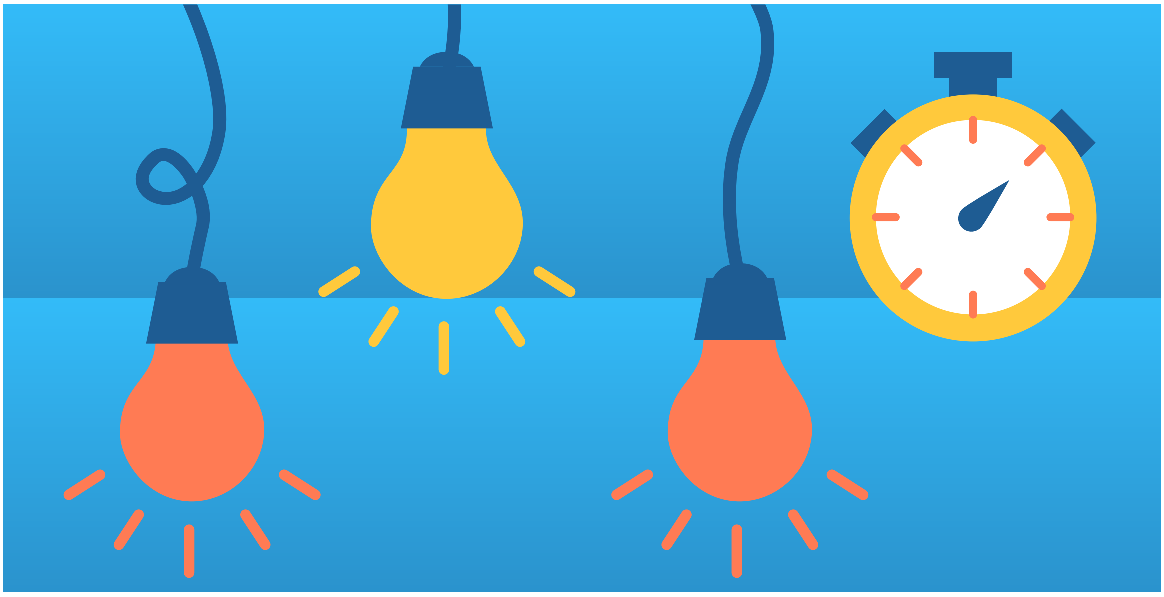

Create a Flashing Beacon

Discover how to control the speed of your code by programming a blinking light sequence.

Courses

- Grades 3-12

Materials

- Cellphone, tablet, or computer
- Internet connection

Description

In this activity, students will learn the fundamental programming concept of "timing." They will use delay blocks to control the sequence of a virtual lamp, creating a classic blinking beacon effect and understanding why pauses are crucial in coding.

Educational Objectives

- Understand the concepts of "LED" and "timing".
- Create a technological object (prototype) using a device.
- Identify relationships between technology and the surrounding world.
- Evaluate personal and others' work in individual or team work.
- Discuss and reflect on improvement ideas.

Start (10 minutes) - Introducing the Concept of Time

1. Welcome the students and introduce the activity: **"Today we will learn how to prototype a beacon that flashes on and off."**
2. To spark the discussion, begin with a question: **"Do you know what makes your cellphone screen light up?"**
3. After they share their ideas, explain that the screen is made of tiny lights called LEDs. Then, pose a thought experiment: **"Imagine you write a program that tells a lamp to turn on, and on the very next line, you tell it to turn off. What do you think you'd actually see?"** Let them debate. The answer is... nothing! This is the perfect moment to

explain that computers are incredibly fast and why we need to program pauses, introducing the concept of "Timing."

What is an LED?

Have you ever looked really closely at your TV or smartphone screen? It's made up of millions of tiny, powerful lights called **LEDs**. LEDs are everywhere around us, from the big ones that light up a room to the tiny ones that work together to create the images you see on a screen.

What about the cellphone screen?

Your cellphone screen is packed with thousands of these tiny LEDs. When you watch a video or look at a photo, these are the lights that turn on and off to form the image. When we prototype with Protobject, these are the very LEDs we are controlling with our code!

The Problem with Speed

Here's a puzzle: imagine you tell your prototype to turn a lamp **ON**, and then in the very next instruction, you tell it to turn **OFF**. You'd expect to see a quick flash, right? But you'd actually see nothing! Why? **Machines are incredibly fast!** The lamp does turn on, but it turns off so quickly that our human eyes can't even perceive it.

Instructions Must Be Precise!

Computers can't guess what we want. If our instructions are ambiguous, the program won't work as expected. In our flashing light puzzle, the missing piece was telling the computer **how long to wait** between turning the light on and turning it off. This action of pausing for a specific duration is called **Timing**.

What does 'timing' mean?

To make a light flash like a beacon, we need to create a sequence that includes **wait times**, or delays, between the ON and OFF commands. In programming, we can measure time in different units. In Protobject, we can use **seconds** or **milliseconds** (1000 milliseconds = 1 second) to control our timing perfectly.

Development (20-30 minutes) - Building the Beacon

1. With the theory of timing covered, it's time to put it into practice and build the flashing beacon.
2. Guide your students through **the instructions for connecting the device and coding the light sequence** detailed in the hands-on section below. Encourage them to experiment with different delay times to see how it affects the blinking speed.

Closure (5-10 minutes) - Reflection and Future Applications

1. Excellent! Once everyone has a working beacon, transition into a final discussion to solidify their understanding and challenge them to think bigger.
2. Start the reflection by asking the key question from the activity: **"In what other situations could we use timing?"** Use the following prompts to guide a conversation about how timing is used in technology all around us.

Reflect

You've learned how to control a beacon. Now, think bigger!

- In what other real-world situations could you use timing and delays? Think about games, animations, or even tools you use every day.