

Blink and It's Gone

Description

This unit is created for students with a basic understanding of computer coding. Students will build and test a coding program to turn an LED light on and off using an Arduino Uno board. The students will connect the hardware to a breadboard, program the code using Arduino software, test the system, adapt variations in blinking times, evaluate their results, and share observations with their classmates.

Packet Contents

- [Introduction](#)
- [Curriculum Alignment](#)
- [Objectives](#)
- [Time and Location](#)
- [Teacher Materials](#)
- [Student Materials](#)
- [Safety](#)
- [Student Prior Knowledge](#)
- [Teacher Preparation](#)
- [Activities](#)
- [Assessment](#)
- [Critical Vocabulary](#)
- [Modifications](#)
- [Author Information](#)

Lesson Plan Tags

Check the standards that are met in your lesson plan, check all that apply.

Middle School High School 7th Grade Science Middle School Math
Other High School Science High School Math HS BFIT HS Technology

Introduction

"Blink and It's Gone" lesson discovers how computer programmers work to solve the challenges of systems. An example of this challenge is providing an automatic code for turning lights on and off. Students will work to set up and program an Arduino board to turn a light on and off an LED light at a 5 second on and off interval and also, a 5 second on and 2 second off interval. Each student will build the system, program and test it, reflect on the challenge, and present their experiences to their classmates.

Curriculum Alignment

Mathematical Practice Standards:

- Look for and make use of structure
- Make sense of problems and persevere in solving them
- Use appropriate tools strategically
- Attend to precision

CCSS.Math.Content.HSN.Q.A.2

Define appropriate quantities for the purpose of descriptive modeling.

CTE BD10 4.00

Understand the fundamentals of web design

Science

National Science Education Standards

CONTENT STANDARD A: Science as Inquiry

As a result of activities, all students should develop

- Abilities necessary to do scientific inquiry

CONTENT STANDARD B: Physical Science

As a result of their activities, all students should develop understanding of

- Interactions of energy and matter

CONTENT STANDARD E: Science and Technology

As a result of activities, all students should develop

- Abilities of technological design
- Understandings about science and technology

CONTENT STANDARD F: Science in Personal and Social Perspectives

As a result of activities, all students should develop understanding of

- Science and technology in local, national, and global challenges

Objectives

- Students will understand and use the engineering design process.
- Students will study circuits, computers, and software coding.
- Students will explain how engineering can help solve society's challenges.
- Students will practice appropriate problem solving strategies.

Time & Location

Location: Computer Lab

Time: 1 Class period – 90 minutes

Teacher Materials

- Student Resource Sheet
- Student Programming Reflection Sheet
- Computer
- Internet Access
- Arduino Uno Basic Kit
- Arduino software (downloaded to each computer)
- Chocolate (optional)
- Golden Nugget (ticket out the door)

Student Materials

- Student Resource Sheet
- Student Programming Reflection Sheet
- From the Arduino D Kit:
 - Arduino Board
 - 3 jump wires
 - USB Cord
 - Breadboard
 - 1 LED light (any color)
 - 1 Resistor (brown, orange, black, gold)

Safety

- No liquids or food around computer
- No horse playing
- Assemble the bread board before hooking USB to computer (prevents getting shocked)

Student Prior Knowledge

- Conversion of metric system
- Basic coding knowledge¹

¹ See modifications for students with no prior knowledge of this concept.

- Understand concept of a unit rate a/b with a ratio $a:b$ with b not equal 0, and use rate language in the context of a ratio relationship. (CCSS.Math Content 6.RP.A.2)

Teacher Preparations

Before activity:

- Download Arduino software to computers and drivers

Activities

1. Bell Work: 10 minutes

How are stoplights timed? Answer in 3-5 sentences.

Have a few students share their answers.

Student Answers may vary

According to: http://www.traffic-signal-design.com/how_do_traffic_signals_work.htm

Fixed Time: Under fixed time operation the traffic signals will display green to each approach for the same time every cycle regardless of the traffic conditions. This may be adequate in heavily congested areas but where a lightly trafficked side road is included within the sequence it is very wasteful if in some cycles there are no vehicles waiting as the time could be better allocated to a busier approach.

Vehicle Actuation: one of the most common modes of operation for traffic signals and as the name suggests it takes into account the vehicle demands on all approaches and adjusts the green time accordingly.

2. The Activity: 55 minutes

- **Say:** Today, we are going to look at programming your own light. This activity will require you to use the coding practice you have been working on and introduce a new platform, Arduino Uno. <Now pass out the kits>
- Hand out Student Resource Sheet
 - o Read over the Resource Sheet with students – this is the instructions for activity.
 - o Ask if there are any questions
- Working independently, have students follow the Resource Sheet to set up the LED circuit and program the LED light to blink.
- Circulate the room monitoring the students. Facilitate any questions but offer minimal assistance. Allow the students to work through frustrations and challenges.

- **Challenge 1:** once students have their LED light working and blinking according to the instructions on the Resource Sheet:
 - o **Say:** Now that you are getting your LED light to work, I have a challenge for you, make you LED light come on for 5 seconds then off for 5 seconds.
 - **Challenge 2:**
 - o **Say:** Now that you have your LED light coming on and off for 5 seconds, I want you to make you LED light come on for 5 seconds but off for 2 seconds.
- 3. Class Discussion: 5 minutes**
- Have a few students share their experience and thoughts about using the Arduino and programming.
 - Were they able to complete both challenges? One challenge?
- 4. Follow up: 15 minutes**
- Hand out Student Programming Reflection.
 - Have students complete for grade.
- 5. Evaluate: Golden Nugget (Ticket Out the Door) 5 minutes**
- One thing they liked
 - One thing they would change
 - o Turn it in going out the door for a piece of “gold” wrapped chocolate.

Assessment

- Teacher will evaluate Student Programming Reflection.

Critical Vocabulary

- Computer Programming: the process of developing and implementing a set of instructions that tells the computer how to complete certain tasks.
- Circuits: a closed path which an electric current flows
- Modeling: is the process of choosing and using appropriate mathematics and statistics to analyze empirical situations, to understand them better, and to improve decisions. From: <http://www.corestandards.org/Math/Content/HSM/>
- Quantity: a particular amount of something

Modifications

- For students requiring help, they may pair up (EC students)
- Students with no coding experience there is a page at the end that has coding practice. This should be delivered prior to this lesson.

Author Information

Kenan Fellow:

Brooke Woodard, Princeton High School, Johnston County Schools, Princeton
Grades 9-12, CTE Business

Teaching Experience: 6 years


































Email: BrookeWoodard@Johnston.k12.nc.us

Mentor:

- where you work
- about your work
- special certifications, degrees, experience, or other qualifications you'd like to share
- email

Student Resource Sheet

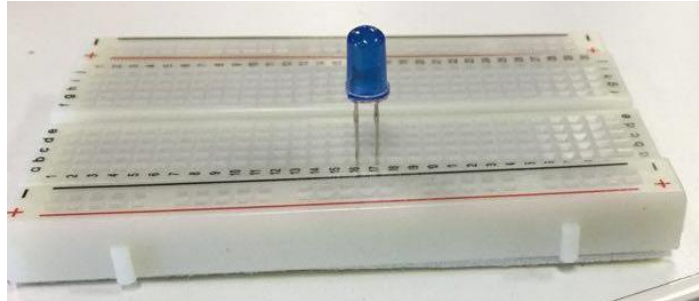
Introduction to Arduino Uno Components:

 Resistor 220 ohm	 Resistor 560 ohm	 Resistor 1k ohm	 Resistor 4.7K ohm	 Resistor 10K ohm	 Resistor 1M ohm
 Resistor 10M ohm	 Potentiometer 10k	 Capacitor 100 pF	 Capacitor 100 nF	 Capacitor 100 uF 25v polarized	 LED Red
 LED Green	 LED Blue	 LED Yellow	 LED WHITE	 LED RGB CC	 Breadboard Small
 Arduino uno	 LCD 16 x 2	 Pushbutton	 Diode	 NPN Transistor (BJT)	 N-MOSFET
 Photoresistor (LDR)	 Temperature Sensor [TMP36]	 Tilt Sensor 4-pin	 Piezo	 Optocoupler [4N35]	 H-bridge Motor Driver [L293D]
 DC Motor	 9V Battery	 Micro Servo			

Pictures obtained from Arduino Projects.

Step 1: Setting up the Breadboard

First, place your LED light on the breadboard. The LED light has a positive and negative end. The longer end is the positive side.



Now you are going to give your LED light power (however it will not come on yet) by connecting the breadboard to the Arduino Uno. In this step, you need your 3 jump wires and 1 resistor.

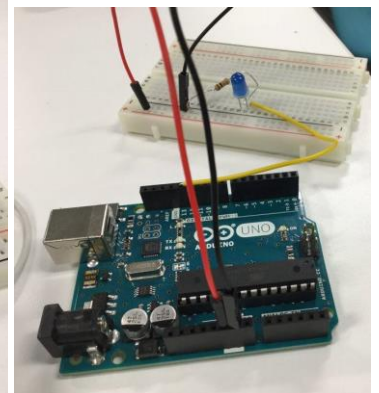
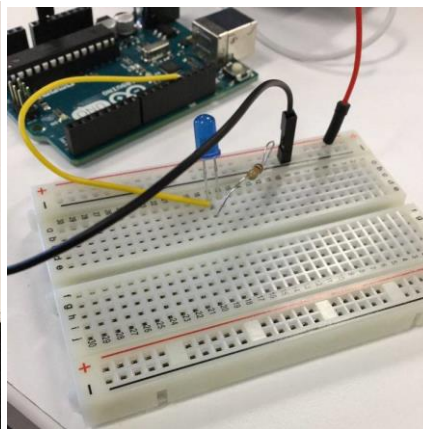
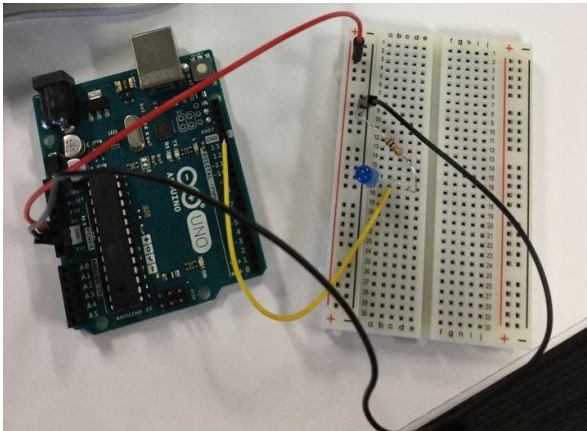
The wires will be known as positive wire, negative wire (ground wire), and digital wire (going from LED to Arduino Uno). The positive wire connects the breadboard to the Arduino Uno voltage pin to give power. The negative wire is your ground wire or no charge. The digital wire will power on the LED light. Using a resistor between the LED light and power will keep the bulb from burning out quickly.

Place the positive wire in any hole in the + column on the a-e side of the breadboard. Then place the other end of this wire in the 5V pin on the Arduino Uno board.

Next, place the ground wire in any hole in the - column on the a-e side of the breadboard. Then place the other end of the wire in the GND pin on the Arduino Uno board.

Then, place one end of the resistor in the same row as the negative leg of the LED light and the other end of the resistor plugged into any part in the ground wire column (- column).

Finally, place your last wire, the “digital wire” into the row with the positive leg of the LED light and the other end in pin 13 on the Arduino Uno board.



Step 2: Connecting the Board

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should go on.

Step 3: Launch the Arduino application

Double-click the Arduino application on your computer to open.

Step 4: Select board:

Under the Tools menu select "Board Options and choose Arduino Uno.

Step 5: Select your serial port

Select the serial device of the Arduino board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

Step 6: Coding

You will now begin typing your code. Arduino language is case sensitive and you must complete each command with a semicolon.

// signifies a 'comment' in the code. This means it is not part of the code, but is instructions for the user on what the code means.

Follow the instructions below and use the code to the right to create your code.

In the program to the right, the first thing you do is to initialize pin 13 as an output pin with the line `pinMode(led, OUTPUT);`

In the main loop, you turn the LED **on** with the line:
`digitalWrite(led, HIGH);`

Then you turn it **off** with the line: `digitalWrite(led, LOW);`

The `delay()` commands tell the Arduino to do nothing for X milliseconds

```
/*
  Blink
  Turns on an LED on for one second, then off for one second,
  repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage
  level)
  delay(1000); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Student Programming Reflection

Math Connection:

1. Were you able to adjust the code to change the intervals to 5 seconds on and off and 5 seconds on and 2 seconds and off? How did you adjust the code to meet the interval change?
2. Did you have to increase or decrease the 1000 in the code to get 5 seconds and 2 seconds?
3. How many milliseconds are in one second? Show the math conversion. 5 seconds? 2 seconds?

Reflection:

1. What challenges did you have, if any, to programming the Arduino to blink on and off at 5 second intervals? How did you resolve any challenges you encountered?
2. What do you think about the Arduino Uno? Was it a good way for you to explore basic computer programming? How does this compare to the computer programming you have previously been exposed to?
3. Think about your cell phones and customized ringtones. How complicated or different do you think code would be to provide instructions to a cell phone to play a particular ringtone?

Golden Nugget

Positive

Change



Golden Nugget

Positive

Change



Golden Nugget

Positive

Change



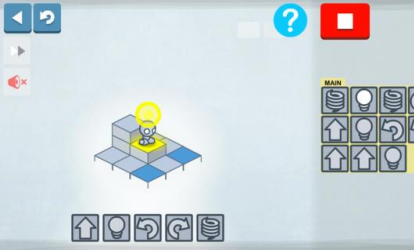

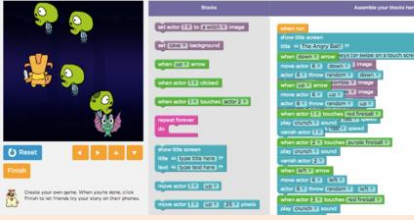
Golden Nugget

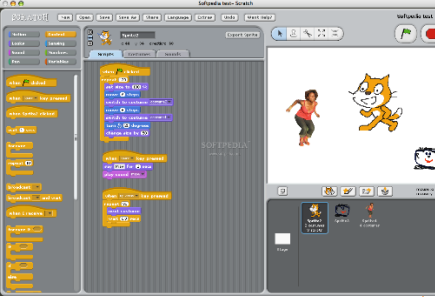
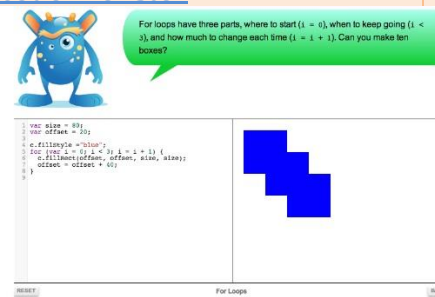
Positive


Change

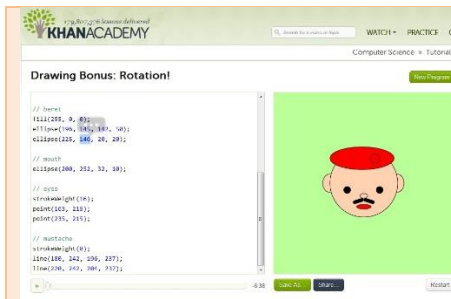


Coding Websites

Website	Programming Language	Description	Instruction	Ease of use
<p>Lightbot: www.lightbot.com</p> 	<p>Drag and drop</p>	<p>Introduction to programming. Very simple and focuses on a student's ability to give simple commands to move a 'lightbot'. Very basic.</p>	<p>This is a paid website, but if you scroll down and click "demo puzzles: web" students can go through 10+ puzzles. This is primarily an app and can be downloaded for around \$5 on most tablet and phone devices if you want to have more levels or need it on a device vs computer.</p>	<p>Very easy, it will not take students long to complete all free levels</p>
<p>Tynker: https://www.tynker.com/hour-of-code/puzzle</p> 	<p>Mainly drag and drop</p>	<p>Introduction to programming using drag and drop code.</p>	<p>Tynker has both paid code lessons and free lessons. The Hour of Code Puzzles is a great introduction to coding and is free. If you enjoy this website and format there are hours of coding programs along with coinciding lessons to buy (\$399 for 16 hours)</p>	<p>Very easy and because it only has a limited number of free games, students will easily complete them.</p>
<p>Code Studio: https://studio.code.org/</p> 	<p>Mainly drag and drop</p>	<p>Introduction to programming mainly using drop and drag code. Tons of lessons and different levels for students to practice coding. Daily lessons available for teachers.</p>	<p>There are multiple parts to this website. A great place to start is https://code.org/curriculum/course2 for daily class lessons for teachers to teach coding. Objects, activities, and assessments are all outlined on this website. There are also multiple 'unplugged' lessons that do not require a computer. For middle school start with Course 2 or 3. The students can go to https://studio.code.org/s/course2 to find step by step different coding activities where they build upon their previous knowledge. Students will get a certificate for the completion of each course. If you don't want students to</p>	<p>Very easy, but has higher levels. If you have students who know how to program Java, Robot C++, Python or similar languages this website will be too rudimentary for them. For first time</p>

			<p>follow a unit plan, but simply want them to explore coding you have them go to</p>	<p>programmers this is a great website.</p>
<p>Scratch: https://scratch.mit.edu</p> 	<p>Drag and drop</p>	<p>Students can create simple games, animations, and stories using drag and drop code. The great thing about this website is the project possibilities are endless. 6th graders particularly tend to like this program.</p>	<p>Go to the website and click 'create'. On the right side click "Getting Started with Scratch" which will prompt the student to what they need to do</p>	<p>This website gives pretty good instructions on how to use it. Some students lose interest after a while, but others can work on this website for incredible amounts of time. Sometimes it helps for the students to look at the project page to get ideas.</p>
<p>Code Monster: http://www.crunchzilla.com/code-monster</p> 	<p>JavaScript</p>	<p>This website more of a tutorial rather than a game. It is a good introduction to JavaScript in that students are writing basic code and manipulating the second screen.</p>	<p>Simply go to the website and start coding. The monster prompts you to do everything.</p>	<p>This website gives very explicit instructions for the students to follow. The only problem is that the students can continue clicking without finishing the activity and it will</p>

<p>Code Combat: www.codecombat.com</p> 	<p>Python, JavaScript, CoffeeScript , Io, Lua, Clojure</p> <p>(Students can choose which language they would like to program in)</p>	<p>This game is pretty addictive and makes students actually write the code instead of drag and drop. It is great that there are different language options because the students can really see a difference in the different commands and syntax. Students will go through medieval mazes collect jewels, armor, and weapons while battling orgs. The high levels have students input coordinates if/else statements.</p>	<p>Simply go to the website and choose your programming languages. This website does not have daily lessons, but does give great explanations and prompts for the students. For the most part middle school students have great success with this, but it will be helpful for the teacher to first play the game to be able to troubleshoot the coding problems students will have. Boys especially love this game.</p>	<p>count it as complete.</p> <p>Definitely more difficult than the drag and drop code, but still not too advanced. The great thing about this game is it is similar to most drag and drop games in that students can instantly see what their commands are doing.</p>
<p>Khan Academy: https://www.khanacademy.org/computing/computer-programming</p>	<p>JavaScript, HTML, SQL</p>	<p>Khan Academy is great in that it is truly formatted</p>	<p>Simply go to the website and choose which language you would like to get started in. Students can follow the step by step instructions and lessons.</p>	<p>Probably the most advanced out of these websites.</p>



as a class. Students can complete each section and advance as they go. This is not drag and drop coding, therefore students should have some exposure to coding. This is less of a game than a lot of the other websites.

***For more resources and app coding visit the following links:**

<https://code.org/learn/beyond>

<http://askatechteacher.com/2014/11/18/hour-of-code/>

<https://www.edsurge.com/guide/teaching-kids-to-code>