

LESSON: CodeX Link-Up Project		Time: 45 minutes
Overview: This is an easy coding lesson that can be done near the beginning of the school year to introduce coding and the CodeX. This is an engaging project that enables students to “talk” to each other using their CodeX and radio signals. Students use the radio capability of CodeX to send and receive messages. All students will be on the same channel. Therefore they will send a message to everyone and receive messages from everyone. Messages should be school appropriate. Extensions all for further programming opportunities.		Coding Objectives: • I can import a library. • I can use print statements to display text on the screen. • I can code a button press. • I can use an infinite loop to send and receive messages. • I can break out of a loop. • I can blink a pixel. • I can clear the screen.
Grades 6-8 CS Standards: 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-11 Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	Grades 9-10 CS Standards: 3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. 3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. 3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. 3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.	Grades 11-12 CS Standards: 3B-DA-06 Select data collection tools and techniques to generate data sets that support a claim or communicate information. 3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs. 3B-AP-17 Plan and develop programs for broad audiences using a software life cycle process.
Preparation: • Download slides • Be familiar with the final code • Read through the teaching guide	In the folder: • CodeX Link-up project slides • CodeX Link-up final code • Link-up code with extensions	Agenda: • Warm-up (5 minutes) • Complete program using slides (35 minutes) • Optional: Wrap-up (5 minutes)
Teacher Notes: • This lesson is designed as a class project, with individual students creating their own programs. • Students will send a message to other students. You may need to stress the importance of school appropriate messages. If you are concerned about this, you can mandate a safe message, like “Hello from ...” • Almost all mistakes made by students are typing mistakes. If students get errors when they run their code, first look over the code for spelling, punctuation and especially indenting. • The project slides include three extensions. Any or all of them can be done. They do not all have to be done, or in the order given. • The first extension is at a beginner level and can be done by all students. • The second extension requires more typing and more messages. • The third extension is more advanced. It is recommended for students who are familiar with programming and/or Python.		

Slide Extensions:

- Get students in groups and have each group set their own channel. Use an odd-numbered channel.
- Use more buttons to display different messages.
- Select an RGB color for each sent message.

Extensions:

- Make a list of colors, and select a random color for each received message.
- Display a welcome message on the screen when the program starts. Display it for a few seconds and then clear the screen and start sending and receiving messages.
- Use a button to end the program and clear the screen, or display an ending message.
- For advanced programmers, have them try to switch between different channels. For example, if they select "A", then set the channel and send and receive from there. But when they select "B", set a different channel and send and receive from that channel. This would be a great opportunity to use a function.

Cross-curricular Connections:

- **LANGUAGE ARTS:** Have students write down the steps for creating the program, using technical writing.
- **LANGUAGE ARTS:** Have students receive 5 messages, and then use them as inspiration for a short story.
- **MATH:** Give students 10 seconds to send and receive messages. Make a graph of the data, such as the messages received or the colors used.
- **MATH:** Give students 10 seconds to send and receive messages. Analyze the data and calculate the probability of receiving a message from a designated student.
- **COMPUTER SCIENCE:** The messages sent in this project are open, without security. Have a discussion on digital security measures and/or encryption. This also meets additional CSTA and computer science standards.
- **COMPUTER SCIENCE:** Computer data is digital, but radio waves are analog. Have a lesson about converting analog and digital data.
- **SCIENCE:** Learn about radio signals.
- **SCIENCE:** The program extension uses red, green and blue lights to form colors. Have a lesson about light.

Teaching Guide

Warm-up (5-10 minutes)

Slide 2 – Cell phones are not allowed in many classrooms. Discuss with students some ways they can communicate with each other if they can't send texts. You might mention that in the old days, students passed notes!

Create/Run the Program (30 minutes)

 For this project, it is best if each student has his/her own CodeX. They will create their own program and can use it any time throughout the class period to communicate how they are doing.

Students should already have an account in CodeSpace. The programming portion may take longer if your students are doing the extensions.



Teaching tip:

This project is not included in CodeSpace. Download and follow the slides. They include step-by-step instructions as well as code snippets to guide students through the program code creation.

This program is a class project. Show the slides on a large screen or monitor and have the class work on each step together. This is a fairly short program, so keeping the class together and letting them help each other is preferred.



Slides 3-5

These slides give background for the coding project. Students see the buttons on the CodeX and learn about the radio capabilities. They also learn about the string data type. When typing a string in Python, either double quotes (" ") or single quotes (' ') can be used.



Slides 6-7

Students get into CodeSpace to create their program. They can click on the sandbox icon to complete the program. There are no instructions in CodeSpace. Students start a new file and import the libraries.

NOTE: The radio library **MUST** be imported first!

Slide 8: Step 1

Turn on the radio and set the channel. Display a message on the screen to see that the CodeX is ready.

Slide 9: Step 2

Use an infinite loop so the CodeX is constantly listening for messages. Also, students can send their message multiple times without restarting the program.

Use an if statement with BTN_A to send a message. Students should use their name (or initials) instead of ...

Slide 10: Step 3

Write code to listen for a message. If a message is heard, print it on the display screen.

NOTE: All messages will print in the default color of WHITE. You can add a color as a parameter. However, all messages displayed will be in the same color.

Slide 12: Step 4

Students add code to blink a pixel when the button is pressed, as an indicator that they sent a message. In the example, pixel 0 is used. Students can be creative here. They can use a different pixel, or all the pixels. They can use a different color than GREEN.

Slide 14: Step 5

Clear the display by pressing BTN_B. The thing to watch for here is the indenting. The if statement needs to be indented inside the while loop, but not inside the if msg; block.

Slides 15: Finished!

Students can test their code.

- Indenting is very important! Troubleshoot and debug any errors.
- Let students have fun sending and receiving messages!

Slides 16: Extensions

Three extensions are listed on the slide, with instructions following on slides 17-23. Extension #3 requires students to use RGB colors. A website to help with this is given on slide 21. Check the website in advance and make sure it is accessible to students.

Wrap-up / Optional (5 minutes)

 You can wrap-up this project in a variety of ways, depending on your students and your classroom procedures.

- Students can fill out a journal entry about their experience or what they learned during the lesson.
- Students can share with each other or in small groups something they learned, or how they might apply what they learned to a different project.
- You can discuss what coding concepts they learned or used during the project.
- Discuss other projects (or remixes) they think of that can use the coding concepts from this project.
- Incorporate CSTA Standard 2-CS-01 or 3A-AP-21 by having students recommend improvements to the program and/or CodeX as a communication tool, based on how users interact with the device.