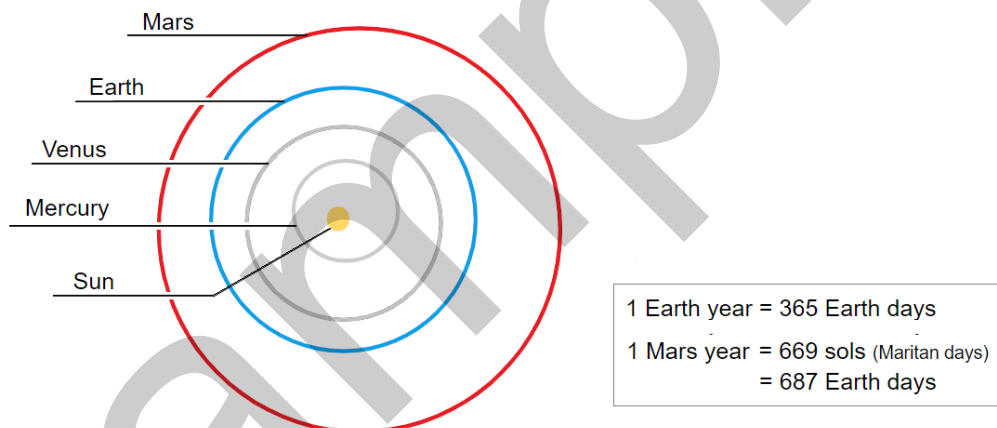# Houston, This is Codey

Introduction

Did you know it takes 8 minutes for light to travel from the sun to earth? Light travels very fast, but the solar system is very, very large. Communicating with robots—and, one day, people—on Mars is tricky because messages can only travel as fast as light waves.

Look at the orbits of Earth and Mars in Figure 1. You can see that sometimes Mars and Earth are closer together and sometimes they are further apart. When Mars and Earth are close, it takes as little as 4 minutes for messages to travel between the planets. But when Mars and Earth are on opposite sides of the sun, it can take 21 minutes! A lot can happen to a robot on an alien planet in 21 minutes. Because communication is so slow between Earth and Mars, Martian robots need to be able to be able to do much on their own, such as navigate around obstacles.

Mars
Earth
Venus
Mercury
Sun

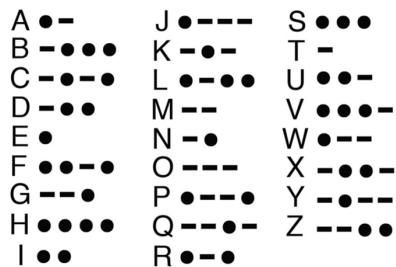| 1 Earth year = 365 Earth days |
| 1 Mars year = 669 sols (Maritan days) |
| = 687 Earth days |

FIGURE 1 Earth is the third planet from the sun; Mars is the fourth

In this activity, you will get a chance to create a simple communication program using mBlock 5 software or the mBlock app.

Beep, Beep

There are many ways to communicate or send a message to someone else: through written or spoken words, pictures, sounds, or even smoke signals. One of the earliest methods of communicating quickly over large distances was the telegraph machine.

A telegraph machine is like a car horn. You push on a button and the machine makes a beep. Engineers would connect neighboring towns with long telegraph wires. So, when someone pushed the button in one town, the machine would beep in the neighboring town. Morse Code is a set of rules for translating beeps into letters. With enough beeps, you could send a whole book!

A ●—	J ●———	S ●●●
B —●●●	K —●—	T —
C —●—●	L ●—●●	U ●●—
D —●●	M ——	V ●●●—
E ●	N —●	W ●——
F ●●—●	O ———	X —●●—
G ——●	P ●——●	Y —●——
H ●●●●	Q ——●—	Z ——●●
I ●●	R ●—●	

**FIGURE 2** In the Morse Code diagram, dots are short beeps and dashes are long beeps.

Let's turn Codey into a telegraph machine by writing the Telegraph Machine program. When running the program, Codey will make short and long beeps when its buttons are pressed.

TelegraphMachine Program

Often when coding, you will often want to write a program that makes something happen when a particular event occurs. For instance, you might want a light to come on when you select a button on the screen. In this case, the "button click" is the event and the light turning on is the thing we want to happen. To turn Codey Rocky into a telegraph machine, you want Codey to beep when a button is pressed. You will write your program using blocks from the Events palette and the Speaker palette.

We'll start with a block from the Events palette: the "when button is pressed" hat block (see Figure 3). Most of the blocks in the Events palette are what we call "hat blocks" because they look kind of like a hat. In mBlock, a section of code that you want to be run on Codey must always begin with a hat block.

In the Events palette, you will also find other events that involve Codey including when the robot is turned on (starts up), when it's shaken, and when one of its buttons is pressed.

FIGURE 3 Events palette

Coding with Codey Rocky: Mission to Mars
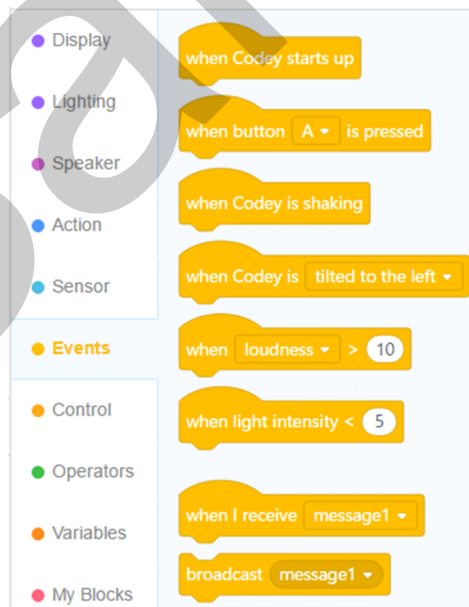
The other block we'll use in the TelegraphMachine program is the "play note" block, which is found in the Speaker palette. The Speaker palette contains sound blocks that make Codey beep or make various sounds.
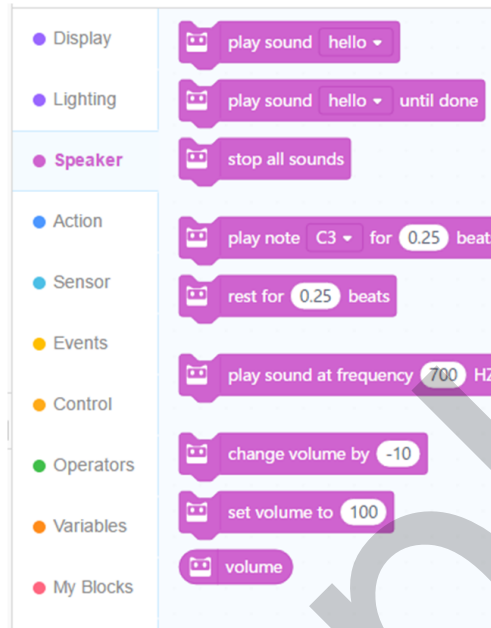

FIGURE 4 Speaker palette

Create the TelegraphMachine program by doing the following:

1. Drag a "when button is pressed" hat block into the programming area in mBlock. By default, this hat block is set to use the A button. In the TelegraphMachine program, we'll use the A button to make a short beep, so we can leave this block as is.

2. Drag a "play note" block from the Speaker palette into the programming area and attach it to the bottom of your "when button is pressed" hat block.

3. Click in the number field where "0.25" is shown and change it to **0.1**. Changing this number will make the beep shorter.

4. Add another "when button is pressed" hat block to the programming area. For the second "when button is pressed" hat block, click or tap the "A" and change it to **B**.

5. Add another "play note" block to the programming area and attach it to the "when button B is pressed" block. Change the number field in the second "play note" block to **0.3** to make a longer beep.

6. When you are done, your program should look like the program in Figure 5.

FIGURE 5 TelegraphMachine program

TRY IT OUT

Create the TelegraphMachine program in mBlock. If you are using the mBlock 5 software, upload the program to Codey by selecting the Upload to Device button. In the mBlock app, no extra upload step is required; the program will run when you press the A or B button. When you press the A button, does Codey make a short beep? Is the beep longer when you press the B button?

Use the Morse Code diagram (Figure 2) to translate your name into Morse Code and tap it out on Codey. Practice sending messages in Morse Code to a partner!

Extensions

1. It is helpful to get lots of feedback when you use a new device, such as your Telegraph Machine. Each time you press a button, you get a beep. It would be nice if Codey also displayed which key was pressed. Use the "show" block in the Display palette to display the button letter on Codey's display when you press a button.

2. Codey has three buttons: A, B, C. Create your own code for each of the letters in the alphabet. For example, if you give each button a different pitched beep, rather than a different length beep as we did in the TelegraphMachine program, you could assign each letter a different beep combination.

Coding with Codey Rocky: Mission to Mars

Houston, This is Codey

Overview

Codey Rocky is a beginner-friendly, programmable robot. In this activity, students have their first opportunity to write code for Codey Rocky with the mBlock 5 software or mBlock app. This activity involves fairly simple programming, as the intention is for students to learn the process of writing code and running it on Codey Rocky. Students will learn about coding "events" and how to write code to make something else happen when particular events occur.

This activity describes the telegraph machine and its operation. As a result, the activity can also serve as a conversation starter about the role technology has played (and continues to play) in changing our society and culture.

If you are new to Codey Rocky and coding, we recommend reading *Getting Started with Codey Rocky*. This getting started guide will help you understand the parts of the Codey Rocky robot and give you an overview of how to write for, and run code on, Codey Rocky.

Objectives

- Become familiar with the mBlock 5 software or mBlock app.
- Write a Codey Rocky program in mBlock software or mBlock app.
- Run a program on Codey Rocky.
- Practice troubleshooting when problems arise.

Computer Science Teachers Association (CSTA) Standards

|	
1B-AP-15	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.
1B-IC-18	Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.

Materials

- Codey Rocky robot (charged)
- mBlock 5 software on Windows/macOS computer or mBlock app on iOS/Android device

Tips

This activity only uses Codey, the detachable controller part of the Codey Rocky robot. If you unpack the robots ahead of class, for this activity you need only distribute the Codey part of the robot.

Solutions to Extensions

1. The addition of a few blocks from the Display palette can make it a little easier to tap out Morse Code messages on Codey.
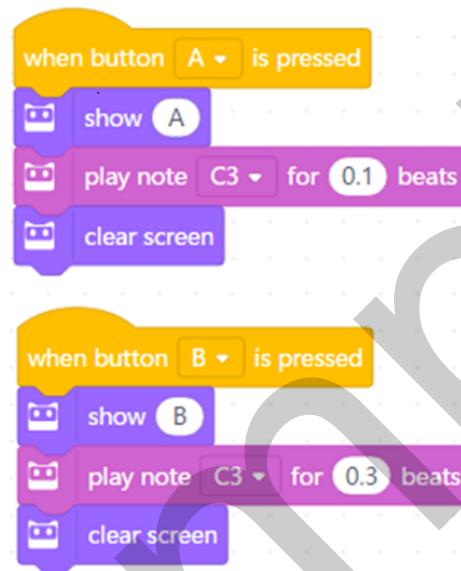
FIGURE 1 TelegraphMachinePlus program includes "show" and "clear screen" blocks to tell the user which button was pressed.

2. In this extension, students are challenged to create their own communication method, similar to Morse Code. Students will create a variety of communication methods. Encourage students to consider the following questions as they design and iterate their code:
   - How easy is it to learn my communication method?
   - Where might someone get confused when using the method?
   - How fast can I communicate using this method? Are there changes I could make to speed it up?
   - How can I simplify my communication method?