makeblock education

Lesson 1

Meet the CyberPi

Subject:	Computer Science	**Grade(s):**	6-12
Duration:	45 Minutes	**Difficulty:**	Beginner

Objectives

By the end of this lesson, students will be able to:

- Identify key features of the CyberPi.
- Connect a CyberPi to a computer using mBlock software.
- Examine example programs for the CyberPi using the mBlock software.

Overview

Meet the CyberPi, a feature-rich micro-controller with a plethora of sensors, buttons and a full-color screen display. Discover a variety of key features of CyberPi through an exploration of sample programs in mBlock.

Key Focus

- Components and features of the CyberPi
- Navigate mBlock software
- Establish a connection between the software and hardware

Pre-lesson Checklist

For the teacher:

- Computer with mBlock 5 installed or mBlock Web version
- CyberPi with USB-C cable
- Pocket Shield (optional)
- Example program(s) included in mBlock software

For the student:

- Computer with mBlock 5 installed or [mBlock Web version]
- CyberPi with USB-C cable
- Pocket Shield (optional)
- Example program(s) included in mBlock software

Content Standards

Type	Indicator	Standard
ISTE	6a	Students choose the appropriate platforms and tools for meeting the desired objectives of their creation of communication.
K12 CS Framework	Practice 2-1	Cultivate working relationships with individuals possessing diverse perspectives, sills, and personalities.

Agenda (45 minutes)

Duration	Content
5 minutes	**Warm-up** • Meet the CyberPi
15 minutes	**Hands-on** • Tour of mBlock 5 • Connect the CyberPi o Add the CyberPi Extension o Test Live Mode • Explore an Example Program o Test Upload Mode
20 minutes	**Try It** • Explore Example Programs
5 minutes	**Wrap-up** • Reflect on the CyberPi Features and Capabilities • Brainstorm Ideas for CyberPi Programs • Lesson Extension(s)

Activities

Warm-Up [5 minutes]

Meet the CyberPi

1. Guide students through an unboxing of the CyberPi. Show the students the following components and have students locate them in their CyberPi Kit:
    a. CyberPi
    b. USB-C Cable
    c. Pocket Shield (not included in Base Kit)
    d. mBuild Sensors (not included in Base Kit)
2. Have students read the CyberPi box and the Quick Start Guide. Then, have students write a short summary listing what they have learned so far about the features and capabilities of the CyberPi.

Hands-On [15 minutes]

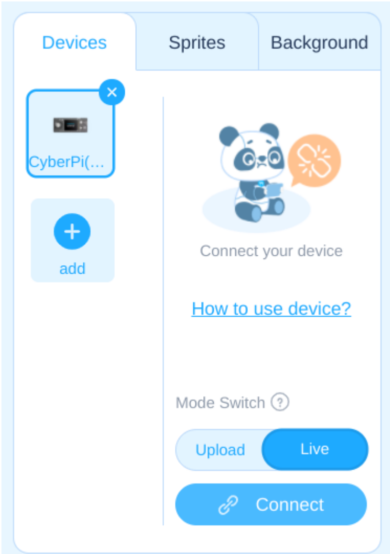
Tour of mBlock 5

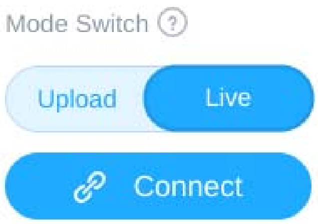
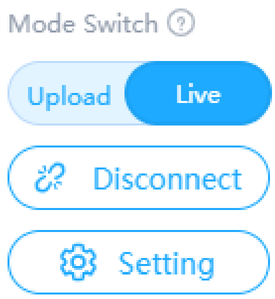
1. Open the mBlock 5 software or [mBlock 5 Web version].
2. Introduce students to the following key areas of the software interface:

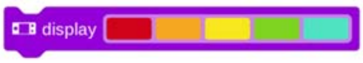
Name of Area	Function
Menu Bar	• Select language • Create, open or save the file • Find an example program, help file, etc.
Stage Area	• View the project stage • Select and edit sprites and backgrounds • Connect hardware devices
Block Area	• Find and select script blocks organized into color-coded categories • Find and add extensions
Script Area	• Combine blocks to create programs or scripts • Drag blocks and arrange them in a certain order to control sprites, backgrounds and/or devices

Connect the CyberPi

3. Plug the CyberPi into the computer using the included cable. The CyberPi should boot up and the screen will display either the last program uploaded or the Home menu.
4. On the **Devices** tab in mBlock, click the **Add** button. Select **CyberPi** and click **Ok**.
5. Click the **Connect** button. Then, select the USB port and click **Connect**.
6. If connected successfully, the button will change to Disconnect.



Not Connected	**Connected**
	

7. Notice, the CyberPi is connected in **Live** mode. Let's test the connection.
8. In the **Block Area**, choose the **LED** category.
9. Click the  block. Observe the CyberPi as the LED strip displays the colors indicated.

Explore Example Programs

10. Click **Tutorials** in the upper right corner. Select **Example Programs**.
11. Choose the **CyberPi** label to see example programs for the CyberPi.
12. Find and select the **Rainbow Lights** program.
13. Have students read the code and predict what will happen.
14. Connect the CyberPi in **Live** mode.
15. Click the  block. Observe the CyberPi as the LED strip displays the colors indicated. (Note, a glowing yellow border surrounds the script in the Script area indicating the script is running.) Click the blocks to stop the program.
16. Explain to students the difference between **Live** mode and **Upload** mode.

Mode	Description
Live Mode	• Program is run by the computer (is not stored on the CyberPi) • CyberPi must remain connected to the computer • mBlock project must remain open • Must be used for stage programming
Upload Mode	• Program is uploaded and stored on the CyberPi • No communication with the computer

	• CyberPi can be disconnected from the computer • mBlock software may be closed • Program will remain on CyberPi until a new program is uploaded in its place

17. Switch the CyberPi to **Upload** mode and click the **Upload** button.
18. The **Upload Progress** window will appear and will disappear when uploading is complete. The CyberPi will reboot and observe the **Rainbow Lights** program. Every time the CyberPi starts up, the **Rainbow Lights** program will run.

Try It
[20 minutes]

Explore Example Programs

1. Instruct students to explore 3 of the remainder of the example programs included with mBlock. Some suggestions include:
   - Buzzer
   - Twinkle Twinkle Little Star
   - Voice Reactive Lights
   - Trigger Reminder
   - Simple Timer
   - Step Counter
   - Motion-sensing Chart
2. While they review the projects, have them document the following for each program they explore:
   - Write a description of the program.
   - Identify which components of the CyberPi are used for each task of the program.
   - Make inferences about what portions of the project code does.

Wrap-Up
[5 minutes]

Reflection & Brainstorming

1. Facilitate a discussion for students to share about their favorite example projects. Encourage students to identify the CyberPi features and capabilities they are most excited to learn about.
2. Have students document what they hope to learn while completing these lessons and any questions they may have.
3. Now that students have seen some examples of the CyberPi in action, encourage them to brainstorm ideas for a problem in their daily life that they want to solve using the CyberPi and mBlock.

Lesson Extension(s)

- If students need additional support using mBlock, consider having the students complete the mBlock 5 Getting Started Activities.

Lesson 2

Sound Machine

Subject:	Computer Science	**Grade(s):**	6-12
Duration:	45 Minutes	**Difficulty:**	Beginner

Objectives

By the end of this lesson, students will be able to:

- Identify CyberPi's input(s) and output(s).
- Write pseudocode to plan and design a program in mBlock.
- Create a program in mBlock using the CyberPi buttons to trigger events.
- Select and use programming blocks to control the speaker and LED strip.

Overview

In this lesson, students create a disco party using the on-board LEDs and speaker. This program will use the CyberPi buttons to trigger events and run scripts. Students will also program a button to stop all sounds and lights, as well as a button to restart the CyberPi.

Key Focus

- Input and Output components on the CyberPi
- Writing Psuedocode
- Creating a program in mBlock

Pre-lesson Checklist

For the teacher:

- Computer with mBlock 5 installed or [mBlock Web version]
- CyberPi with USB-C cable
- Pocket Shield (optional)
- Example program(s): CyberPi – Lesson 2 – Sound Machine

For the student:

- Computer with mBlock 5 installed or [mBlock Web version]
- CyberPi with USB-C cable
- Pocket Shield (optional)

Content Standards

Type	Indicator	Standard
CSTA	2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms.
K12 CS Framework	Practice 5-2	Create a computational artifact for practical intent, personal expression, or to address a societal issue.

Agenda (45 minutes)

Duration	Content
5 minutes	**Warm-up** • Input vs. Output
15 minutes	**Hands-on** • Plan a Program with Pseudocode • Write the Program • Randomizing the Output • Restart a CyberPi
20 minutes	**Try It** • Create a Sound Machine
5 minutes	**Wrap-up** • Project Showcase • Lesson Extension(s)

Activities

Warm-Up [5 minutes]

Input vs. Output

3. Discuss the following definitions with the students.

Term	Definition*
input	A device or component that allows information to be given to a computer.
output	Any device or component that receives information from a computer.

*Definitions from Code.org – CSD Unit 1

4. Using a smartphone as an example, have students work in pairs or small groups to create a list of the input and output for a mobile phone. Some examples may include:

Smartphone	
Input	**Output**
Microphone Touch Screen Buttons GPS Motion Sensor (tilting the phone) Light Sensor Camera Internet connection Temperature Sensor Charging Port Bluetooth	Speaker Screen / Display Headphones Vibration Internet connection LED (flashlight / camera flash) Charging Port Bluetooth

5. Have students reflect on Lesson 1 – Meet the CyberPi and create a list of the input and output for the CyberPi. Encourage students to refer back to the product documentation included with the CyberPi if they get stuck.

CyberPi

Input	Output
Microphone Buttons (A, B & Home) Joystick Charging Port Bluetooth Motion Sensor (gyro) Light Sensor Volume/Sound Sensor	Speaker Screen / Display LED Strip Indicator LED (shows charging & power on)
with Pocket Shield and mBuild Kit	
Multi-Touch Sensor Slider Ultrasonic Sensor Third-Party Sensors	Motors (encoder & servo) LED Strip Third-Party Modules

Hands-On
[15 minutes]

Plan a Program with Pseudocode

19. Discuss with the students the importance of planning a program before developing it in the software.

    Introduce student to *pseudocode* which can be a helpful tool for planning an mBlock project.

Term	Definition
pseudocode	Written sequence of steps for a program written in English or the programmer's native language.

20. Using the following project description, guide students through writing the pseudocode for the project.

Sound Machine	
Project Description	Create a project where the CyberPi continuously makes sound with the A button and lights up the LEDs with the B button. Have the middle joystick button stop the sounds and lights.

Pseudocode	When Button A is pressed:
	    Forever set all the LEDs a specified color
	
	When Button B is pressed:
	    Forever Play the buzzer at a specified note

	When middle joystick button is pressed: Stop all sounds and lights

21. Have students examine the pseudocode above and identify the input and output for the program.

Sound Machine Project	
Input	**Output**
Button A Button B Joystick Middle Button	Speaker / Buzzer LED(s)

Write the Program

1. Now that the program is planned, it is time to learn about the new blocks needed to program the pseudocode.

Category	Block	Function
Events	when button A ▾ pressed when joystick pulled↑ ▾	Specifies the action on the CyberPi that triggers the execution of the actions attached.
Control	forever	Loop Statement Continuously execute the actions nested inside the block.
	stop all ▾	Stops all scripts, including all loops.
Audio	play buzzer at 700 Hz for 1 secs	Play a note on the CyberPi buzzer for a specified amount of time. Frequency Range: 0 to 1000
LED	LED all ▾ displays R 255 G 0 B 0	Light all or an individual on-board LED a specified color. Color Value Range: 0 to 255

		Turns off all or an individual on-board LED.

2. Open the mBlock 5 software or [mBlock 5 Web version](). Add the CyberPi in the Devices tab and connect in Live mode.
3. Remind students how to drag-and-drop blocks from the color-coded categories in the Block Area. Instruct them to build each of the following scripts:

Note, the *middle pressed* option can be found on the drop-down menu.

4. Test the program in Live mode and/or Upload mode.
5. Have students experiment with different values for the buzzer and LED blocks to observe how the CyberPi performs.
6. **Extension:** If time permits, teach students about RGB color values. Or, allow them to use a color picker to identify the RGB values of specific colors.

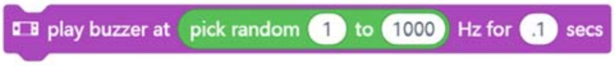
Randomizing the Output

7. In the code above, a specific buzzer frequency and RGB color value were programmed in the code. The CyberPi is repeating the same sound and same LED color forever. The following block can be used to allow the program to select a random value each time the forever loop repeats.

Category	Block	Function

Operators		Computer or CyberPi selects a random value between the specified range.

8. Modify the previous programs to include the following random blocks:



Note, these ranges correspond to the range of values accepted for each block.

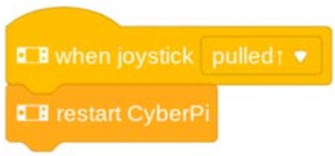
9. Have students test the new program. Note, students can change the seconds value in the play buzzer block to a decimal if they would like a faster buzzer sound.

Restart a CyberPi

10. Programming a button to restart the CyberPi can be a helpful tool in upcoming lessons. So, guide students through programming a button to manually restart the CyberPi.

Category	Block	Function
Control		Restarts or reboots the CyberPi device. The CyberPi will play the last program uploaded to the device.

11. Instruct students to build the following script and test the program:



Try It
[20 minutes]

Create a Sound Machine

3. Provide students time to experiment with their existing code. Encourage them to try different values for the parameters in the blocks.

4. Challenge students to explore other blocks in the LED and Audio categories in the Block Area. Some blocks of interest may be:

5. Once they've explored the various blocks. Have students write pseudocode for an enhanced version of the **Sound Machine** project.
6. Have students create their project using their pseudocode as a guide.

Wrap-Up
[5 minutes]

Project Showcase

4. Group students into pairs and have students present their CyberPi Sound Machine to their partner.
5. Have students ask each other the following questions:
    a. What feature of your project are you most proud of?
    b. What was most challenging about this project?

Lesson Extension(s)

- Program a stop button using the Repeat Until block with the 

block.
- Use the joystick buttons to program multiple different sounds and light shows.

Lesson 3

Sound Recorder

Subject:	Computer Science	**Grade(s):**	6-12
Duration:	45 Minutes	**Difficulty:**	Beginner

Objectives

By the end of this lesson, students will be able to:

- Create a program in mBlock that records and playbacks audio.
- Follow an iterative process to develop a solution to a computing problem.

Overview

By combining the speaker, microphone and integrated storage, students will transform the CyberPi into a pocket-sized audio recorder and playback device. Through an iterative process, students will evaluate their projects and improve their sound recorders.

Key Focus

- Record audio with the CyberPi
- Play recordings
- Use an iterative design process

Pre-lesson Checklist

For the teacher:

- Computer with mBlock 5 installed or mBlock Web version
- CyberPi with USB-C cable
- Pocket Shield (optional)
- Example program(s):

  CyberPi – Lesson 3 – Sound Recorder 1 CyberPi – Lesson 3 – Sound Recorder 2

For the student:

- Computer with mBlock 5 installed or [mBlock Web version]
- CyberPi with USB-C cable
- Pocket Shield (optional)

Content Standards

Type	Indicator	Standard
CSTA	2-CS-1	Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.
K12 CS Framework	Practice 3-1	Identify complex, interdisciplinary, real-world problems that can be solved computationally.
K12 CS Framework	Practice 5-1	Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.
K12 CS Framework	Practice 6-1	Systematically test computational artifacts by considering all scenarios and using test cases.
K12 CS Framework	Practice 6-2	Identify and fix errors using a systematic process.
K12 CS Framework	Practice 6-3	Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.

Agenda (45 minutes)

Duration	Content
10 minutes	**Warm-up** • Evolving Computing Solutions
10 minutes	**Hands-on** • Make a Plan • Create Sound Recorder 1.0
20 minutes	**Try It** • Plan and Create Sound Recorder 2.0
5 minutes	**Wrap-up** • Summarize • Lesson Extension(s)

Activities

Warm-Up
[10 minutes]

Evolving Computing Solutions

6. Discuss with students how computing solves everyday, real-world problems. As a class, compile a list of computing solutions that students and their families come across on a daily basis. Some examples may include:
   a. Smartphone
   b. Alarm Clock
   c. Refrigerator Alarm
   d. Automobiles
   e. Public Transit
   f. GPS
   g. Microwave
   h. Weather Report or App

7. Discuss with students how computing solutions have evolved over time.

   Here is a specific example you may want to share:

Evolution of Directions to a Destination

- Using a printed road map or transit map to determine a route from one place to another prior to departure.
- Using a website on a computer to generate printable directions prior to departure. (i.e. Mapquest or a transit website)
- Using a mobile device to view a website with directions while in route.
- Using a mobile device to view a map while in route (no directions provided). (Note, this was the first iteration of the Google Maps app; it had no directions.)
- Using a mobile device to view a map with driving directions or transit directions.
- Using a mobile device to provide turn-by-turn GPS driving directions (no transit or walking support, yet).
- Using a mobile device to provide turn-by-turn GPS driving directions with route adjustments based on live traffic data.
- Using a mobile device to provide GPS-assisted driving, walking, or transit directions in real-time.

The use of GPS directions on a smartphone or mobile device has evolved over time. For example, the first iteration of Google Maps was missing many key features (i.e., walking directions, public transit, live traffic, road closures, destination information, etc.) which have been added over time through evaluation of the solution, user feedback and data collection.

Hands-On
[10 minutes]

Make a Plan

22. Using the following problem statement, guide students through planning a solution to the problem. A sample solution has been provided for you. Note, keep the initial solution very simple. This will allow for students to iterate and add features as they work on developing a feature-rich sound recorder.

Sound Recorder 1.0	
Problem	One of your classmates is taking a foreign language class. They would like to practice their pronunciation of some of the vocabulary learned in class, but by the time they get home, they forget how their teacher said the words. How can the CyberPi be used to help your classmate with this problem?
Proposed Solution	*Create a sound recording using the CyberPi to record the teacher pronouncing the words and then playback the recording when studying at home.*
Pseudocode	*When Button A is pressed:* *Start recording* *When the middle joystick button is pressed:* *Stop recording* *When Button B is pressed:* *Playback recording*

Create Sound Recorder 1.0

23. Now that the program is planned, it is time to learn about the new blocks needed to program the pseudocode.

Category	Block	Function

Audio	start recording	Starts recording audio. Limited to a 10 second recording.
	stop recording	Stops recording audio.
	play recording	Plays the last recording stored on the CyberPi.

24. Open the mBlock 5 software or [mBlock 5 Web version](). Add the CyberPi in the Devices tab and connect in Live mode.
25. Remind students how to drag-and-drop blocks from the color-coded categories in the Block Area. Instruct students to build the following scripts and test the program:

```
when button A pressed
start recording
```

```
when joystick middle pressed
stop recording
```

```
when button B pressed
play recording
```

26. Ask students if the device is ready to give to the classmate to use in their foreign language class. Many students will begin to identify areas of improvement for this design. Have students brainstorm and create a list of ways to improve this basic sound recording device. If students need additional guidance, consider asking the following questions:
    a. How will your classmate know how to use the CyberPi Sound Recorder?
    b. How will your classmate know if the CyberPi is recording?
    c. What other features could be useful for your classmate?
    d. What other components of the CyberPi can be used to create a more feature-rich solution? (i.e., display, LED strip, speaker, motion sensor)

Try It
[20 minutes]

Plan and Create Sound Recorder 2.0

1. Explain to students that programmers and software developers improve upon their solutions often. (Note, this is why software apps have updates and redesigns.) Software development is an iterative process.
2. Using the list of improvements they created in the previous section, have students identify the three most important features they would like to add to the **Sound Recorder** program.
3. Have students describe and justify the features they will be adding. Then, have students write the pseudocode for each feature. A sample plan is provided below:

<table>
<tr><th colspan="2">Sound Recorder 2.0</th></tr>
<tr><td rowspan="2">Feature #1</td><td>Add instructions to the display telling the user what buttons to press to control the CyberPi Sound Recorder.</td></tr>
<tr><td>When the CyberPi starts up:
Display on the screen:
"Press A to start, Press joystick to stop, Press B to play"</td></tr>
<tr><td rowspan="2">Feature #2</td><td>Use the LEDs to tell the user when the CyberPi is recording.</td></tr>
<tr><td>When Button A is pressed:
Set all LEDs to display green
Start recording

When the middle joystick button is pressed:
Set all LEDs to display red
Stop recording

When Button B is pressed:
Set all LEDs to display blue
Playback recording</td></tr>
<tr><td rowspan="2">Feature #3</td><td>Add the ability to change the CyberPi's volume.</td></tr>
<tr><td>When the CyberPi starts up:
Set the volume to 50%

When the joystick is pulled up:
Increase the volume by 10%

When the joystick is pulled up:
Decrease the volume by 10%</td></tr>
</table>

Note, there are two example programs included with this lesson. These should be for the teacher to review. Students should be encouraged to brainstorm and create a program from their own ideas.

4. Have students create the **Sound Recorder 2.0** using their pseudocode as a guide.

Wrap-Up

[5 minutes]
Summarize 1. Have students share a few of the ideas they incorporated into their sound records. 2. Remind students that the next activity will have them continue developing the Sound Recorder project through an iterative process. **Lesson Extension(s)** • Have students explore UX Design (User Experience Design) and the role it plays in software development. • Have students research accessibility and usability with software development.

Lesson 4

Sound Recorder Iteration

Subject:	Computer Science	**Grade(s):**	6-12
Duration:	45 Minutes	**Difficulty:**	Beginner

Objectives

By the end of this lesson, students will be able to:

- Provide constructive feedback on a computing solution.
- Modify an existing program to improve user experience.
- Follow an iterative process to develop a solution to a computing problem.

Overview

Continuing with the Sound Recorder project, students will acquire peer feedback and reflect on their initial solution. Then, students will plan and create a feature-rich, sound recorder project.

Key Focus

- Collect and evaluate peer feedback
- Use an iterative design process

Pre-lesson Checklist

For the teacher:

- Computer with mBlock 5 installed or [mBlock Web version]
- CyberPi with USB-C cable
- Pocket Shield (optional)
- Example program(s): CyberPi – Lesson 2 – Sound Machine

For the student:

- Computer with mBlock 5 installed or mBlock Web version
- CyberPi with USB-C cable
- Pocket Shield (optional)

Content Standards

Type	Indicator	Standard
CSTA	2-IC-22	Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.
CSTA	2-CS-1	Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.
K12 CS Framework	Practice 2-3	Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders.
K12 CS Framework	Practice 3-1	Identify complex, interdisciplinary, real-world problems that can be solved computationally.
K12 CS Framework	Practice 5-1	Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.
K12 CS Framework	Practice 6-1	Systematically test computational artifacts by considering all scenarios and using test cases.
K12 CS Framework	Practice 6-2	Identify and fix errors using a systematic process.
K12 CS Framework	Practice 6-3	Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.

Agenda (45 minutes)

Duration	Content
15 minutes	**Warm-up** • Collect Peer Feedback
20 minutes	**Try It** • Plan and Create Sound Recorder 3.0
10 minutes	**Wrap-up** • Project Documentation • Lesson Extension(s)

Activities

Warm-Up
[15 minutes]

Collect Peer Feedback

6. With the entire class, conduct a formal peer review and feedback activity. Explain how software developers rely on user feedback, reviews and data to plan for additional iterations of a computing solution.
7. Have students place their plan for Sound Recorder 2.0 on their desk, open the program in mBlock and place the CyberPi with the uploaded recorder on their desk.
8. As time permits, have students rotate around the room and provide feedback on their classmates projects. Some guiding questions for feedback may include:
   a. What feature do you like best about their project?
   b. Were instructions clear on how to use their recorder? Did you have to guess or make any assumptions on how to use it?
   c. Is there anything they could add to their project to make it more user-friendly?
   d. Is there a feature you think would enhance their recorder project?

Try It
[20 minutes]

Plan and Create Sound Recorder 3.0

1. Have students review the feedback they received and to brainstorm a list of ideas for **Sound Recorder 3.0**. Some ideas for improvements:
   - Include a title that appears when the CyberPi starts.
   - Change the colors of the on-screen text.
   - Set a specific duration for the recording instead of using a stop button.
   - Use the joystick to control the volume and/or playback speed.
   - Display the current volume level on the screen or use the LED strip to indicate the volume.
   - Have the LED strip animate while recording.
   - Use the joystick to control the duration of the recording by storing the recording duration in a variable.
2. Following the steps from the previous lesson, have students identify improvements, write pseudocode and create their Sound Recorder 3.0 project.

Wrap-Up
[10 minutes]

Project Documentation

1. Now that students have created a feature-rich sound recorder, have them write a description for their recorder that explains the features they have included in their design.

Lesson Extension(s)

- Have students collect feedback from a variety of stakeholders (i.e., parents, teachers, friends, etc.).
- Have students add improvements to a different student’s project.

Lesson 5

Game Controller

Subject:	Computer Science	**Grade(s):**	6-12
Duration:	45 Minutes	**Difficulty:**	Beginner

Objectives

By the end of this lesson, students will be able to:

- Differentiate between stage programming and device programming.
- Examine and describe how an existing project functions.
- Create a program in mBlock using the CyberPi to control sprites.
- Modify an existing program.

Overview

In this lesson, students will turn the CyberPi into a game controller by combining device programming and stage programming in mBlock. Students will examine example programs to discover how the CyberPi can control the movement of a sprite. Then, through pair programming, students will modify an existing game to program a CyberPi game controller.

Key Focus

- Combine stage programming and device programming
- Pair Programming
- Decomposition and abstraction

Pre-lesson Checklist

For the teacher:

- Computer with mBlock 5 installed or mBlock Web version
- CyberPi with USB-C cable
- Pocket Shield (optional)
- Example program(s):

  CyberPi – Lesson 5 – Chase Game   CyberPi – Lesson 5 – Space Adventures

For the student:

- Computer with mBlock 5 installed or [mBlock Web version]
- CyberPi with USB-C cable
- Pocket Shield (optional)
- Example program(s):

  CyberPi – Lesson 5 – Chase Game

  CyberPi – Lesson 5 – Space Adventures

Content Standards

Type	Indicator	Standard
CSTA	2-AP-16	Incorporate existing code, media, and libraries into original programs, and give attribution.
ISTE	2c	Students demonstrate an understanding of and respect for the rights and obligations of using and sharing intellectual property.
K12 CS Framework	Practice 4-1	Identify complex, interdisciplinary, real-world problems that can be solved computationally.
K12 CS Framework	Practice 7-3	Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.

Agenda (45 minutes)

Duration	Content
5 minutes	**Warm-up** • Video Game Discussion
15 minutes	**Hands-on** • Explore Example Games • Abstraction and Decomposition
20 minutes	**Try It** • Pair Programming • Modify an Example Project
5 minutes	**Wrap-up** • Respecting Intellectual Property • Lesson Extension(s)

Activities

Warm-Up
[5 minutes]

Video Game Discussion

1. Have students discuss the following with a partner:
    - Do you play video games? If so, how often?
    - What device or video game system do you prefer to play games on?
    - What's your favorite video game controller?
    - What features make it your favorite?

Hands-On
[15 minutes]

Explore Example Games

1. Assign students partners and determine who is Partner A and who is Partner B.
2. Instruct each student to open the appropriate example project:
    - **Partner A** Lesson 5 – Chase Game
    - **Partner B** Lesson 5 – Space Adventure
3. Have students connect the CyberPi in Live mode and play the game.
4. Each project combines stage programming and device programming to create an mBlock game that is controlled by the CyberPi. Explain the following to students:

Term	Definition
stage programming	Sequences of programming blocks that interact with the sprites and background of the stage in mBlock.
device programming	Sequences of programming blocks that interact with the physical computing device(s) connected in mBlock.

5. Have students examine their assigned example program and differentiate between the stage programming and the device programming.
   Note, device programming will be on the device(s) listed on the Devices tab and stage programming will be on the sprites and backdrops listed on the Sprites and Background tabs.

Abstraction and Decomposition

6. Students will be using computational thinking strategies, abstraction and decomposition, to examine the example projects and determine how to modify an existing game to add a CyberPi game controller. They will use abstraction to ignore or filter out parts of the program that are unnecessary to the challenge and use decomposition to break down the stage programming and device programming parts that are needed to create the controller.

Term	Definition
abstraction	Simplify a problem by hiding, filtering out or ignoring unnecessary details.
decomposition	Break a problem down into smaller pieces.

7. Instruct students to carefully examine their assigned game and to complete the following challenge:

Game Controller	
Problem	You are tasked with taking an existing program and adding a CyberPi game controller. Use the example program to learn how to complete this task.
Challenge	Write a comprehensive explanation of how the CyberPi and the stage interact in the example game.

Computational Thinking	
Abstraction	Encourage students to use abstraction to filter out unnecessary parts of the programs that are not interacting with the CyberPi. Some examples include: **Space Game** o The Title, ball, asteroids and health sprites o The background o The “when I receive gameStart” scripts **Chase Game** o The Title and Bat sprites

	o The background o The “when I receive gameStart” scripts
Decomposition	Encourage students to decompose the parts of the program. Be sure students include specifics such as: o Which buttons on the CyberPi are being used? o Which sprites are being controlled? o How do the buttons on the CyberPi control the sprites?

8. Through the exercise above, students should have discovered the following blocks:

Category	Block	Function
Events		Send a message from one device, sprite or background to another. Used to synchronize actions.
Control		Conditional Statement Executes the actions nested inside if a condition is met.
Sensing		Used with a conditional statement to detect whether the joystick is pressed or moved by the user.

Try It
[20 minutes]

Pair Programming

5. Students will be working with a partner to add a CyberPi game controller to an existing project. Introduce students to the Pair Programming roles:

Pair Programming	
Navigator	Keeps track of the big picture and helps to decide what to do next.
Driver	The person using the computer actually writing the code.

*Definitions from Code.org – CSD Unit 1

6. Have students switch roles every 3-5 minutes during the portion of the lesson.

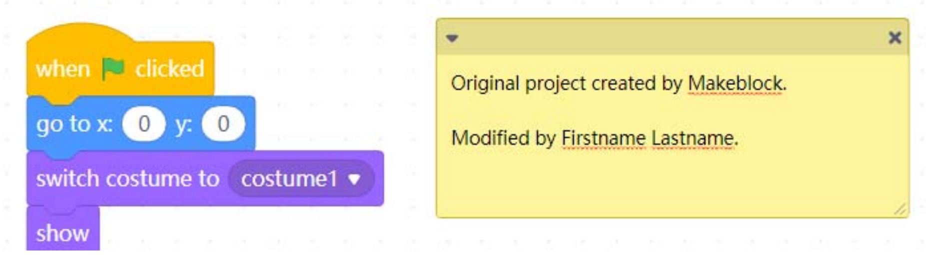
Modify an Existing Project

7. Click **Tutorials** in the upper right corner. Select **Example Programs**.
8. Choose the **Stage** label to see example programs programs for the Stage.
9. While pair programming, have students choose an example program and modify the program scripts to add a game controller.

Wrap-Up
[5 minutes]

Respecting Intellectual Property

3. Discuss the importance of respecting intellectual property and providing credit to creators.
4. Have students add a comment to their project that provides credit. To add a comment, right-click on the script area and select Add comment.



Lesson Extension(s)

- Have students research intellectual property, copyright, creative commons and citations.
- Have students create a new game that incorporates stage programming and device programming.
- Have students add a title and game instructions to the project.

Lesson 6

Sensor Meter

Subject:	Computer Science	**Grade(s):**	6-12
Duration:	45 Minutes	**Difficulty:**	Beginner

Objectives

By the end of this lesson, students will be able to:

- Describe how CyberPi sensors detect the surrounding environment.
- Debug errors in programs in mBlock.
- Document a program using comments in mBlock.

Overview

Discover how the on-board sensors on the CyberPi represent loudness and light intensity of the surrounding environment. Students will learn about data representation and graphing of sensors values.

Key Focus

- Data representation
- Debugging programs
- Understanding sensors

Pre-lesson Checklist

For the teacher:

- Computer with mBlock 5 installed or [mBlock Web version]
- CyberPi with USB-C cable
- Pocket Shield (optional)
- Example program(s):

  CyberPi – Lesson 6 – Sensor Meter CyberPi – Lesson 6 – Sensor Meter V2

For the student:

- Computer with mBlock 5 installed or [mBlock Web version]

- CyberPi with USB-C cable
- Pocket Shield (optional)
- Example program(s):

  CyberPi – Lesson 6 – Sensor Meter

Content Standards

Type	Indicator	Standard
CSTA	2-AP-19	Document programs in order to make them easier to follow, test, and debug.
CSTA	2-CS-03	Systematically identify and fix problems with computing devices and their components.
ISTE	5b	Students collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making.

Agenda (45 minutes)

Duration	Content
5 minutes	**Warm-up** • Data and Society
15 minutes	**Hands-on** • Exploring Sensor Data • Charting the Sound Sensor
20 minutes	**Try It** • Charting the Light Sensor
5 minutes	**Wrap-up** • Documentation • Lesson Extension(s)

Activities

Warm-Up [5 minutes]
Data and Society

27. Find a current event about smart home devices to present to the class. Discuss how these solutions use sensors to provide security, convenience and automation for consumers.

Hands-On
[15 minutes]

Exploring Sensor Data

1. Open the mBlock 5 software or [mBlock 5 Web version](). Add the CyberPi in the Devices tab and connect in Live mode.
2. Open the Lesson 6 – Sensor Meter example project. Click the Green Flag to run the program. Observe the values for the CyberPiVolume and CyberPiLightIntensity.
3. Have students run the program and observe the sensor values and graphing in different scenarios, such as:
    - CyberPi sitting on the desk
    - Covering up the CyberPi (light intensity should decrease)
    - Shining a flashlight on the CyberPi (light intensity should decrease)
    - Trying to be as quiet as possible (volume should decrease)
    - Clapping or talking near the CyberPi (volume should increase)
4. Challenge students to determine the minimum and maximum values the sensors report to the computer. Note, the Light sensor and Sound sensor have a range of 0 to 100.
5. Introduce students to the blocks that are used for the sensors on the CyberPi:

Category	Block	Function
	ambient light intensity	Stores a numerical value representing the light intensity detected by the light sensor on the CyberPi.

Sensing	loudness	Stores a numerical value representing the volume detected by the sound sensor on the CyberPi.

Charting the Sound Sensor

6. Introduce students to the blocks that are for creating a line chart on the CyberPi:

Category	Block	Function
Display	line chart, add data (50)	Plots a point on the line chart on the CyberPi display.
	line chart, set spacing to (5) pixel	Changes the horizontal spacing of the line chart.
	set brush color ( )	Sets the color of the line chart.

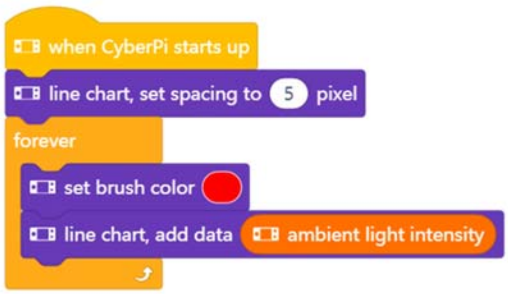
7. Open the mBlock 5 software or [mBlock 5 Web version](). Add the CyberPi in the Devices tab and connect in Live mode.
8. Instruct students to build the following script and test the program:

Try It
[20 minutes]

Charting the Light Sensor

1. Challenge students to modify the code to include a red line chart for the light intensity. The CyberPi should show both charts at the same time.

2. Some solutions may, at first, appear to be correct. But, encourage students to think critically and fully test their solution. For example, the following code does not properly graph the data. When testing this code, you should observe that speaking loudly by the CyberPi makes both the red and blue lines increase.



3. Encourage students to persevere through fixing or debugging their code until they get the correct solution. The following is a correct solution:

Wrap-Up
[5 minutes]

Documentation

1. When programming, it can be helpful to document programs to make them easier for someone to follow, test and debug. Often, multiple programmers work on one project. Instruct students to use a comment(s) to explain how the code works.

Lesson Extension(s)

- Have students program the LEDs or speaker to react to the loudness or light intensity.
- Have students add a title and game instructions to the project.
- Have students program a sprite or background to change costumes based on the loudness or light intensity. (Note, a variable is required to store each sensor value.)

Lesson 7

Color Mixer

Subject:	Computer Science	**Grade(s):**	6-12
Duration:	45 Minutes	**Difficulty:**	Beginner

Objectives

By the end of this lesson, students will be able to:

- Use a variable to store a value.
- Change a variable based on user input.
- Write a program in mBlock that executes a program if a condition is met.

Overview

Students will be introduced to variables to create a CyberPi Color Mixer. This program will use the joystick and buttons to control the R, G, B color values of all of the on-board LEDs. Then, students will use conditional statements to ensure that the R, G, B values do not go out-of-range.

Key Focus

- Storing data with variables
- Using conditional statements

Pre-lesson Checklist

For the teacher:

- Computer with mBlock 5 installed or [mBlock Web version]
- CyberPi with USB-C cable
- Pocket Shield (optional)
- Example program(s): CyberPi – Lesson 7 – Color Mixer

For the student:

- Computer with mBlock 5 installed or [mBlock Web version]

- CyberPi with USB-C cable
- Pocket Shield (optional)
- Example program(s): CyberPi – Lesson 7 – Color Mixer

Content Standards

Type	Indicator	Standard
CSTA	2-CS-02	Design projects that combine hardware and software components to collect and exchange data.
CSTA	2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.
CSTA	2-DA-07	Represent data using multiple encoding schemes.

Agenda (45 minutes)

Duration	Content
5 minutes	**Warm-up** • Conditional Statements Game
15 minutes	**Hands-on** • Storing Data with a Variable • Using a Conditional Statement
20 minutes	**Try It** • Completing the Program
5 minutes	**Wrap-up** • Variables • Lesson Extension(s)

Activities

Warm-Up [5 minutes]

Conditional Statements Game

28. Play a game similar to Simon Says with your class to demonstrate conditional statements. With your students, read each statement below and have the students obey the command. (Feel free to write your own commands for your students.)
    - **IF** your name has the letter "S" in it, **THEN** raise your hand.
    - **IF** you have a pet cat, **THEN** clap your hands.
    - **IF** you play a sport, **THEN** stomp your feet.
    - **IF** you are wearing socks, **THEN** touch your feet.
    - **IF** your favorite ice cream is chocolate, **THEN** say "Yum."
29. Share with students that these commands are examples of conditional statements. In programming, conditional statements are used to perform specific actions if a condition is true.

Hands-On [15 minutes]

Storing Data with a Variable

9. Discuss the following definition with the students:

Term	Definition*
variable	A placeholder for a piece of information that can change.

*Definitions from Code.org – CSD Unit 1

10. Review the following challenge and pseudocode with the students:

Color Mixer	
Project Description	Create a project where the joystick and buttons on the CyberPi control the R, G, B values of all of the LEDs.
Pseudocode	When the joystick is pulled up: Increase the red (R) value of all LEDs by 5

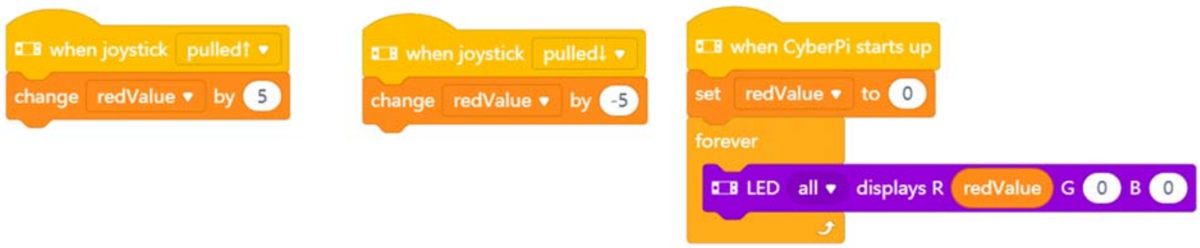
	When the joystick is pulled down: Decrease the red (R) value of all LEDs by 5 When the joystick is pulled right: Increase the green (G) value of all LEDs by 5 When the joystick is pulled left: Decrease the green (G) value of all LEDs by 5 When Button A is pressed: Increase the blue (B) value of all LEDs by 5 When Button B is pressed: Decrease the blue (B) value of all LEDs by 5

11. To program this project, a variable must be used to store the value of each of the LEDs. The variables will start at zero (0) when the CyberPi starts and will be adjusted using the joystick or buttons.
12. Open the mBlock 5 software or [mBlock 5 Web version](). Add the CyberPi in the Devices tab and connect in Live mode.
13. Go to the **Variables** section of the **Block Area**. Click the **Make a Variable** button.
14. Name the new variable **redValue** and leave **For all sprites** selected.
15. Introduce students to the following new blocks which are now available in the **Variables** section:

Category	Block	Function
Variables	set redValue ▾ to 0	Sets a variable to a specific value.
	change redValue ▾ by 1	Changes a variable by a specific value (positive or negative).
	redValue	Returns the current value stored in the variable.
Operators	( ) < 0	Compares two values and determines if the first value is **less than** the second value. Returns TRUE or FALSE.
	( ) > 255	Compares two values and determines if the first value is

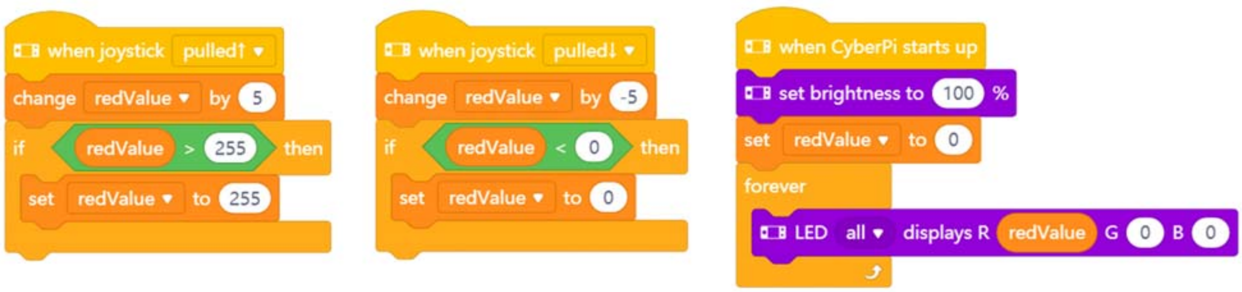
		greater than the second value. Returns TRUE or FALSE.

16. Instruct students to build the following scripts and test the program to observe the joystick up/down controls increasing the red value of all of the LEDs:



Using a Conditional Statement

17. Remind students that the R, G, B values have a range of 0 to 255. With the program above, the **redValue** variable can be changed to a value that falls outside of that range. We can use conditional statements to control the minimum and maximum values. Instruct students to modify the code and add the following conditional statements to the program:



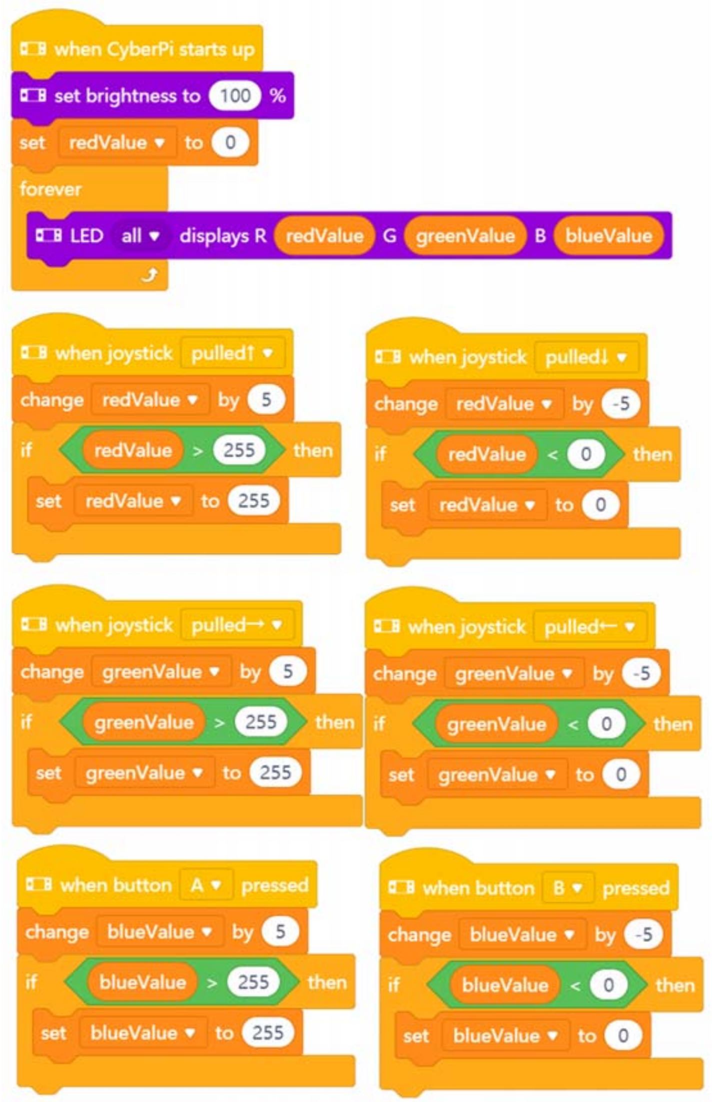
Note, students can also add a block to set the brightness to 100% when the program starts.

18. Explain to the students how the conditional statements do not allow the redValue to ever be greater than 255 or less than 0.

Try It
[20 minutes]

Completing the Program

1. Instruct students to finish the program based on the pseudocode provided earlier in the lesson. The following is an example of a completed program:



2. Students may find it helpful to add the following script inside the forever loop to see the values of the variables on the CyberPi display:

Wrap-Up
[5 minutes]

Variable Review

2. Discuss with students how variables are used to store information in a variety of computing devices and applications. Discuss the following examples with students:

- Fitness trackers stores the number of steps.
- Automobiles (with a digital display) stores the number of miles driven.
- Mobile devices stores the battery level and reports it as a percentage.
- Store shopping cards store the number of visits until you earn a reward.
- Video games store health, lives and scores.

Lesson Extension(s)

- Have students add a title and instructions to the project.
- Have students create a variable for the LEDNumber and use the middle joystick button to control which LED is being changed.

Lesson 8

Strength Meter

Subject:	Computer Science	**Grade(s):**	6-12
Duration:	45 Minutes	**Difficulty:**	Beginner

Objectives

By the end of this lesson, students will be able to:

- Write a program in mBlock that keeps score.
- Write a program that executes actions for a specified amount of time.
- Display text on the Cyberpi display.

Overview

In this lesson, students will create a fun game with the CyberPi where the player shakes the CyberPi for ten seconds. The students will program the game to keep score of how many times the shaking strength is greater than 50.

Key Focus

- Keeping score
- Using the CyberPi Timer
- Displaying text on the CyberPi display

Pre-lesson Checklist

For the teacher:

- Computer with mBlock 5 installed or mBlock Web version
- CyberPi with USB-C cable
- Pocket Shield (optional)
- Example program(s): CyberPi – Lesson 8 – Strength Meter

For the student:

- Computer with mBlock 5 installed or [mBlock Web version]
- CyberPi with USB-C cable
- Pocket Shield (optional)
- Example program(s): CyberPi – Lesson 8 – Strength Meter

Content Standards

Type	Indicator	Standard
CSTA	2-CS-02	Design projects that combine hardware and software components to collect and exchange data.
CSTA	2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.

Agenda (45 minutes)

Duration	Content
5 minutes	**Warm-up** • Variable Review
15 minutes	**Hands-on** • Detecting Strength • Keeping Score
20 minutes	**Try It** • Creating Strength Meter 2.0
5 minutes	**Wrap-up** • Motion Sensing • Lesson Extension(s)

Activities

Warm-Up [5 minutes]

Variable Review

30. Review variables with the class and explain how variables can be used to keep track of a score in an mBlock project. Review with students how to create a variable, set a variable and change a variable in mBlock.

Hands-On [15 minutes]

Detecting Strength

1. Review the following challenge and pseudocode with the students:

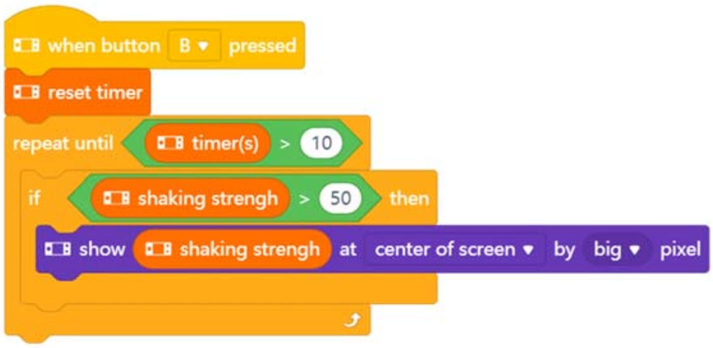
Strength Meter 1.0	
Project Description	Create a game where the user shakes the CyberPi for 10 seconds and earns a point for every time the shaking strength exceeds 50.
Pseudocode	When Button B is pressed: Set the score to zero (0) Reset the timer Repeat for 10 seconds If shaking strength is greater than 50 then, Change the score by one (1) When Button A is pressed: Show the score on the CyberPi display

2. Open the mBlock 5 software or mBlock 5 Web version. Add the CyberPi in the Devices tab and connect in Live mode.

3. Introduce students to the following new blocks:

Category	Block	Function
Control	repeat until	Loop Statement Execute the actions nested inside the block until a condition is TRUE.
Sensing	reset timer	Sets the timer on the CyberPi to zero (0).
Motion Sensing	shaking strengh	Returns a value representing how strong the CyberPi is shaken.
Display	show makeblock at center of screen ▾ by 16 ▾ pixel	Displays text on the CyberPi display at a specified position and size.

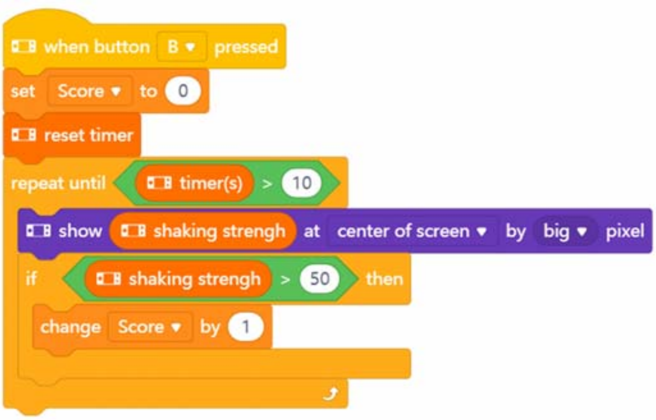
4. Instruct students to build the following scripts and test the program to observe the shaking strength values:



Keeping Score

5. Make a variable named **score** and leave **For all sprites** selected.

6. Instruct students to add the following scripts to the program:



7. Now that the CyberPi is keeping track of the score, add the following scripts to display the score on the screen with button A is pressed:

8. Have students play the game and see how many points they can earn.

Try It
[20 minutes]

Creating Strength Meter 2.0

3. With a simple version of a Strength Meter game created, there are many improvements that can be added to the program. Have students identify areas of improvement for this design. Some ideas for improvement include:
   - Use the LED Strip to inform the user when the game is running.
   - Add a sound effect each time a point is earned.
   - Add a sound effect when time runs out.
   - Add a title and instructions.
   - Use the joystick and a variable to change the difficulty of the game (i.e., shaking strength for easy is 30, medium is 50 and hard is 70).
2. Have students develop a plan and pseudocode for the improvements they would like to add to their project.

3. Have students create the **Strength Meter 2.0** using their pseudocode as a guide.

Wrap-Up
[5 minutes]

Motion Sensing

3. The CyberPi has a 3-axis gyroscope and a 3-axis accelerometer which detects motion, acceleration and vibration. The shaking strength block uses this component to determine how strong the CyberPi is being shaken. Have students brainstorm a list of devices they use that use a gyroscope. Some examples may include:
   - Mobile phones change the screen orientation based on device rotation.
   - Screens on mobile phones light up when the device is picked up.
   - Video game controllers detect motion.
   - Robotic vacuums detect if they have fallen or tipped over.

Lesson Extension(s)

- Have students program the LEDs to progressively light up based on the strength (see example).
- Have students program a 2-player game where it tracks a score for both players, compares the scores and declares a winner.

makeblock education

Lesson 9

Gift Alarm

Subject:	Computer Science	**Grade(s):**	6-12
Duration:	45 Minutes	**Difficulty:**	Beginner

Objectives

By the end of this lesson, students will be able to:

- Use wireless network technology to communicate between computing devices.
- Solve a problem using a computational solution.
- Identify uses for wireless communication.

Overview

Students will use the CyberPi to create a program which detects whether or not a friend has shaken their birthday present. Through the use of wireless communication, students will send messages between computing devices, allowing one device to control another.

Key Focus

- Using wireless networks
- Communicating between devices

Pre-lesson Checklist

For the teacher:

- Computer with mBlock 5 installed or mBlock Web version
- Two (2) CyberPi's with USB-C cable **or** One (1) CyberPi and One (1) Halocode
- Pocket Shield (optional)
- Example program(s): CyberPi – Lesson 9 – Gift Alarm

For the student:

- Computer with mBlock 5 installed or [mBlock Web version]
- Two (2) CyberPi's with USB-C cable **or** One (1) CyberPi and One (1) Halocode
- Pocket Shield (optional)
- Example program(s): CyberPi – Lesson 9 – Gift Alarm

Content Standards

Type	Indicator	Standard
CSTA	2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
ISTE	5c	Students break problems into component parts, extra key information, and develop descriptive models to understand complex systems or facilitate problem-solving.
K12 CS Framework	Practice 3-1	Identify complex, interdisciplinary, real-world problems that can be solved computationally.
K12 CS Framework	Practice 3-2	Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures.
K12 CS Framework	Practice 3-3	Evaluate whether it is appropriate and feasible to solve a problem computationally.

Agenda (45 minutes)

Duration	Content
5 minutes	**Warm-up** • Internet of Things
15 minutes	**Hands-on** • Wireless Communication in mBlock
20 minutes	**Try It** • Creating a Gift Alarm
5 minutes	**Wrap-up** • Brainstorming Ideas • Lesson Extension(s)

Activities

Warm-Up [5 minutes]

Internet of Things

31. Facilitate a short research activity on the Internet of Things. Have students research and present their findings. Their research should lead them to learn more about how internet-connected physical computing devices are used in the following applications:
    a. Smart Homes
    b. Healthcare Monitoring
    c. Transportation
    d. Agriculture
    e. Weather
    f. Manufacturing
    g. Environmental Research
    h. Military

Hands-On [15 minutes]

Wireless Communication in mBlock

9. There are two types of wireless communication in mBlock: Wi-Fi and LAN. Review the following information and blocks for each type.

Wi-Fi		
Using a Wi-Fi connection, data is shared with the cloud message function. You can share data across devices and projects with the same mBlock 5 account. Physical proximity or distance is no longer a restriction, as these devices do not need to be in the same location. To use, each device must be connected to the internet. See the blocks below for connecting the CyberPi, Halocode and the mBlock project to cloud messages.		
Category	**Block**	**Function**
IoT	connect to Wi-Fi (ssid) password (password)	Connects the CyberPi to a wireless network.
	network connected?	Returns TRUE if the CyberPi is connected to the internet.
	when receiving user cloud broadcast (message)	Event block that triggers the execution of the actions attached

		when a specific cloud broadcast is received.
	send user cloud broadcast (message)	Sends a cloud broadcast.
The CyberPi can communicate with a Halocode or an mBlock sprite through Wi-Fi cloud broadcast.		
Halocode Wi-Fi	connect to Wi-Fi (ssid) password (password) Wi-Fi connected? when receiving user cloud message (message) broadcast user cloud message (message)	See description of CyberPi blocks above.
Sprite User cloud message	when I receive user cloud message (message) send user cloud message (message)	See description of CyberPi blocks above.

LAN		
A LAN (local-area network) is a network that links a group of computers or devices within a certain location. The group of computers share communications to send messages to each other. A local area network can be formed between CyberPi's to allow one CyberPi to control another.		
Category	**Block**	**Function**
LAN	when receiving LAN broadcast (message)	Event block that triggers the execution of the actions attached when a specific LAN broadcast is received.
	broadcast (message) on LAN	Sends a LAN broadcast.

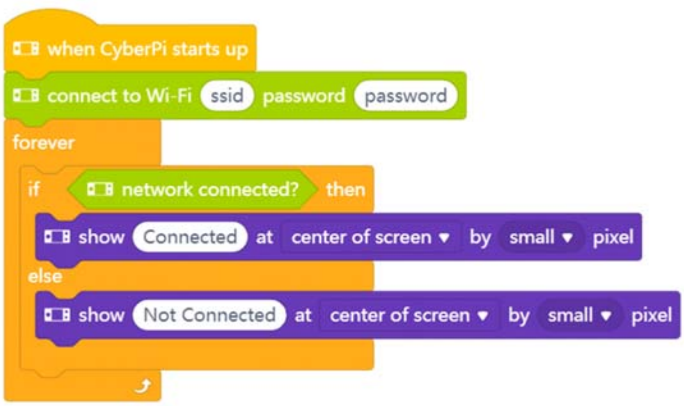
10. Select the type of communication that will work for with your classroom and students. Use the following steps as an example of how wireless communication works in mBlock.

11. Open the mBlock 5 software or [mBlock 5 Web version](). Add the CyberPi in the Devices tab and connect in Upload mode.

12. On the Panda sprite, in the **Block Area**, click the  button. Find the **User cloud message** extension and click **+Add**.
13. Instruct students to build the following scripts:

CyberPi	Sprite
	

14. Update the Wi-Fi ssid and password with the information for the wireless router in your location for the CyberPi to connect.
15. Upload the program to the CyberPi and test the program. Students should observe Panda walk when the CyberPi is shaken.
16. If the program does not appear to work, add the following code to the CyberPi script to troubleshoot the Wi-Fi connection.



Try It
[20 minutes]

Creating a Gift Alarm

1. Using the following problem statement, instruct students to plan a solution to the problem. There are many ways to solve this problem; one sample solution has been provided for you.

Gift Alarm	
Problem	One of your friend's birthday is soon. This friend likes to shake presents to try to figure out what is inside. You would like to figure how to make a silent alarm that will notify you wirelessly if the gift is shaken. How can mBlock, the CyberPi and/or the Halocode help you determine if your friend shakes the present you give them this year?
Proposed Solution	Program the Halocode to send a cloud message if it is shaken. Secure the Halocode and a battery pack inside the present before wrapping. Program the CyberPi to play a sound and flash LEDs if a cloud message is received indicating the gift was shaken. Bonus: Have a sprite in mBlock also shake when the gift is shaken.

2. Have students write pseudocode for the **Gift Alarm** and then create the project using their pseudocode as a guide.

Wrap-Up
[5 minutes]

Brainstorming Ideas

4. Wireless communication and cloud messages eliminate barriers such as cord length and device location. With the class, brainstorm a list of ideas for programs that could benefit from wireless communication. Some ideas may include:
    - Weather station which reports to a separate device.
    - Survey collection device in a main location of the school which reports results to the classroom.
    - Walkie talkie or text messaging between devices.

Lesson Extension(s)

- Have students develop their own project using wireless communication.
- Have students research networks and the role they in society.