Into Cyberspace

Grade 4 | Lesson 1

Lesson Overview

60 mins

During this lesson, students will gain an understanding of what an algorithm is and the characteristics of a well-designed algorithm. They will integrate and exhibit learning by helping Sam turn on and code a start-up sequence for her Cyber Scanner.

Learn Warm-Up Mini-lesson	**Demonstrate** what an algorithm is and the importance of precise language. **Identify** which algorithm is best designed to solve a problem.
Do Let's Build Plan Challenge Debug Chili Challenges	**Plan**, **code** and **extend** a program that will display a start-up sequence for Sam's Cyber Scanner.
Reflect	**Reflect** and **evidence** learning in the Mission Journal.

Standards Focus (Refer to Standards Alignment Map)

Grade 4 Computer Science: Sequences and Algorithms (IB-AP-08)

Materials Required

SAM Labs Learn to Code kit, including micro:bit

Learn

Warm-Up

Demonstrate what an algorithm is and the importance of precise language.

Key Information

"What is an algorithm?"

- An algorithm is a set of step-by-step instructions to write and follow, in order to solve a given problem.
- The language in instructions needs to be precise to ensure a task is properly completed.
- A sequence is a series of algorithmic steps or instructions in a precise order.

Unplugged Activity

- Students create a handshake with at least three elements.
- First they describe their handshake to a partner orally and discuss how easily they followed the instructions.
- Then they write down step-by-step instructions and compare if this was easier for their partner to follow.

Quick Reflection
What can happen if instructions are not clear enough?

Link Forward
Students look at example algorithms and the importance of precise instructions in an algorithm.

Mini-lesson

Identify which algorithm is best designed to solve a problem.

Key Information

"What are algorithms designed to do?"

- Algorithms are designed to complete a task.
- In order to design an algorithm, two questions need to be answered:
  - What is the problem you want to solve?
  - What are the steps to solve it?
- A computer program is a series of processes created to complete a task and/or solve a need.

Unplugged Activity

Compare and Contrast

"Which algorithm would be best to follow to make pizza?"

Students look at the two algorithms and decide which algorithm would produce a pizza and discuss why.

Quick Reflection

Is there a problem in your school that a computer program could solve?

Checks for Understanding

- Which of the following are algorithms that occur in everyday life?
- Which of the following are characteristics of a well-designed algorithm?

Keywords

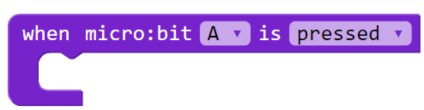
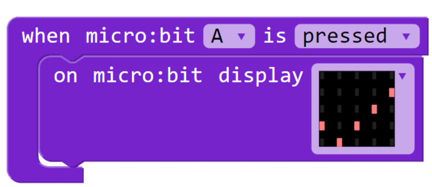
Algorithm	Step-by-step instructions to write and follow, in order to solve a given problem.
Sequence	Following an algorithm from start to finish in a specific order.
Steps	Each action within an algorithm.
Program	A collection of algorithms executed by a computer to perform a specific task.

Link Forward
Students code a start-up sequence to display on the micro:bit.

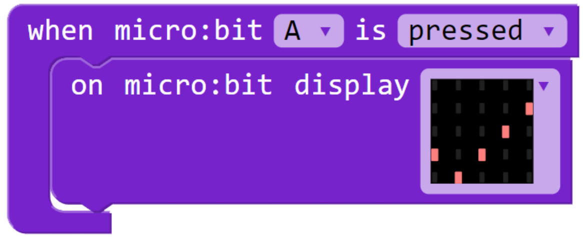
Do

Let's Build

Code a sequence to display an output on the micro:bit.

Instructions	Workspace	Explain that...
Step 1 Click 'ADD DEVICE' and select: ● 'micro:bit'. Connect the battery, click 'CONNECT' and 'Pair'.		The micro:bit screen will enable students to visualize the sequence.
Step 2 From 'micro:bit' 'Events', drag onto the workspace: ● 1 'when micro:bit [A] is [pressed]' block.		If the 'A' button on the micro:bit is pressed, an event is activated.
Step 3 From 'micro:bit' 'Actions', drag onto the workspace: ● 1 'on micro:bit display [heart]' block. Set to 'yes'. Snap into the 'when micro:bit [A] is [pressed]' block.		Students select from the drop-down of display options. The 'yes' option is a check mark shape. 'Snap' refers to the way blocks connect on the workspace.

Step 4
Click ‘RUN’ and test your program.



Encourage students to test that, when the ‘A’ button is pressed, the micro:bit displays a check mark.

Explain that this sequence will be used throughout the lessons to test the micro:bit and thereby Sam’s Cyber Scanner is functioning.

Quick Reflection
What happens if another option is selected from the drop-down?

Plan

Recap Sam’s story so far using the Lesson Slides. Students can then use the Mission Journal to complete planning tasks.

I need to code a start-up sequence for my Cyber Scanner.

Can you help me use images and sound?

Building on the code on the workspace from Let’s Build, students plan a program to display a start-up sequence on Sam’s Cyber Scanner (on the micro:bit).

Challenge

Code a start-up sequence on Sam's Cyber Scanner.

<table>
<tr><th>Instructions</th><th>Workspace</th><th>Explain that...</th></tr>
<tr><td>Step 1
Click 'ADD DEVICE' and select:
● 1 Buzzer.

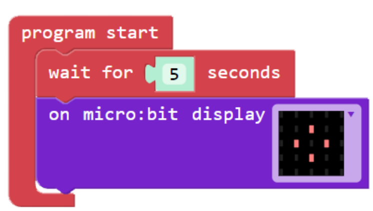
Turn the block on, click 'CONNECT' and 'Pair'.</td><td>
</td><td>The Let's Build program should remain on the workspace. Both programs are activated by 'RUN'.

The Buzzer will be the output which will alert students that the start-up sequence is working.</td></tr>
<tr><td>Step 2
From 'General', drag onto the workspace:
● 1 'program start' block.</td><td>
</td><td></td></tr>
<tr><td>Step 3
From 'General', drag onto the workspace:
● 1 'wait for (2) seconds' block.

Snap into the 'program start' block.

Set to '5 seconds'.</td><td>
</td><td>The shorthand '( )' refers to sections in blocks where values are entered. '[ ]' refers to drop-down/pre-set options.</td></tr>
<tr><td>Step 4
From 'micro:bit' 'Actions', drag onto the workspace:
● 1 'on micro:bit display [heart]' block.

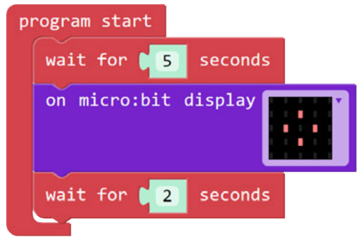
Snap into the 'wait (5) seconds' block.

Set to 'small diamond'.</td><td>
</td><td>This display will be used to simulate the 'Cyber Scanner' switching on.

When the program runs, if the micro:bit is already displaying a check mark, it will change after 5 seconds to a small diamond.</td></tr>
</table>

Step 5
From 'General', drag onto the workspace:
- 1 'wait for (2) Seconds' block.

Snap into the 'on micro:bit display' block.



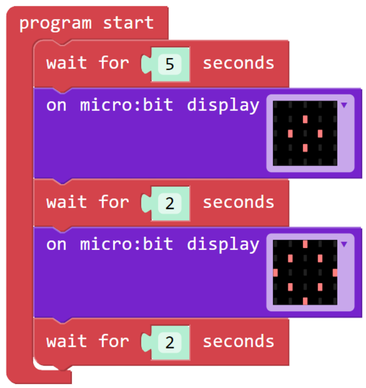
The use of the 'wait for (2) seconds' block allows the image to stay on the display before changing to the next.

The default setting of the block is 2 seconds.

Step 6
Duplicate:
- 'on micro:bit display' block.
- 'wait for (2) Seconds' block.

Snap the blocks into the 'wait for (2) seconds' block.

Set the new 'on micro:bit display' to 'diamond'.

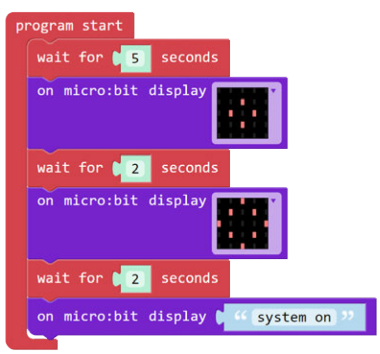


To duplicate, students right-click and select 'duplicate' from the options. This can save time.

Step 7
From 'micro:bit' 'Actions', drag onto the workspace:
- 1 'on micro:bit display ("word")' block.

Snap into the last 'wait for (2) seconds' block.

Enter 'system on' into the text field.



The field at the end of the block allows text to be entered which will then be displayed in a scrolling effect across the micro:bit display.

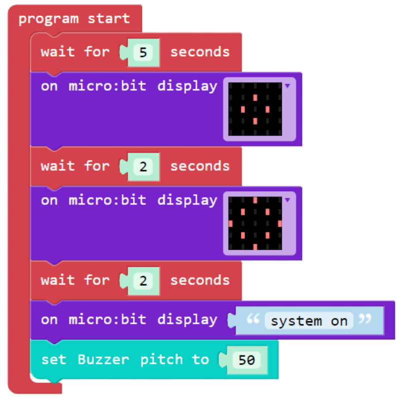
Text can be entered as either upper or lowercase.

Step 8
From ‘Buzzer’ ‘Actions’, drag onto the workspace:

- 1 ‘set Buzzer pitch to (0)’ block.

Snap into the ‘on micro:bit display “(system on)”’ block.

Set to ‘50’.



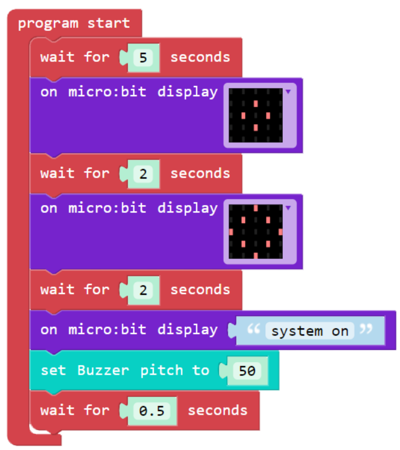
Students can experiment with the pitch value of the Buzzer.

Step 9
Duplicate:

- 1 ‘wait for (2) Seconds’ block.

Snap into the ‘set Buzzer pitch to (50)’ block.

Set to ‘0.5 seconds’.

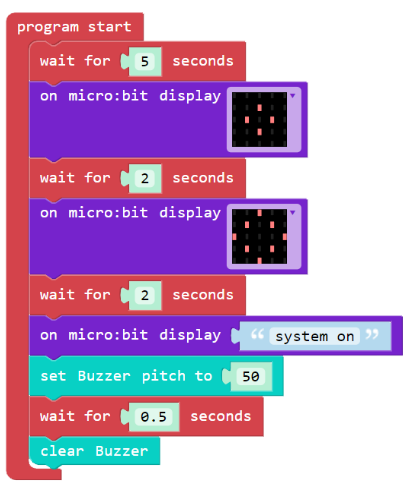


The purpose of the ‘wait for (0.5) seconds’ block is to allow the Buzzer to sound for a set amount of time.

Step 10
From ‘Buzzer’ ‘Actions’, drag onto the workspace:

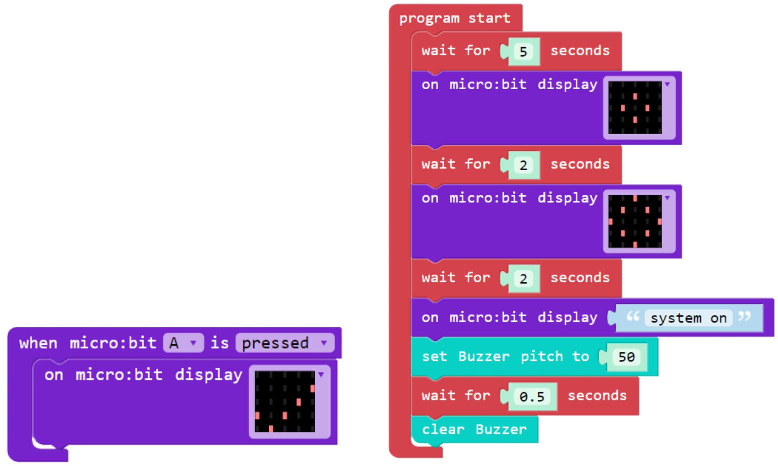
- 1 ‘clear Buzzer’ block.

Snap into the ‘wait for (0.5) seconds’ block.



The ‘clear Buzzer’ block acts as an endpoint to the sound; without it, the Buzzer would sound continuously.

Step 11
Click 'RUN' and test your program.

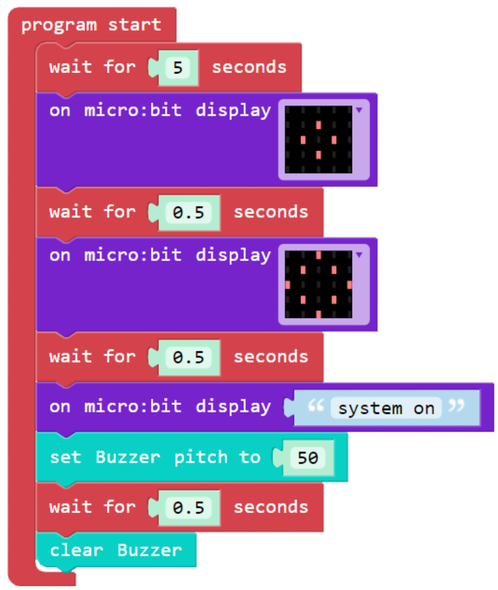


5 seconds after the program is started, the micro:bit will display a small then a larger diamond, then 'system on'. Finally, the Buzzer will sound for 0.5 seconds before ceasing. At any point, if the micro:bit 'A' button is pressed, the micro:bit will display a check mark.

Quick Reflection
Does your program look the same as mine?
Students follow the flowchart on slides.

Debug

The program is running too slowly. How can I debug it?

Instructions	Workspace
Experiment with the settings of the 'wait' blocks. Determine a suitable time delay for each part of the algorithm.	 Encourage students to experiment with the time settings and the effect on the outputs when they are increased, decreased or removed.

Quick Reflection

What did you find out from experimenting with the time settings? What was the impact on the program?

Checks for Understanding

- Which of the below describes the start-up sequence you coded?
- Which is an example of an everyday start-up sequence?

Chili Challenges

Students self-select or teacher can assign an extension activity.

- Experiment further with the 'wait' blocks and the 'on micro:bit display ("word")' block. Can you perfect the Cyber Scanner start-up sequence?
- Experiment with other inputs, such as button 'B' on the micro:bit. Can you code a start-up sequence that is activated 'when micro:bit (B) is pressed'?
- Experiment with 'Loops'. Can you code a program that runs continuously?

Reflect

Students can complete activities in the Mission Journal.

