on and off: (a) red LED, (b) green LED, and (c) blue LED

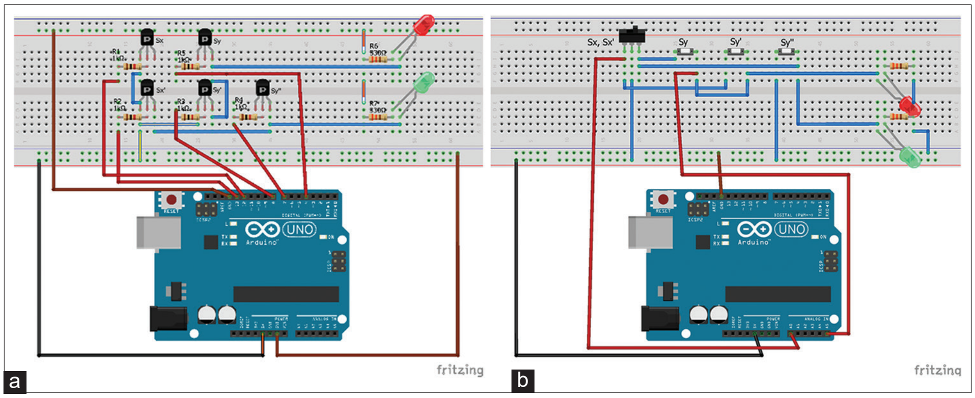


Figure 11: Hardware for the physical computing implementation of the two-LED circuit simulation: (a) hardware controlled by the simulation and (b) hardware controlling the simulation

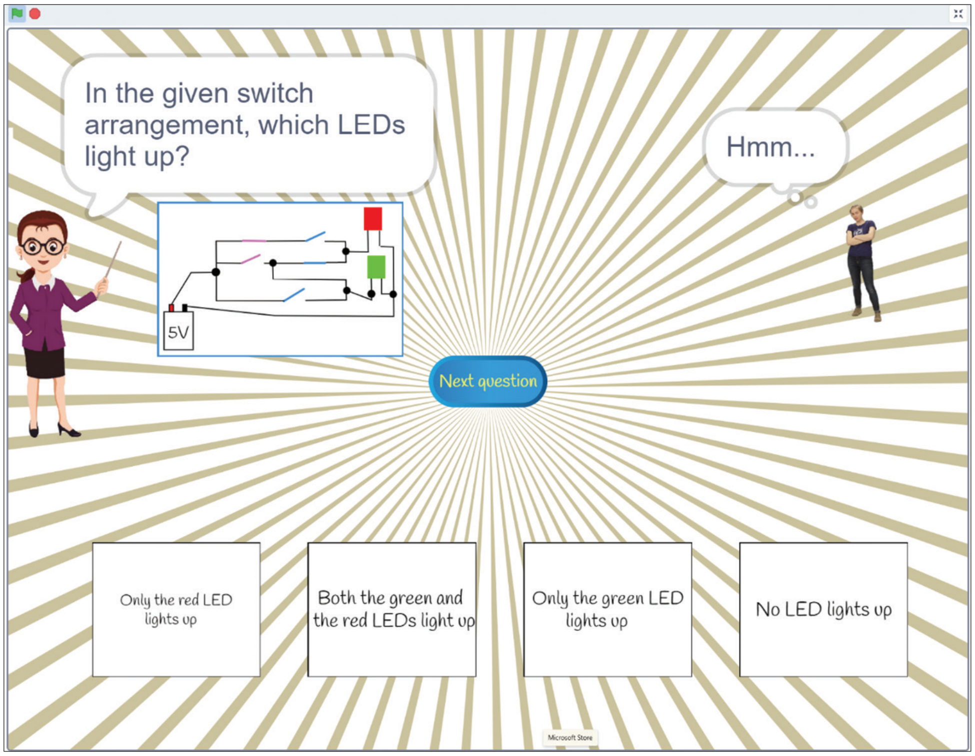


Figure 12: Interactive multiple-choice quiz module

Figure 10 shows the instruction structure for turning on and off the three LEDs:

As with the previous simulations, the ON–OFF switch circuit simulations were extended to include physical computing using Arduino, allowing real-time synchronization between the physical circuit and Scratch environment.

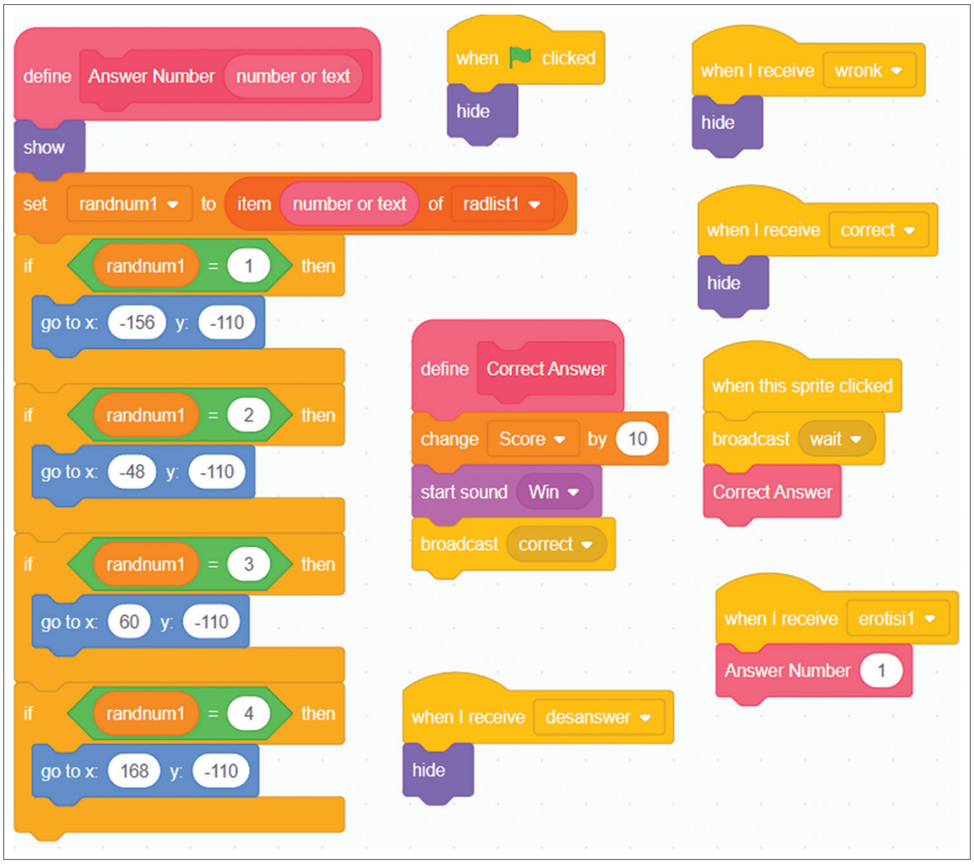
Two complementary hardware setups were implemented for the two LED circuits: one in which Scratch simulation controlled the physical circuit, and one in which the physical circuit controlled the simulation. In the first setup (Figure 11a), the circuit consisted of two LEDs and five transistors acting as switches. The transistors were controlled using Arduino digital pins 12, 13, 4, 8, and 7, which received instructions from Scratch simulation. LEDs blink according to simulated behavior, providing multisensory feedback that links on-screen actions to a real physical response. In the second setup (Figure 11b), the hardware controls the simulation, and the student toggles two physical switches connected to Arduino's analog inputs A0 and A5. The analog input values were transmitted to Scratch, and the states of the simulated switches were dynamically modified. This two-way control loop (hardware ↔ simulation) demonstrates the educational potential of physical computing, bridging conceptual learning, programming, and real-world experimentation.

ASSESSMENT TOOLS

An example of the interactive multiple-choice quiz interface, including the visual layout, student–teacher characters, and response options, is shown in Figure 12. Interactive assessment tools serve as formative instruments that function independently or in conjunction with other educational simulations. They complete the cycle "simulation – interaction – evaluation." Two distinct types of interactive tests have been developed (Pellegrino et al., 2001). Diagnostic and formative assessment approaches, particularly those using structured multiple-choice and interactive formats, have been shown to support conceptual diagnosis and learning progression in science education (Briggs et al., 2006).

- A multiple-choice quiz, and
- Matching text–image quiz.

Both formats promote active learning, self-assessment, and metacognitive reflection while also helping teachers evaluate students' conceptual understanding. The multiple-choice test offers an engaging method for students to assess their grasp of the core electrical concepts introduced in simulations. The matching test builds on the dual-coding theory (Paivio, 2014), according to which the simultaneous presentation of verbal and visual information activates two different cognitive systems (verbal and visual-spatial), thereby improving memory and conceptual understanding. Research has shown that combining text and images supports comprehension, particularly among learners with diverse learning styles. Visual elements can make complex concepts more accessible and foster long-term retention (Georganta, 2019). The design of both assessment tools adhered to instructional design principles emphasizing interaction, experiential learning, and critical thinking while ensuring usability and accessibility for students aged 11–15 years.



```
define Answer Number (number or text)
show
set randnum1 to (item (number or text) of radlist1)
if <(randnum1) = (1)> then
    go to x: (-156) y: (-110)
if <(randnum1) = (2)> then
    go to x: (-48) y: (-110)
if <(randnum1) = (3)> then
    go to x: (60) y: (-110)
if <(randnum1) = (4)> then
    go to x: (168) y: (-110)

when flag clicked
hide

when I receive wronk
hide

when I receive correct
hide

define Correct Answer
change Score by (10)
start sound Win
broadcast correct

when this sprite clicked
broadcast wait
Correct Answer

when I receive erotisi1
Answer Number (1)

when I receive desanswer
hide
```

Figure 13: Example of command structure for one of the 40 answer sprites (here, the correct answer to Question 1)

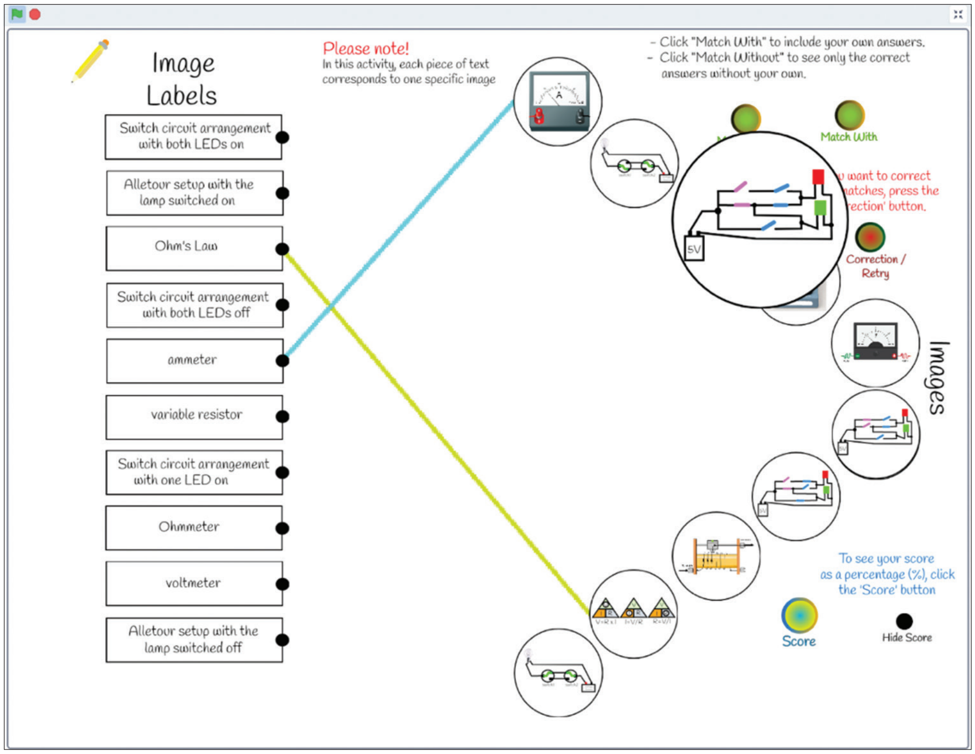


Image Labels

- Switch circuit arrangement with both LEDs on
- Alletour setup with the lamp switched on
- Ohm's Law
- Switch circuit arrangement with both LEDs off
- ammeter
- variable resistor
- Switch circuit arrangement with one LED on
- Ohmmeter
- voltmeter
- Alletour setup with the lamp switched off

Please note!
In this activity, each piece of text corresponds to one specific image

- Click "Match With" to include your own answers.
- Click "Match Without" to see only the correct answers without your own.

u want to correct atches, press the ection' button.

To see your score as a percentage (%), click the 'Score' button

Figure 14: Interactive quiz module using text–image matching method

Multiple-Choice Quiz

The multiple-choice quiz allowed the students to evaluate their knowledge and understanding of the key electrical concepts covered in the simulations. Questions appear sequentially above a teacher's cartoon figure, whereas the student character introduces each question and reacts to the answers. The order of questions and answers was changed for each test run to ensure variability. Each question (which some accompanied by images) offered four answer options, one of which was correct. Correct and incorrect answers trigger different sound effects, and the system automatically scores students' performance.

The project consisted of 44 sprites: four represented the teacher, the student, the "Start Test" button, and the "Next Question" button. Forty sprites represented possible answer choices (10 questions × four answers). Two backdrops were used: one for the start screen and another for the active quiz session. This modular design allows educators (or even students) to add or modify questions and answers easily by editing the script, thus turning the activity into an opportunity for creative engagement through coding. Each answer option is a separate sprite that "knows" whether it is correct or incorrect. When clicked, it sends the message "Correct" or "Incorrect," prompting the corresponding visual and auditory feedback. The code of each sprite-answer is shown illustratively in Figure 13, specifically, it is answer 1 of question 1, which is also the correct answer to question 1.

Text–Image Matching Quiz

The structure and functionality of the text–image matching assessment tool are illustrated in Figure 14. The text–image matching tool allows students to evaluate their understanding by matching text labels with the corresponding images. This activity requires learners to connect items from two columns – descriptions on one side and visual representations on the other—thus promoting active recall and conceptual association rather than rote memorization. The pedagogical value of this tool lies in its interactivity and randomized layout. Every new session displayed text and image elements in a different order, preventing memorization and maintaining engagement across multiple attempts. An additional feature enlarges any image when the mouse pointer hovers nearby, aiding in selection and accessibility.

The project uses 27 sprites and two backdrops, organized as follows: ten text sprites representing descriptive terms or definitions, ten image sprites depicting circuit components or diagrams related to the electricity topics covered in the simulations, six button-type sprites for actions such as checking answers or restarting, and one utility sprite to draw connecting lines between matched pairs and showing the correct associations when requested. The two backdrops serve distinct purposes: as a layout reference during development, and as the main test interface. These interactive tests reinforce knowledge and promote self-regulated learning by encouraging reflection, repetition, and immediate feedback. The combination of visual cues, interactivity, and assessment supports cognitive integration of theoretical knowledge acquired through simulations, thereby closing the educational loop between exploration and evaluation.

CONCLUSIONS

This paper presents the design and development of an integrated suite of Scratch–Arduino digital tools for teaching fundamental concepts of electricity. The design philosophy of these tools is based on the following three pillars.

Constructionist Learning (Constructionism)

Students learn by designing, testing, and discovering through active interactions with their environment. Scratch functions as an "ideas laboratory (Papert, 1980), where programming directly connects to observable physical outcomes. Through the iterative process of design–predict–test–reflect, learners cultivate both scientific reasoning and computational thinking (Wing, 2006).

Visualization and Conceptual Understanding

These tools transform invisible quantities – current, voltage, and resistance – into tangible animated representations. The motion of electrons, the brightness of LEDs, or the switching of components act as visual analogies that make abstract relationships explicit and intuitive, lowering cognitive barriers to understanding.

Feedback and Self-Regulation

Built-in feedback mechanisms enable students to self-correct and develop metacognitive skills (Hattie and Timperley, 2007; Shute, 2008). Explanations embedded within both the simulations and assessment tests function as integrated guidance, promoting reflection, and conceptual reinforcement.

Beyond their high computational, graphical, and interactive qualities, these tools incorporate physical computing and establish a two-way bridge between the digital and physical domains. This dual interactivity enables learners to experience science as a process of experimentation, programming, and inquiry, and not merely as an abstract theory.

Scratch–Arduino educational suite represents an open, extensible framework that is adaptable to new topics (e.g., electromagnetism, energy, and mechanics), offering low implementation costumes and high pedagogical value.

As the focus of the present paper is the development of the digital tools themselves, the study does not report on students' degree of concretization of abstract concepts. This outcome-based evaluation forms the next step of the research, during which the tools will be implemented in real classroom settings and examined systematically through the methodologies outlined above.

Future research should focus on evaluating the effectiveness of these tools through controlled studies and developing an online repository in which educators can download, adapt, and share simulations, fostering collaboration, accessibility, and continuous improvement.

REFERENCES

Adams, W.K. (2010). Student engagement and learning with interactive PhET simulations. Proceedings of the 2010 Physics Education Research Conference 1289, 33-36.

Ainsworth, S. (2022). A systematic review on combining real and virtual experiments in science education. Review of Educational Research, 92(5), 749-782.

Banda, A., & Nzabahimana, J. (2021). Effectiveness of PhET simulations on students' conceptual understanding in physics: A meta-analysis. European Journal of Science and Mathematics Education, 9(3), 106-122.

Barrera, M., Rodríguez, J., & García, F. (2019). Using Arduino to enhance STEM learning in secondary education: A practical approach. International Journal of Educational Technology in Higher Education, 16(1), 12-25.

Blikstein, P. (2013). Digital fabrication and "making" in education: The democratization of invention. In: Walter-Herrmann J., & Büching, C., (Eds.), FabLabs: Of Machines, Makers and Inventors. Germany: Transcript Verlag, pp. 1-21.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In: Proceedings of the American Educational Research Association (AERA), Vancouver, Canada.

Briggs, D.C., Alonzo, A.C., Schwab, C., & Wilson, M. (2006). Diagnostic assessment with ordered multiple-choice items. Educational Assessment, 11(1), 33-63.

De Jong, T. (2006). Computer simulations: Technological advances in inquiry learning. Science, 312(5773), 532-533.

De Jong, T., & van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. Review of Educational Research, 68(2), 179–201.

Design-Based Research Collective. (2003). Design-based research: An emerging paradigm for educational inquiry. Educational Researcher, 32(1), 5-8.

Engelhardt, P.V., & Beichner, R.J. (2004). Students' understanding of direct current resistive electrical circuits. American Journal of Physics, 72(1), 98-115.

Georganta, E. (2019). The power of animation and educational video in promoting learning. International Conference on Open and Distance Education, 10, 1-11.

Govender, R.G., Govender, N., & Govender, D.W. (2021). A physical computing approach to introducing programming. African Journal of Research in Mathematics, Science and Technology Education, 25(2), 149-160.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. Educational Researcher, 42(1), 38-43.

Hattie, J., & Timperley, H. (2007). The power of feedback. Review of Educational Research, 77(1), 81-112.

Kafai, Y. B., & Burke, Q. (2009). Constructionist gaming: Understanding the benefits of making games for learning. Educational Psychologist, 44(4), 313–334.

Kafai, Y.B., & Burke, Q. (2015). Connected Code: Why Children Need to Learn Programming. 2nd ed. Cambridge: MIT Press.

Korsunsky, S., & Agarwal, A. (2013). Teaching Ohm's Law with simulations and guided inquiry. Physics Education, 48(3), 317-322.

Linn, M.C., & Eylon, B.S. (2011). Science Learning and Instruction: Taking Advantage of Technology to Promote Understanding. London: Routledge.

Lye, S.Y., & Koh, J.H.L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K–12? Computers in Human Behavior, 41, 51-61.

Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004). Scratch: A sneak preview. In: Second International Conference on Creating, Connecting and Collaborating through Computing, pp. 104-109.

Marín-Marín, J.A., García-Tudela, P.A., & Terrón, P.D. (2024). Integrating Arduino-based projects to foster systems thinking in STEM education. Education and Information Technologies, 10, e29177.

Martín-Gutiérrez, J., Mora, C.E., Añorbe-Díaz, B., & González-Marrero, A. (2021). Integrating Arduino-based projects in STEM education: A case study in middle school. Education and Information Technologies, 26(1), 1021-1040.

McDermott, L.C., & Shaffer, P.S. (1992). Research as a guide for curriculum development: An example from introductory electricity. American Journal of Physics, 60(11), 994-1003.

Ntourou, K., Papadakis, S., & Kalogiannakis, M. (2021). Arduino-based activities for developing computational thinking in secondary education. Education and Information Technologies, 26(3)..

Olympiou, G., & Zacharia, Z.C. (2012). Combining real and virtual experiments in physics education: A meta-analysis. Journal of Science Education and Technology, 21(1), 6-25.

Paivio, A. (2014). Mind and its Evolution: A Dual Coding Theoretical Approach. New York: Psychology Press.

Papert, S. (1980). Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books.

Pellegrino, J. W., Chudowsky, N., & Glaser, R. (2001). Knowing what students know: The science and design of educational assessment. Washington, DC: National Academy Press.

Perkins, K., Adams, W., Dubson, M., Finkelstein, N., Reid, S., & Wieman, C. (2006). PhET: Interactive simulations for teaching and learning physics. The Physics Teacher, 44(1), 18-23.

Resnick, M., Maloney, J., Hernández, A.M., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. Communications of the ACM, 52(11), 60-67.

Rutten, N., van Joolingen, W.R., & van der Veen, J.T. (2012). The learning effects of computer simulations in science education. Computers and Education, 58(1), 136-153.

Sarı, A., & Yıldırım, A. (2022). Integrating Scratch and Arduino in secondary education: Effects on students' engagement and understanding. Computer Applications in Engineering Education, 30(4), 1045–1060.

Shipstone, D. (1984). A study of children's understanding of electricity in simple DC circuits. European Journal of Science Education, 6(2), 185-198.

Shute, V. J. (2008). Focus on formative feedback. Review of Educational Research, 78(1), 153–189.

Wieman, C. (2008). Why not try a scientific approach to science education? Change: The Magazine of Higher Learning, 40(5), 9–15.

Wing, J.M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Wörner, J., Ochsner, A., & Scheiter, K. (2022). Combining real and virtual experiments in physics education: A meta-analysis. Journal of Science Education and Technology, 31(2), 179-195.