



DesignNews

MicroPython Embedded Applications

DAY 3 : Converting Blockly Code to MicroPython

Sponsored by



Webinar Logistics

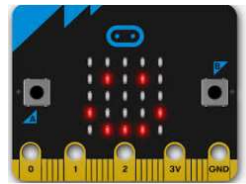
- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click "Help" or submit a question asking for assistance.
- Participate in 'Group Chat' by maximizing the chat widget in your dock.

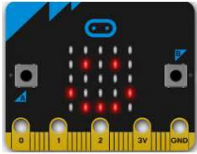


Don Wilcher

Visit 'Lecturer Profile' in your console for more details.

Course Kit:
Keystudio 37 in 1 Starter Kit with BBC micro:bit





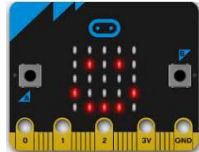
Agenda:

- Review of Visual Programming Language (VPL)
 - a) What is Blockly?
 - b) Blockly Language Examples
 - i. Microsoft Makecode: micro:bit Blockly Code to Python Converter
 - ii. EduBlock: Blockly Code to MicroPython Converter
- Lab Activity
 - a) Blink and Breath Device

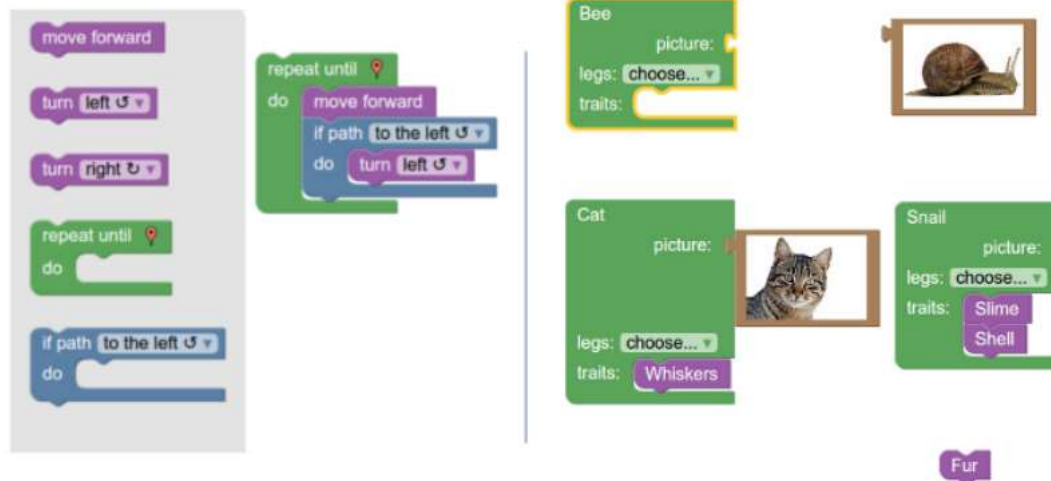
Review of VPL

What is a Visual Programming Language (VPL)?

- It is a programming language
- Allows a user to create and manipulate programs using graphical icons.



Example: Blockly Games



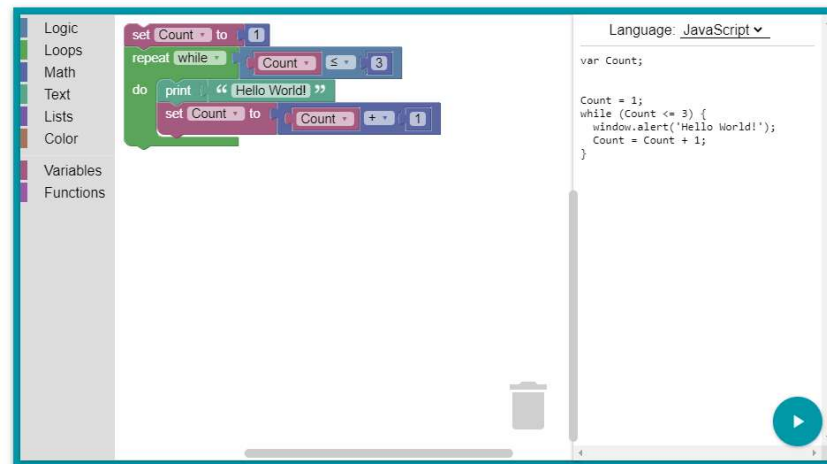
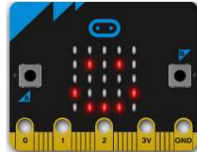
Source: Pasternak, E., Fenichel, R., & Marshall, A. N. (2017). *Tips for creating a block language with blockly*.
<https://developers.google.com/blockly/publications/papers/TipsForCreatingABlockLanguage.pdf>

Review of VPL . . .

What is Blockly?

- An open-source developer library
- Adding block-based coding to an app.
- First released in May 2012.
- Under active development as of 2017.

Try Blockly



Source: <https://developers.google.com/blockly>

Question 1

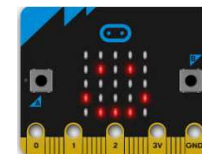
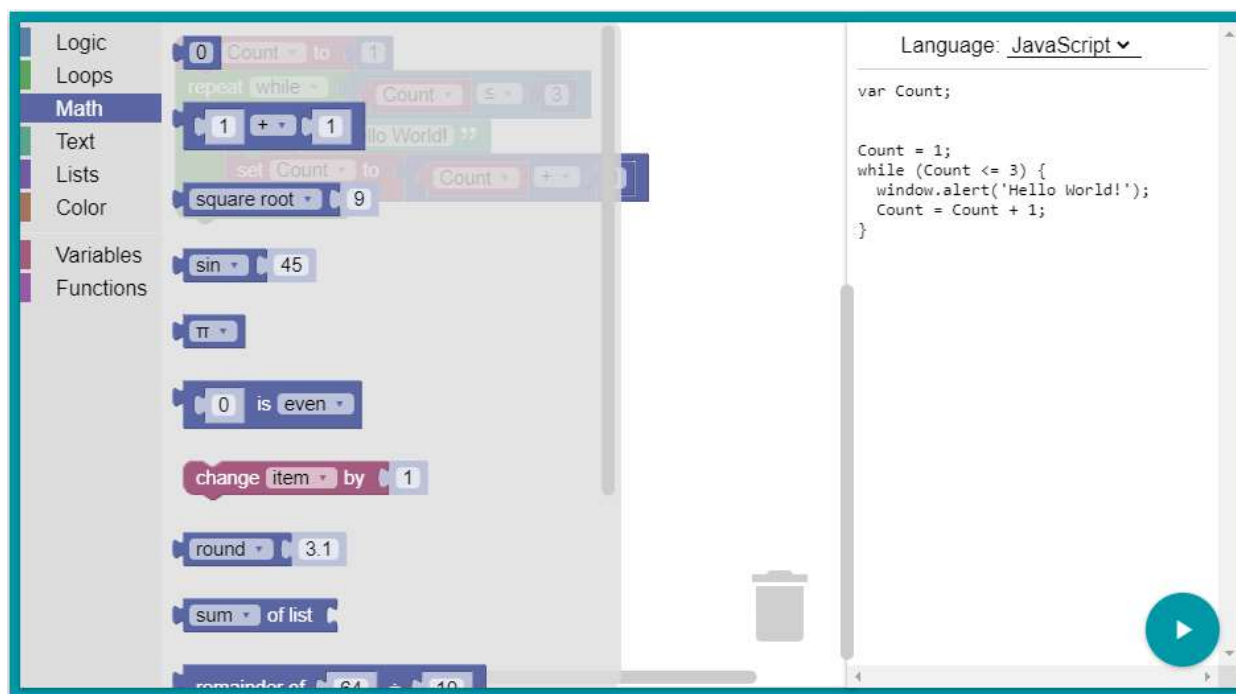


What is a Visual Programming Language?

Review of VPL. . .

What is Blockly?

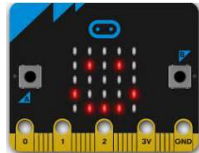
Example:
Math Blockly Blocks



Source: <https://developers.google.com/blockly>

Review of VPL. . .

What is Blockly?



Output Response

Blockly

Try Blockly

...d page at developers-dot-devsite-v2-prod.appspot.com says
Hello World!

OK

Language: Python

```
Count = None

Count = 1
while Count <= 3:
    print('Hello World!')
    Count = Count + 1
```

Click Run button with mouse

Source: <https://developers.google.com/blockly>

Question 2

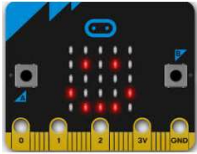


Which statement is incorrect regarding Blockly?

- a) An open-source developer file**
- b) Adding block-based coding to an app**
- c) First released in May 2012**
- d) Under active development as of 2017**

Review of VPL. . .

What is Blockly?



Output Response

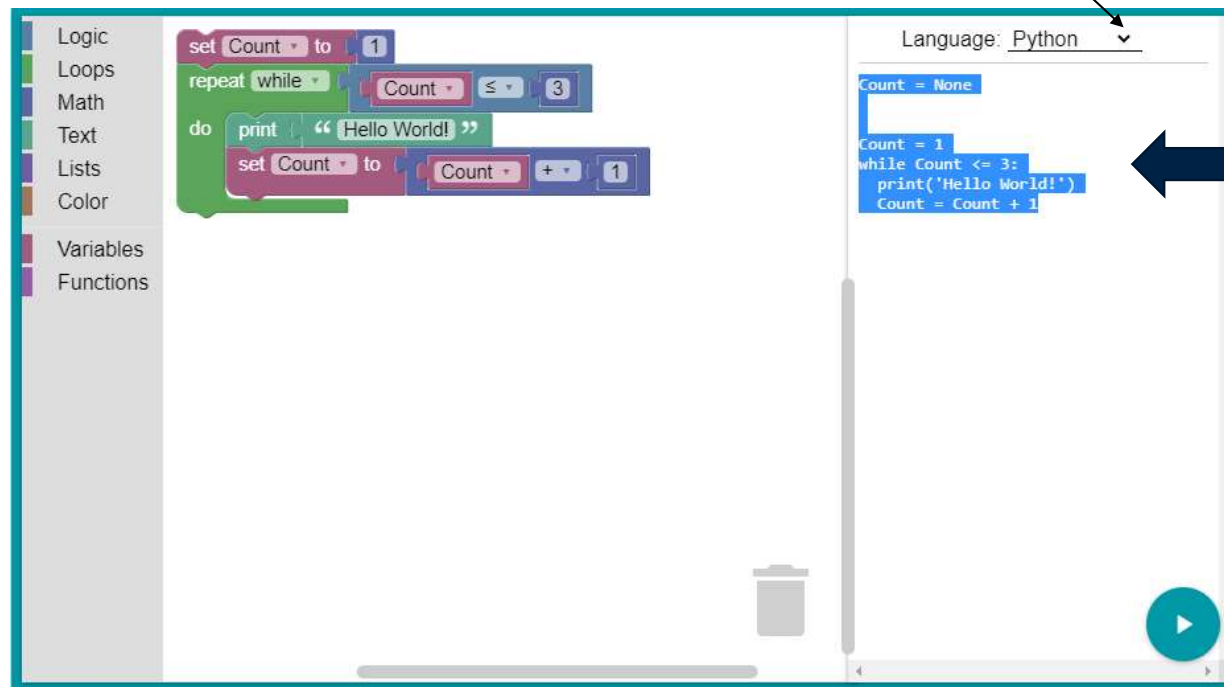
...d page at developers-dot-devsite-v2-prod.appspot.com says
Hello World!

OK

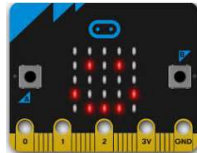
Source: <https://developers.google.com/blockly>

Review of VPL. . .

What is Blockly?



Select Python Code Here!

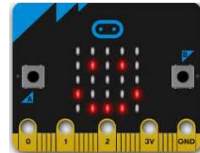


Copy Python Code

Source: <https://developers.google.com/blockly>

Review of VPL. . .

What is Blockly?



Paste Python code
into Mu Code Editor



Click the



button to see

Output result



```
Hello_Python_code.py
1 Count = None
2
3
4 Count = 1
5 while Count <= 3:
6     print('Hello World!')
7     Count = Count + 1

Running: Hello Python code.py
Hello World!
Hello World!
Hello World!
>>>
```

Source: <https://developers.google.com/blockly>

Review of VPL. . .

What is Blockly?

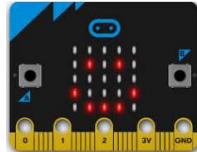
Building a Blockly app

From a user's perspective, Blockly is an intuitive, visual way to build code. From a developer's perspective, Blockly is a ready-made UI for creating a visual language that emits syntactically correct user-generated code. Blockly can export blocks to many programming languages, including these popular options:

- JavaScript
- Python
- PHP
- Lua
- Dart

Here's a high-level breakdown of what goes into building a Blockly app:

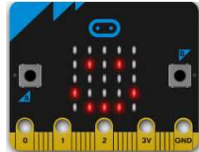
1. **Integrate the Blockly editor.** The Blockly editor at its simplest consists of a toolbox to store block types, and a workspace for arranging blocks. Learn more about integrating Blockly in the [Get Started](#) docs.
2. **Create your app's blocks.** Once you've got Blockly in your app, you need to create blocks for your users to code with, then add them to your Blockly toolbox. Learn how in [Create Custom Blocks Overview](#).
3. **Build the rest of the app.** By itself, Blockly is just a way to generate code. The heart of your app is in deciding what to do with that code.
4. **Give Blockly attribution.** If you'd like to let people know that you used Blockly to build your app, you can grab a Built on Blockly badge from the [Attribution page](#).



Source: <https://developers.google.com/blockly/guides/overview>

Review of VPL. . .

What is Blockly?



Home > Products > Google for Education > Blockly > Guides

Generating Code

Most Blockly applications need to turn blocks into code for execution. This page describes how to add a code generator to a custom block.

First, go to the `generators/` directory and choose the subdirectory that corresponds to the language you want to generate (JavaScript, Python, PHP, Lua, Dart, etc). Assuming your block(s) don't fit in the existing categories, create a new JavaScript file. This new JavaScript file needs to be included in the list of `<script ...>` tags in the editor's HTML file.

A typical block's code generator looks like this:

```
Blockly.JavaScript['text_indexOf'] = function(block) {  
  // Search the text for a substring.  
  var operator = block.getFieldValue('END') == 'FIRST' ? 'indexOf' : 'lastIndexOf';  
  var subString = Blockly.JavaScript.valueToCode(block, 'FIND',  
    Blockly.JavaScript.ORDER_NONE) || '';  
  var text = Blockly.JavaScript.valueToCode(block, 'VALUE',  
    Blockly.JavaScript.ORDER_MEMBER) || '';  
  var code = text + '.' + operator + '(' + subString + ')';  
  return [code, Blockly.JavaScript.ORDER_FUNCTION_CALL];  
};
```

Source: <https://developers.google.com/blockly/guides/create-custom-blocks/generating-code>

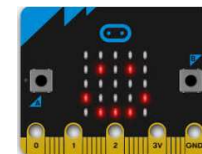
Question 3



Reviewing slide 16, what language is being generated in the block code generator?

Review of VPL. . .

What is Blockly?



Examples:

Projects built with Blockly

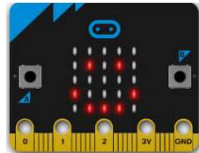
Built with Blockly

Blockly is being used by hundreds of projects, most of them educational:



Source: <https://developers.google.com/blockly>

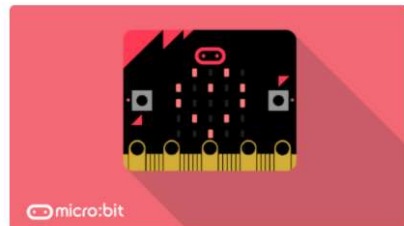
Review of VPL. . .



Blockly Code Example: micro:bit (Microsoft Makecode)

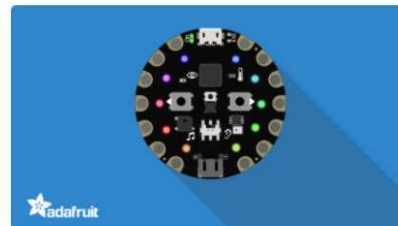
Hands on computing education

Microsoft MakeCode brings computer science to life for all students with fun projects, immediate results, and both block and text editors for learners at different levels.



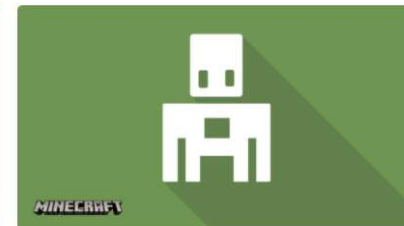
micro:bit

[Learn more with micro:bit >](#)



Circuit Playground Express

[Use Circuit Playground Express >](#)



Minecraft

[Learn more with Minecraft >](#)



LEGO® MINDSTORMS® Education EV3

[Learn more with LEGO MINDSTORMS >](#)



Cue

[Learn more with Cue >](#)



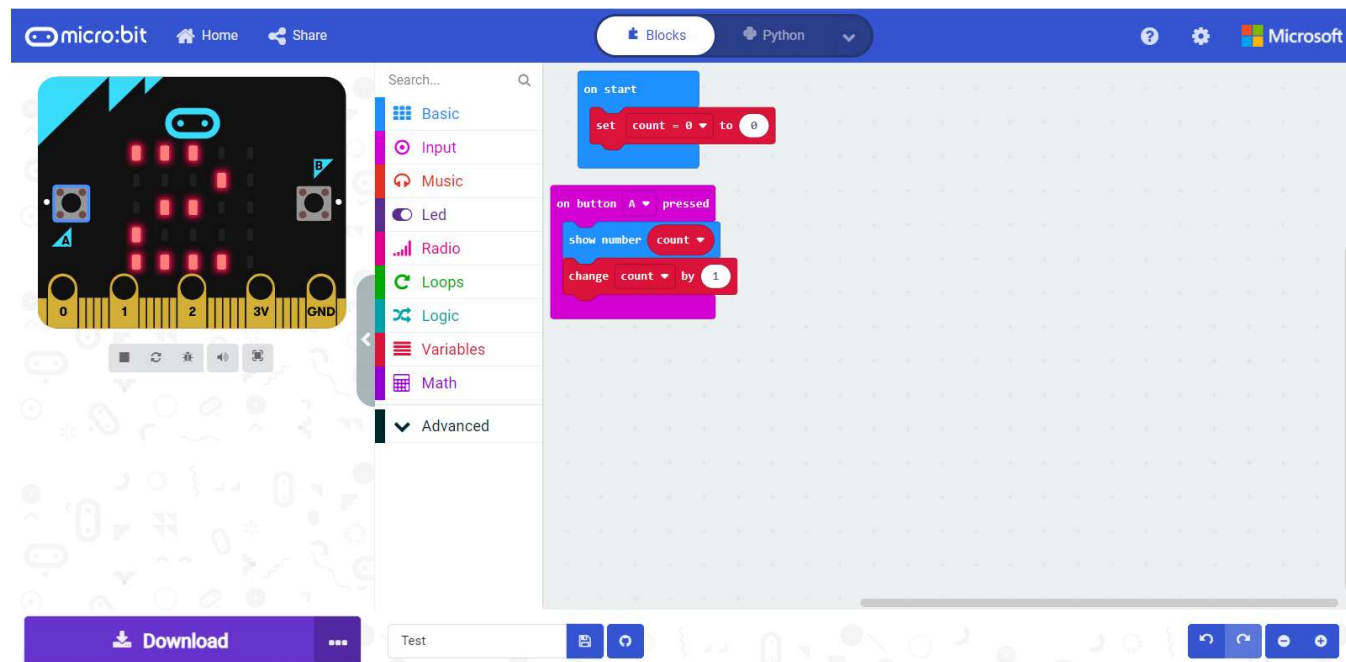
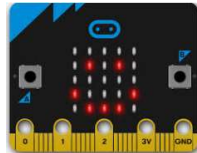
Arcade

[Learn more with Arcade >](#)

Source: <https://www.microsoft.com/en-us/makecode>

Review of VPL. . .

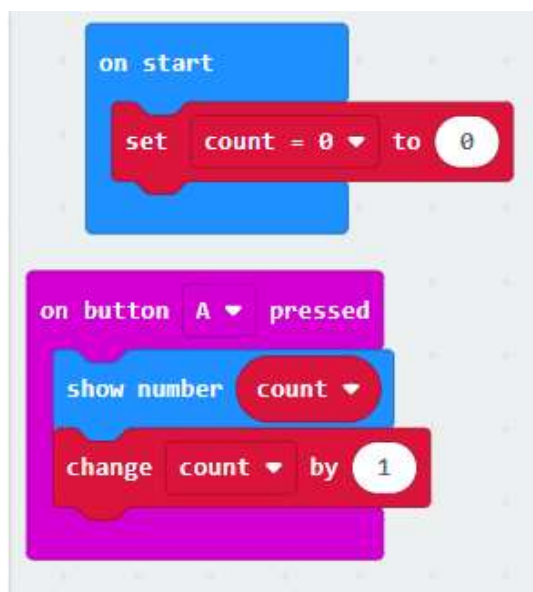
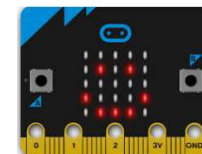
Blockly Code Example:
A simple micro:bit
counter



Source: <https://makecode.microbit.org/#editor>

Blockly Code Example:
A simple micro:bit counter
(Microsoft Makecode)

Review of VPL. . .

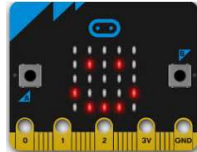


```
1 def on_button_pressed_a():
2     global count
3     basic.show_number(count)
4     count += 1
5 input.on_button_pressed(Button.A, on_button_pressed_a)
6
7 count = 0
8 count__0 = 0
```

Source: <https://makecode.microbit.org/#editor>

Review of VPL. . .

Blockly Code Example:
A simple micro:bit
counter



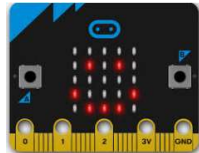
```
1 def on_button_pressed_a():
2     namespace input
3     (Button.A, on_button_pressed_a)
4     Events and data from sensors
5     input.on_button_pressed(Button.A, on_button_pressed_a)
6
7     count = 0
8     count__0 = 0
```

"A **namespace** is a collection of currently defined symbolic names along with information about the object that each name references. You can think of a namespace as a dictionary in which the keys are the object names and the values are objects themselves"(Real Python, n.d).

Source: <https://realpython.com/python-namespaces-scope/#:~:text=Remove%20ads-,Namespaces%20in%20Python,values%20are%20the%20objects%20themselves.>

Review of VPL. . .

Blockly Code Example:
Microsoft Makecode
micro:bit Python
Documentation



Python

The Microsoft MakeCode programming environment uses Python along with the [JavaScript](#) language.

These topics give a brief introduction to Python with MakeCode:

- [Calling](#) - How to use a function
- [Sequencing](#) - Ordering statements in code
- [Variables](#) - Remember data and save values
- [Operators](#) - Operations to change and compare values
- [Statements](#) - The elements of code that take action
- [Functions](#) - Portions of code to use again and again
- [Classes](#) - Contain related data and operations together

Source: <https://makecode.microbit.org/python>

Question 4

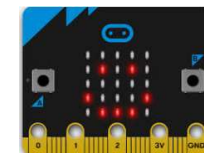


A namespace is a _____ of currently _____ defined:

- a) objects, code**
- b) block, instructions**
- c) collection, code**
- d) collection, symbolic names**

Review of VPL. . .

Blockly Code: EduBlocks to
MicroPython Example. A simple
micro:bit counter



EduBlocks Code



Code Generator

```
Blockly.JavaScript['text_indexOf'] = function(block) {  
    // Search the text for a substring.  
    var operator = block.getFieldValue('OP') == 'FIRST' ? 'indexOf' : 'lastIndexOf';  
    var substring = Blockly.JavaScript.valueToCode(block, 'TND',  
        Blockly.JavaScript.ORDER_NONE) + '""';  
    var text = Blockly.JavaScript.valueToCode(block, 'VALUE',  
        Blockly.JavaScript.ORDER_NUMBER) + '""';  
    var code = text + '.' + operator + '(' + substring + ')';  
    return [code, Blockly.JavaScript.ORDER_FUNCTION_CALL];  
};
```

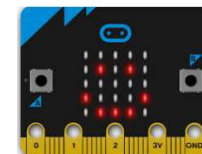
EduBlocks MicroPython

```
1 from microbit import *  
2 pin0.write_digital(0)  
3 while True:  
4     pin0.write_digital(1)  
5     sleep(1000)  
6     pin0.write_digital(0)  
7     sleep(1000)
```

Source: <https://edublocks.org/microbit.html>

Review of VPL. . .

Blockly Code EduBlocks Example:
A simple micro:bit counter



EduBlocks-MicroPython Code

```
1 from microbit import *
2 pin0.write_digital(0)
3 while True:
4     pin0.write_digital(1)
5     sleep(1000)
6     pin0.write_digital(0)
7     sleep(1000)
```

Copy and
Paste to

Mu Code Editor

```
1 from microbit import *
2 pin0.write_digital(0)
3 while True:
4     pin0.write_digital(1)
5     sleep(1000)
6     pin0.write_digital(0)
7     sleep(1000)
```

Source: <https://edublocks.org/microbit.html>

Review of VPL. . .

Blockly Code EduBlocks Example:
A simple micro:bit counter

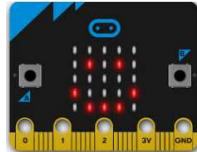
Mini Question and Answer Session with Don!

Why Copy and Paste
EduBlocks' MicroPython
to the Mu Code Editor?

Audience Question



EduBlocks can save application code as a .hex file but not a .py file. Therefore, copy and paste into Mu Code editor will accomplish the latter.



Lab Activity

micro:bit Blink and Breath Device

Check out the
video for device
operation!

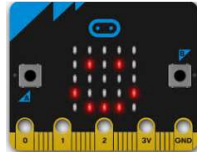
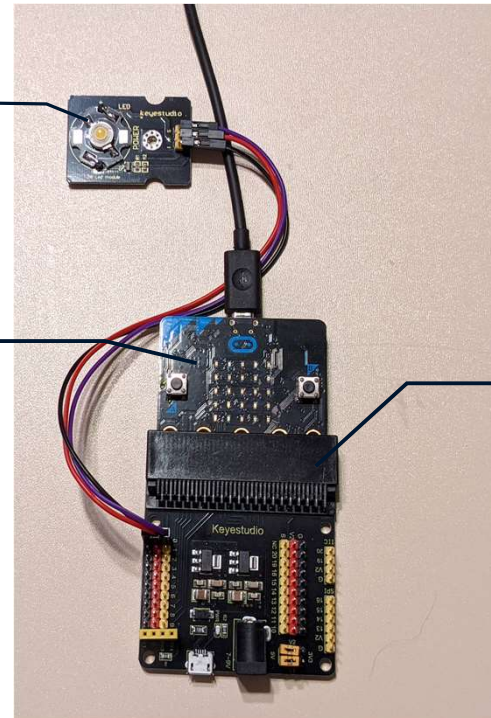
<https://youtu.be/hrvhMs0Ge9g>



3W LED module

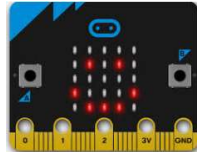
micro:bit

micro:bit Sensor Shield



Lab Activity

micro:bit Blink and Breath Device



3W LED Specifications

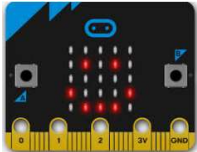
- Supply Voltage: 3.3V to 5V
- Power: 3W
- Light angle: 140 degree
- Working temperature: -50~80°C (-58 ~ 176°F)
- Storage temperature: -50~100°C(-58 ~ 212°F)
- Current: 700~750mA
- Color temperature: 6000~7000K (Cloud Day ~Cool White)



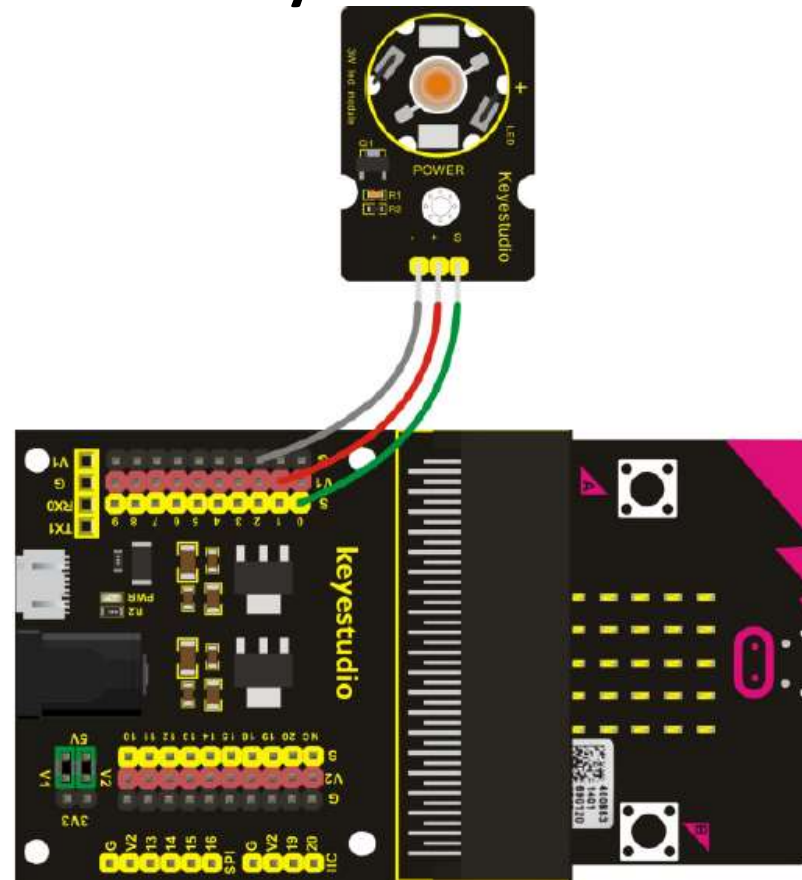
Source: <https://www.dropbox.com/sh/4roisinegqvpy8l/AAApEil-sRDxIISeuDLb0nnoa?dl=0&preview=KS0010+3W+LED.pdf>

Lab Activity

micro:bit Blink and Breath Device



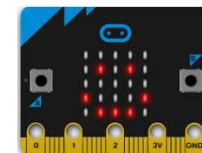
Pictorial Wiring Diagram



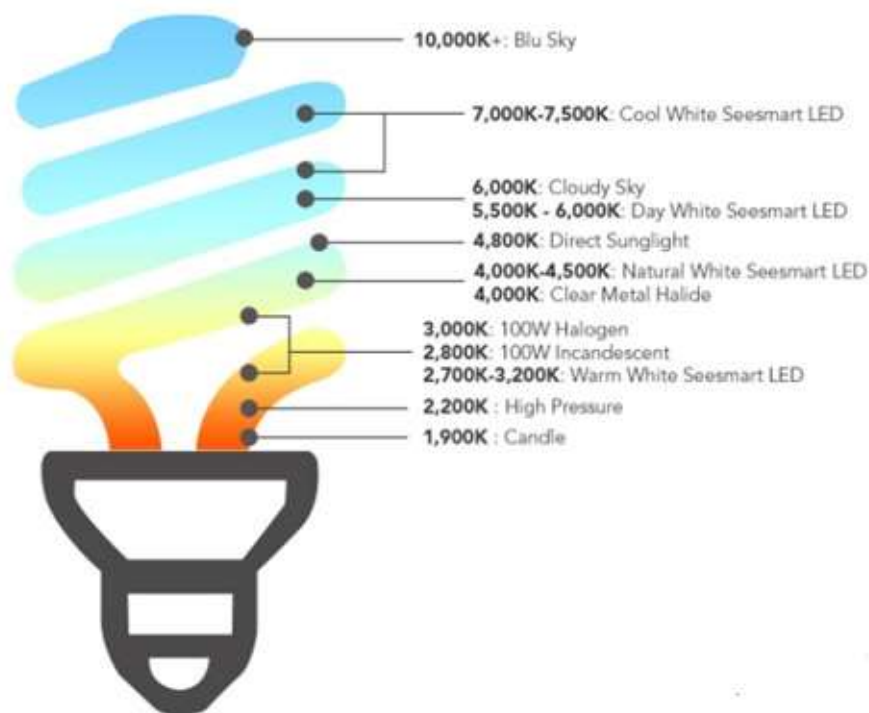
Source:

micro:bit Blink and Breath device

Lab Activity



Color Temperature Chart



Source: <https://www.lumens.com/how-to-and-advice/kelvin-color-temperature.html>

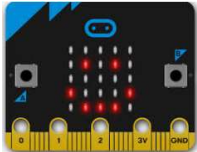
Question 5



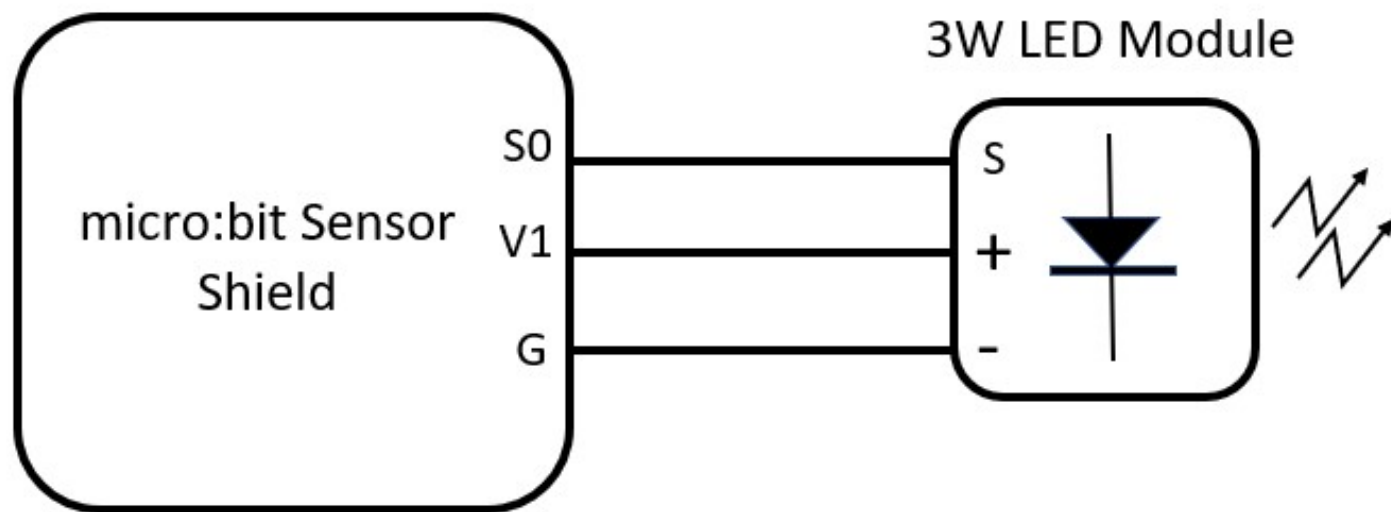
What optoelectronic component is being controlled by the Blink and Breath Device?

Lab Activity

micro:bit Blink and Breath Device



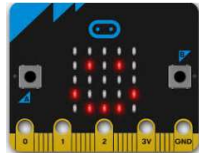
Electrical Wiring Diagram



Source:

Lab Activity

micro:bit Blink and Breath Device
MicroPython Code



```
3 #initialized pin0 and val
4 pin0.write_digital(0)
5 val = 0
6 #main loop
7 while True:
8     for i in range(2): #3W LED will flash 2 times
9         pin0.write_digital(1)
10        sleep(1000)
11        pin0.write_digital(0)
12        sleep(1000)
13    for i in range(2): #3W LED will gradually increase to bright intensity then dim 2 times
14        while val < 1023:
15            val = val + 1
16            pin0.write_analog(val)
17            sleep(5)
18        while val > 0:
19            val = val - 1
20            pin0.write_analog(val)
21            sleep(5)
22
23
```

Question 6



In reviewing MicroPython Code on slide 34, the 1023 value represents what internal component of the nRF51822 microcontroller?

Thank you for attending

Please consider the resources below:

- Google Blockly Code paper
Pasternak, E., Fenichel, R., & Marshall, A. N. (2017). *Tips for creating a block language with blockly*.
<https://developers.google.com/blockly/publications/papers/TipsForCreatingABlockLanguage.pdf>
- Blockly Developer Website
<https://developers.google.com/blockly>
- Building Blockly App
<https://developers.google.com/blockly/guides/overview>
- EduBlocks: micro:bit
<https://edublocks.org/microbit.html>
- Blink and Breath Device YouTube Video
<https://youtu.be/hrvhMs0Ge9g>
- 3W LED Module Specification
<https://www.dropbox.com/sh/4roisinegqvpy8l/AAApEil-sRDxllSeuDLb0nnoa?dl=0&preview=KS0010+3W+LED.pdf>
- Color Temperature Chart
<https://www.lumens.com/how-tos-and-advice/kelvin-color-temperature.html>



DesignNews

Thank You

Sponsored by

