MISSION 3: Bright Byte Lights **Lesson 2 (Objective 6)**	**Time Frame: 40-60 minutes**
Project Goal: Students will convert between decimal and binary numbers and use them to control LEDs. **Learning Targets** • I can define “bit” and “byte”. • I can convert a decimal number to binary. • I can convert a binary number to decimal. • I can use a binary number to turn on/off user LEDs. • I can use a binary number to turn on/off line sensor LEDs.	**Key Concepts** • Computers use binary numbers to represent all information. • You can control (turn on and off) LEDs using binary numbers. This is much faster, and less lines of code, than turning on and off each LED individually. • Binary numbers have only two digits: 0 and 1, and it uses the base 2 number system.
Assessment Opportunities • Mission 3 Lesson 2 Log • Submit completed program ***BinaryLEDs*** • Submit the program with extensions • Mission 3 Obj. 6 Review Kahoot!	**Success Criteria** ☐ Convert a number between 0 and 15 to binary ☐ Convert a binary number to decimal ☐ Convert a number between 0 and 255 to binary ☐ Write code using binary to turn on/off user LEDs ☐ Write code using binary to turn on/of ls LEDs
Teacher Materials in Learning Portal • Mission 3 Lesson 2 Slides • Mission 3 Lesson 2 Log • Mission 3 Lesson 2 Answer Key	**Additional Resources** • Mission 3 Obj. 6 Review Kahoot! • Secret Coders video (introduces binary numbers and Activity #1) • ***BinaryLEDs*** sample code (learning portal)

Vocabulary

- **Binary:** A number system, or computer language, that uses only 0s and 1s
- **Bit:** A single binary digit (on/off or 1/0)
- **Byte:** A set of 8 bits of binary data
- **Bit banging:** Controlling hardware with binary digits

New Python Code

`leds.user(16)`	Use a decimal number to turn on/off all user LEDs
`leds.user(0b10101010)`	Use a binary number to turn on/off all user LEDs (0b for binary, then 0=off, 1=on for each LED)
`leds.ls(0b11111)`	Use a binary number to turn on all line sensor LEDs
`leds.ls(0b00100)`	Use a binary number to turn on only the middle line sensor LED

Real World Applications

Computers combine binary digits (bits) into binary codes that enable them to represent any type of data, from numbers and text to images and sounds.

- Text: a unique numerical code is assigned to each character, which is stored and processed as binary sequences.
- Images: An image is represented using a grid of pixels, with each pixel assigned a number for its color.
- Sound: Computers convert continuous analog sound waves into a series of digital values using binary code.

Teacher Notes:	**Extensions / Cross-Curricular:**
• This lesson uses Mission 3 Objective 6 at the beginning and end, but adds three activities to help students understand binary numbers. The best way to present this lesson is to start with Obj. 6 for the intro, and then use the slides all the way until the end. They can use CodeSpace instructions and CodeTrek if needed for the program. • If you don't have much time, or you don't want the extra activities on binary, you can cut them out of the lesson and just do Objective 6. If so, use the Mission 3 Obj. 1-6 Kahoot review instead of the one in the additional resources.	• Make a poster or chart of Python commands. • Use the line sensor LEDs and user LEDs in a light show. • Do an unplugged activity with binary, like binary bracelets. • **MATH:** Binary uses base 2. Discuss number systems with other bases, like the octal system with base 8 or hexadecimal with base 16. Other bases are fun, too, like base 3. • Supports **language arts** through reading instructions and reflection writing.

Preparing for the lesson:

- Look through the slides and workbook. Decide what materials you want to use for presenting the lesson. The slides can be converted to Google Slides. They can be projected on a large screen. The workbook (if used) can be printed or remain digital through your LMS and given to students.
- Be familiar with the mission log assignment and the questions they will answer. This lesson includes three extra activities for converting binary and decimal numbers. You will want to practice the activities before assigning them to students. Prepare the assignment to give through your LMS.
- Watch the video and be familiar with activity #1 (if using it in your lesson). If you want students to complete the first activity, they will need small objects (can be bits of paper) and a printed form to use.
- If you have a word wall, or another form of vocabulary presentation, prepare the new terms.

Lesson Tips and Tricks:

💡 **Teaching tip:**

You can use a variety of discussion strategies to get the most engagement from your students. For example, you can have students write their answers before asking anyone for an answer. You can use one of many think-pair-share methods. You can have students write their answer and share with someone, and then have other students share answers they heard from their peers. You can randomly select students to answer.

👫 **Pre-Mission Warm-up:** -- slides 2-4

Students can write in their log first and then share, or discuss first and then write in their log.

- Question: How many ways can you represent the value 5?
- This question is to get students to think about 7 and how the number we use is just a symbol. There can be many ways to represent 7: fingers, with objects, tally marks, domino or die, words, math problems, etc. This is a good question to have students share-out their ways and see how many representations of 7 they can come up with. You can show slide 3 for examples.
- Question: How does a computer represent 7?
- Students are not expected to know the real answer to this question. Appreciate their creativity as they discuss how a computer may represent 7. Some students may know that a computer uses binary numbers.

💻 **Mission 3 Lesson 2 Activities:**

Each student will complete a Mission Log. Students could work in pairs through the lesson, or they can work individually. They will need a CodeBot and cable for the lesson.

💡 **Teaching tip: Objective 6 Introduction** -- slides 5-6
The two slides contain the information from the first two paragraphs of Objective 6. After that, the students don't need the instructions in CodeSpace for a while.

💡 **Teaching tip: Binary Numbers** -- slide 7
This slide introduces binary numbers and has a link to a video. The video is on YouTube. The entire video is about 9 minutes, but the beginning and end talk about the book series. My recommendation is to start at 1:09 and stop at 7:32, for about a 6 ½ minute video clip. The video ends with instructions for activity #1.

💡 **Teaching tip: Activity #1** -- slide 8
This slide refers students to their mission log assignment to complete activity #1. It is based off of the video. The instructions say to use small objects, like bits of paper, to fill in the columns. If students are going to do this, you need to have this document, or a similar paper, printed for students to use.

💡 **Teaching tip: Binary Numbers** -- slides 9-11
These slides transition from 4-bit numbers to 8-bit numbers.

💡 **Teaching tip: Binary Numbers** -- slides 12-13
These slides give examples of converting binary to decimal and decimal to binary. You may need to go over the examples and indicate what a 0 and 1 mean on the chart.

💡 **Teaching tip: Activity #2** -- slide 14
This slide refers students to their mission log assignment to complete activity #2. It is based off the examples from slides 12-13.

💡 **Teaching tip: Binary Numbers** -- slide 15
This slide transitions to using binary in Python code to turn on/off LEDs.

💡 **Teaching tip: Activity #3** -- slides 16-18
These slides have the instructions on completing the activity. Students will use the Console Panel in CodeSpace. They type code directly into the console and see the results immediately on CodeBot.

Note: This activity continues after some more instruction.

💡 **Teaching tip: Binary Numbers** -- slides 19-20
This slide transitions to using binary in Python code to turn on/off LEDs.

💡 **Teaching tip: Activity #3** -- slide 21
Students continue activity #3. They will use a different Python command that uses binary to turn on/off LEDs. They will make up their own combinations of LEDs to try and record them in their mission log.

💡 **Teaching tip: Binary Numbers** -- slide 22
This slide shows the Python code for turning on/off the line sensor LEDs.

💡 **Teaching tip: Objective 6** -- slide 23
This activity is the same one in Mission 3 Objective 6. Students can read the instructions in CodeSpace or from the slide. They must complete the three bullet points to meet the goal. They can do more than those three lines of code, though. They should be able to do the program without CodeTrek, but it is there to help them if needed.

💡 **Teaching tip: Extension** -- slide 24
If you have time, students can do an extension by combining what they have been learning throughout Mission 3 so far.

💡 **Teaching tip: Quiz** -- slide 25
Students take a ❓ short quiz over objectives 1-6. The quiz questions are shown below.

Optional: 🗝️ Mission 3 Obj 6 Kahoot! Review.

A review Kahoot! Is available for this objective. You can do the Kahoot together as a class, or assign it independently.

If you are not doing the three activities on binary and decimal conversion, you can skip a Kahoot! Review, or use the Mission 3 Obj. 1-6 Kahoot! Review.

🧑‍🤝‍🧑 **Post-Mission Reflection:**

There are 2 slides that discuss real-world applications and how computers use binary to represent all data. You can expand on this information as needed.

You can use an extension or cross-curricular activity as post-mission activity.

You can use the Mission 3 Obj. 6 Kahoot as a lesson review. (link above)

End by collecting the Mission 3 Lesson 2 Log.

SUCCESS CRITERIA:

- ❑ Convert a number between 0 and 15 to binary
- ❑ Convert a binary number to decimal
- ❑ Convert a number between 0 and 255 to binary
- ❑ Write code using binary to turn on/off user LEDs
- ❑ Write code using binary to turn on/of ls LEDs

Quiz questions on next page

Why would you add a delay (sleep) after you turn on each LED? +5 XP

- [ ] To give the LEDs time to cool off.
- [ ] So they will turn on.
- [x] So you can see them turn on one at a time.

When you use the debugger, the line of code with the highlight: +5 XP

- [ ] Ran the last time you pressed STEP.
- [ ] Will stop the program.
- [x] Will run the next time you press STEP.
- [ ] Is currently running.

The statement `sleep(1.5)` +5 XP

- [x] Pauses the program for 1.5 seconds.
- [ ] Pauses the program for 1.5 milliseconds
- [ ] Allocates 1.5 kilobytes of sleep space.

What does `from time import sleep` do? +5 XP

- [x] Gives this code access to the "sleep" function from the "time" library.
- [ ] Sleeps from time to time.
- [ ] Allows this code to read the current time.

Which LED does the following turn ON: `leds.ls(0b00100)` +5 XP

- [x] Line Sensor LED 2
- [ ] Line Sensor LED 3
- [ ] Line Sensor LED 1
- [ ] User LED 5