Step-by-step lesson plan to introduce students to building and programming with the Hummingbird Robotics Kit.

**TEACHING TIME**

20 min: Teach the Basics  
5 min: Video  
5 min: Introduce Challenge  
15 min: Build and Program  
5 min: Showcase  
5 min: Reflection  
5 min: Clean Up  
**TOTAL: 60 minutes**

**SUGGESTED CRAFT MATERIALS***

- 2-3 Styrofoam Balls  
- 2 Pipe Cleaners  
- 2 Googly Eyes  
- Hot Glue Gun  
- Hot Glue Sticks  
- Foam or Paper for wings  
- Scissors  
- Brass Fastener to attach wings

**HARDWARE***

1 Hummingbird Bit  
1 Position Servo  
1 LED  
1 Hummingbird Battery Pack or Power Adapter  
1 USB Cord  
1 Device to Program (Laptop, Chromebook, Tablet)

*For every 2-3 students*
Computational thinking (CT) is a mindset and a process for problem-solving. This mindset and process are based on methods from computer science. CT is not just math and logic; it is a way of thinking and a set of skills & attitudes. Below are some CT skills covered in this 60-minute introductory lesson. See our Computational Thinking Formative Assessment Guide for more CT resources.

Computational Thinking Objectives for this Lesson:

- **Decomposition**: The ability to take a large problem and divide it into smaller problems that are each more manageable. This way when each smaller problem is solved, the complex problem becomes manageable.
  
  *Students will decompose the movements of a bee into programmable steps.*

- **Algorithmic Design**: The ability to develop a step-by-step strategy for solving complex problems.
  
  *Students will write computer code to cause a motor and LED to mimic the movements of a bee.*

- **Incremental Design and Evaluation**: Solve complex challenges by breaking the problem down and implementing simple, manageable parts. Test and perfect each part one-by-one and eventually combine them into the full solution.
  
  *Student will combine known coding solutions (blinking an LED and moving a motor) to solve a larger problem (how to get these things to happen at the same time).*

**PREPARE ROOM AND MATERIALS**

See [www.birdbraintechnologies.com/hummingbirdbit/makecode/program](http://www.birdbraintechnologies.com/hummingbirdbit/makecode/program)*

**FOR EACH TEAM OF STUDENTS:**

1. Pre-connect a single LED and position servo motor to the Hummingbird.
2. Connect the Hummingbird to your programming device with USB or bluetooth.
3. Open your programming language.

**FOR THE CLASSROOM:**

1. Set craft materials out on table.
2. Set up a video viewing screen and test for adequate sound.
3. Create and program a sample bee, but hide it from sight at first.
   - This can help you understand the challenges your students might experience.
   - Use sample Makecode programs in the Challenge Charts (see page 3).
4. Display the Guiding Question and the challenge steps (see page 3).
   - This could be a large poster, a slide in a slideshow, or written on a board.

*Other programming language options can be found at the end of this lesson plan on page 8
GUIDING QUESTION: How can you recreate the motion of a bee waggle dance with robotic components?

*Other programming language options can be found at the end of this lesson plan on page 8
1. Use the tutorials and coding cards at birdbraintechnologies.com/hummingbirdbit/makecode* to teach your students to do the following with a Hummingbird:
   - Single Color LED
   - Position Servo

**BEE WAGGLE VIDEO (5 MIN)**

1. Show the Video: “What’s the Waggle Dance? And Why Do Honeybees Do It?” (2:49) www.youtube.com/watch?v=LU_KD1enR3Q
2. Ask the Guiding Question: How can you recreate the motion of a bee waggle dance with robotic components?

**INTRODUCE THE BEE WAGGLE CHALLENGE (5 MINUTES)**

1. Demonstrate Your Bee
   - Show your example bee, but keep your program hidden. Programming the bee is the challenge the students will complete.

2. Introduce the Challenge
   - Display the steps of the challenge (see page 3). Keep this posted throughout students’ work time for them to refer back to if needed.

3. Helpful Reminders
   - Prompt your students to spend 1-2 minutes making a verbal or sketched out plan before they begin programming and building.
   - Students sometimes get caught up in the build and run out of time to program. Help your students keep this in mind.

*Other programming language options can be found at the end of this lesson plan on page 8*
**FIRST HOUR OF ROBOTICS**

**The Bee Waggle Dance**

**BUILDING AND PROGRAMMING (15 MINUTES)**

1. **Time**
   - Allow students adequate time to build. They will need at least 15 minutes to build and program their bees.

2. **Students as Creative Problem-Solvers**
   - Empower students to solve their own building and programming problems.
   - Debugging is part of the creative process. Solving building and programming problems through creativity doesn’t distract from the lesson; it is the lesson.

3. **Quick Debugging**
   - **Power Supply:** If motors aren’t moving, check or replace power.
   - **Programming:** Check that students used pause blocks correctly.
   - **Device Connection:** Check that the Hummingbird is connected to the programming device. Check the USB or Bluetooth connection.
   - Visit [support.birdbraintechnologies.com](http://support.birdbraintechnologies.com) for more tips.

**SHOWCASE (5 MIN)**

1. **Tell your students to get their bees ready for the showcase.**

2. **Tips to make the showcase even more fun:**
   - Play the song “Flight of the Bumblebee” while students all run their bees.
   - Turn out the lights so you can really see the lights blinking.
   - Take a video on your phone.
   - Post videos on social media! (Tag @BirdBrainTech to show us your robot explorations).

---

**ANSWER STUDENT QUESTIONS WITH QUESTIONS**

What problem are you trying to solve right now?

What solutions have you tried so far?

**TIP**

Remember, there’s always another way to solve a problem. Empower your students to share their code with each other. No two programs will be the same, and no two bees will move the same. Seeing examples of how others achieved the goal will help students understand the relationship between the code on the screen and the movement in the real world.
Allow students time to reflect on the learning they did:
• What was your favorite part of the lesson today?
• What surprised you today?
• What was easy about learning today?
• What was challenging about learning today?
• What do you want to learn next?
• What do you want to build next?
• What problem do you want to solve with programming next?

Classroom practices for faster clean up and reset of technology:
• Disconnect bees from motors and LEDs.
• Leave LEDs and motors connected to the Hummingbird for the next class.
• Clear bee waggle challenge programs from your screen, reset to blank for the next class.
• Turn off battery packs to conserve energy.
**WHAT'S NEXT?**

**VIEW PROJECTS**
See how other educators introduced robotics across the curriculum with free standards-aligned projects

**PLAN**
Use our Curriculum Integration Planning Tool to bring robotics into everyday instruction

**BUILD**
Check out Hummingbird building tips and hacks on our Build Page

**LEARN MORE**
Grow your coding skills with Hummingbird using our online video tutorials

*Other programming language options can be found at the end of this lesson plan on page 8*
THE FIRST HOUR OF ROBOTICS

The Bee Waggle Dance

ADDITIONAL PROGRAMMING RESOURCES

**BIRDBLOX**

Use the BirdBlox programming language with any iOS, Kindle Fire, or Android devices. To see programming tutorials, planning tools, and more, visit [birdbraintecnologies.com/hummingbirdbit/birdblox](http://birdbraintecnologies.com/hummingbirdbit/birdblox).

**CHALLENGE 1**

- **Bit Position Servo**: Tapped 1° 0°
- **Bit LED**: 1° 100°
- **wait**: 0.1 secs

**CHALLENGE 2**

- **Bit Position Servo**: Tapped 1° 20°
- **Bit LED**: 1° 0°
- **wait**: 0.1 secs

**SNAP!**

Use the Snap! programming language with any Windows, Apple, or Chromebook machines. To see programming tutorials, planning tools, and more, visit [birdbraintecnologies.com/hummingbirdbit/snap](http://birdbraintecnologies.com/hummingbirdbit/snap).

**CHALLENGE 1**

- **Hummingbird Position Servo**: Tapped 1° 0°
- **Hummingbird LED**: 1° 100°
- **wait**: 0.1 secs

**CHALLENGE 2**

- **Hummingbird Position Servo**: Tapped 1° 20°
- **Hummingbird LED**: 1° 0°
- **wait**: 0.1 secs