Battery Sensor
Grades 6-12
Designed by Robolink

Summary:
CoDrone EDU batteries have a five-minute life, which means that battery percentage could decrease pretty dramatically while a student is flying. In this lesson, students will create and run a battery checkers using the print statements and the CoDrone EDU's LED.

Guiding Question(s):
- How can the CoDrone EDU's features be used as a tool?

Learning Objectives:
Students will be able to:
- Create and run programs in Python that checks the CoDrone EDU's battery and prints the percentage
- Create and run a program in Python that checks the CoDrone EDU's battery and uses the LED as a battery percentage display

Materials needed:
- CoDrone EDUs, remotes, and USB cables (1 drone for 2 students, or 1:1 if there are enough drones)
- Charged CoDrone EDU batteries
- Technology for showing online videos with sound and shared screens
- Laptops or desktop computers with internet access
- Engineering journal or worksheets

Lesson Title: Battery Sensor **Time:** 1 hour 30 minutes

Engagement: (Introduction)
- Ask students to think of all the electronics that they use and how they know that their battery is low or dead on each of them. If they are stuck, give an example (if they have a phone or laptop on them, they can check for a battery icon). Next, ask them how they can tell if their CoDrone EDU is running out of battery, and if needed, pass out CoDrone EDUs so they can look. Tell them that they will be programming a part of their drone to serve as a battery display.

Exploration: (Activity)

- Have students complete Battery Checker on Robolink Basecamp, including the Low Battery challenge at the end of the lesson. If students are having problems, they need to talk to classmates before asking the teacher!

Explanation: (Recap)

- Ask students to use pseudocode to explain to a partner how they programmed their drone to complete the Low Battery challenge using both their own words and the appropriate academic language.

Elaboration: (Extension)

- Have students add a conditional that will land their drone when it's within 2% of the no energy limit. For example, if a student knows that their drone truly cannot fly at 10% battery, they should add a conditional statement that will land the drone at 10-12% battery life.

Evaluation:

- In a journal or on a worksheet have students answer the following questions. An example of an engineering portfolio is available on Google Sites.

1. What other features on your CoDrone could work as a battery display? How would you program them to show battery percentage?
2. Explain either your LED battery checker or safe landing program using pseudocode. Why did you make your program the way you did?
3. Did you have any problems running your code? If so, what did you do to fix them?
4. What did you learn?

Related Vocabulary: battery, code, conditional, else statement, if statement, if else statement, kill switch, landing, LED, program, pseudocode, Python, request, sensor, value, variable

Standards:

CCSS:

ELA-LITERACY.RST.6-8.3: Follow precisely a multistep procedure when carrying out experiments, taking measurements, or performing technical tasks.
ELA-LITERACY.RST.9-10.3: Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks,

attending to special cases or exceptions defined in the text.
ELA-LITERACY.RST.11-12.3: Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.
MATH.PRACTICE.MP5: Use appropriate tools strategically.
MATH.PRACTICE.MP7: Look for and make use of structure.

CSTA:

2-CS-02: Design projects that combine hardware and software components to collect and exchange data.
2-CS-03: Systematically identify and fix problems with computing devices and their components.
2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.
2-AP-19: Document programs in order to make them easier to follow, test, and debug.
3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.

ISTE:

5D: Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.
6A: Students choose the appropriate platforms and tools for meeting the desired objectives of their creation or communication.