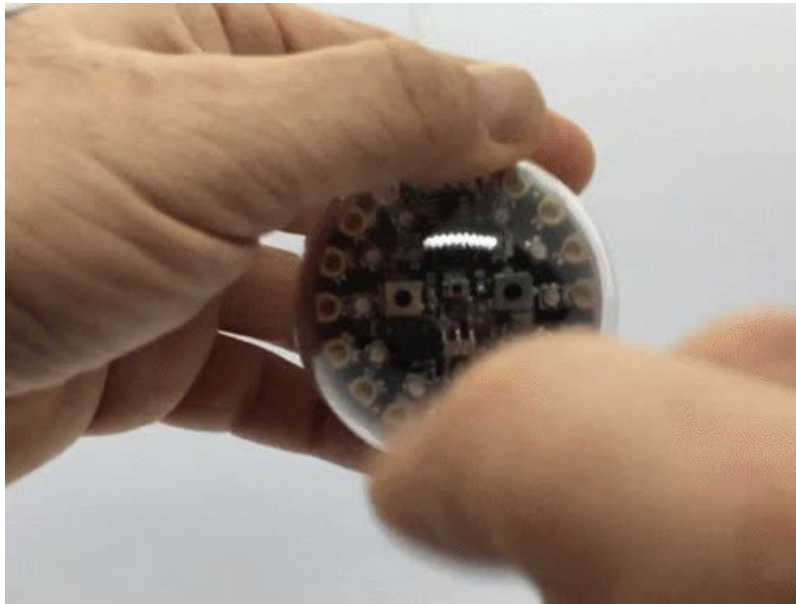




MicroBlocks Circuit Playground Express Ornament

Created by Dave Astels

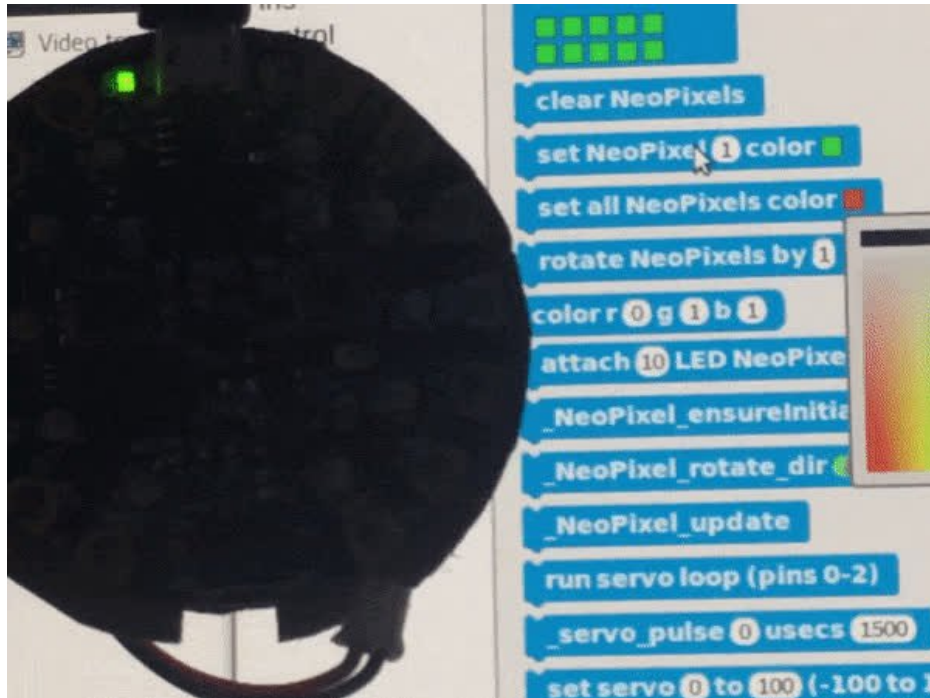


Last updated on 2018-11-28 07:50:43 PM UTC

Guide Contents

Guide Contents	2
Overview	3
Circuit Playground Express	3
Lithium Ion Polymer Battery - 3.7v 500mAh	4
USB cable - USB A to Micro-B	4
DIY Ornament Kit - 6cm Diameter - Perfect for Circuit Playground	4
Exploring MicroBlocks	5
Installation	5
Connecting	6
Real Time Execution	7
Extensions	8
How It Works	8
Making an Ornament	10
The code	11

Overview



Block based programming environments are becoming very popular. First Scratch and now MakeCode, and now there are others like EduBlocks (block based Python) and MicroBlocks.

MicroBlocks is an especially interesting one. It doesn't use an intermediate language like some others; it compiles directly to virtual machine bytecode.

MicroBlocks has three main components:

1. The block based editor
2. The bytecode compiler
3. The code manager that can keep the code on your board seamlessly up to date with what's in the editor.

Not only does it support whole programs, but any valid snippet can be immediately executed on the board. That includes prototype blocks on the palette as well as code under construction. This lets you interactively see how code works, piece by piece as you work on it. This is much like CircuitPython's REPL, but more powerful. For those with an interest in programming languages, this capability is much like what Smalltalk provides: being able to execute any valid code at any time.

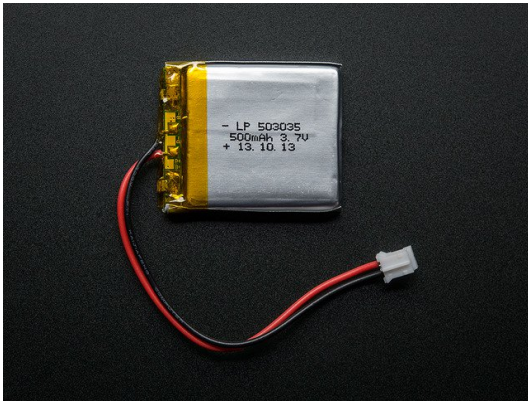
We are going to explore a bit of MicroBlocks through making a festive ornament using a Circuit Playground Express.

Your browser does not support the video tag. [Circuit Playground Express](#)

\$24.95

IN STOCK

ADD TO CART



Lithium Ion Polymer Battery - 3.7v 500mAh

\$7.95
IN STOCK

ADD TO CART



USB cable - USB A to Micro-B

\$2.95
IN STOCK

ADD TO CART



DIY Ornament Kit - 6cm Diameter - Perfect for Circuit Playground

\$1.95
OUT OF STOCK

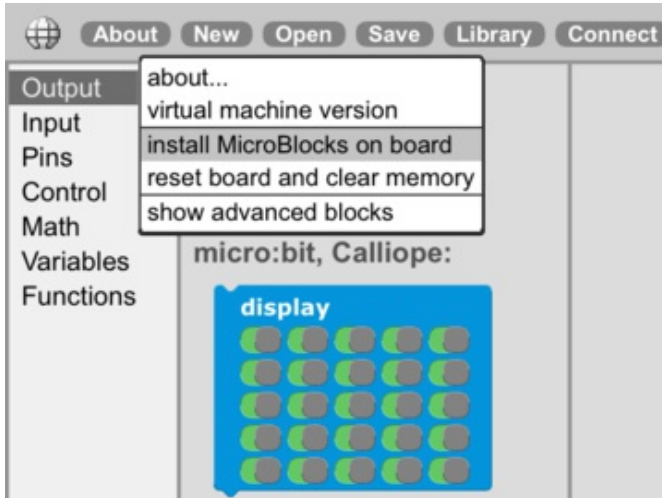
OUT OF STOCK

Exploring MicroBlocks

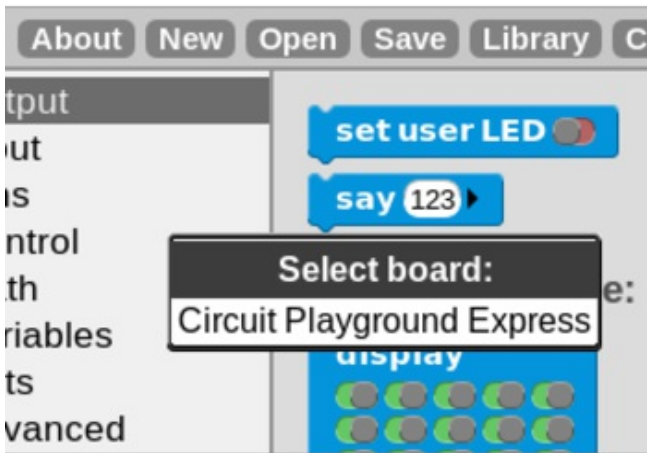
Installation

Unlike MakeCode, MicroBlocks needs to be installed on your computer. While this isn't always a great thing (like where you aren't allowed to install software, at some schools, for example), it does make possible some interesting capabilities.

On the [Get MicroBlocks \(https://adafru.it/D7h\)](https://adafru.it/D7h) page you can find installers for various platforms. Download and install as is appropriate for your system and launch the app.

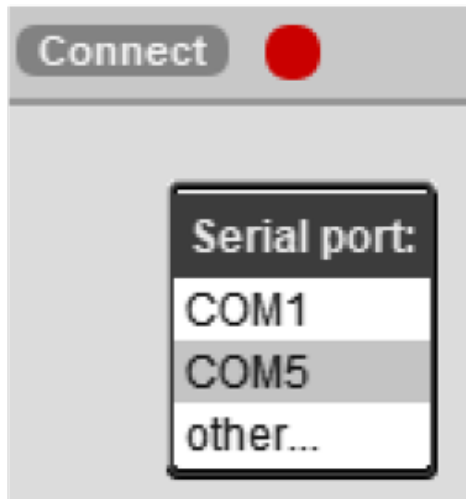


MicroBlocks needs to install its VM (virtual machine) and runtime on your board. In this regard it is much like CircuitPython. Plug in your Circuit Playground Express, double-press the reset button (the small button between the larger A and B buttons). Then select **install MicroBlocks on board** from MicroBlocks' **About** menu.



You'll be asked to select the model of board. Click on it and your board will be prepared.

Connecting



In order to use MicroBlocks with your board, you need to connect them. Use the Connect menu for this. You will be asked to select the serial port to use.

Next to the connect menu button there is an indicator. If it's red you will need to do this connection step, if it's green you're good to go.



When the connection is established, the indicator will change to green.

Real Time Execution

One major difference with MicroBlocks is that it's live. By that, I mean that it's a programming environment like MakeCode but it's also like CircuitPython's REPL.

To illustrate, select the **Output** blocks from the list on the far left. You will see one at the top of the list of blocks just to the right of that. At the top is one to set the user LED (aka the D13 LED).

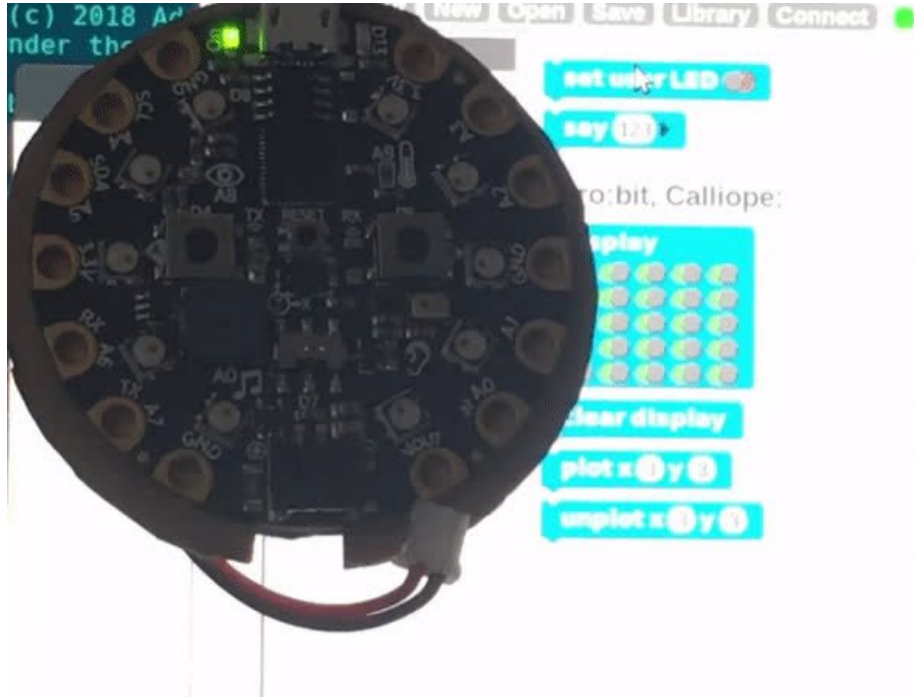


That's pretty cool, and applies generally: if you click on a block anywhere in the IDE, it will get executed immediately

on the board.

Any blocks that are currently running on the board will have a faint outline in the editor pane. As you make changes, they go into effect immediately on the board.

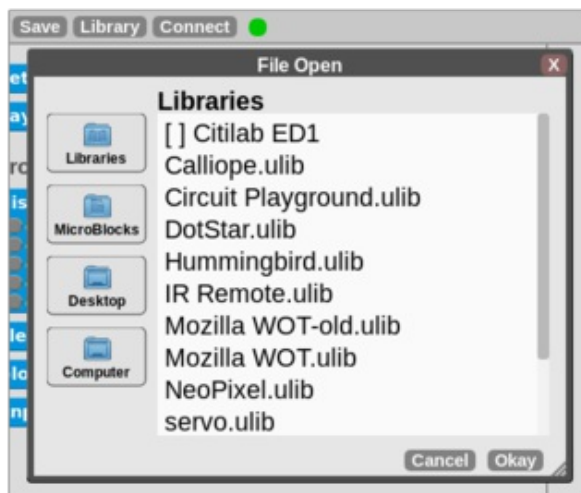
There's no save required, no downloading to the board, it gets updated on the board immediately, in real time, without you having to do anything. That's pretty cool, and very powerful.



Click on the block. Assuming your board is installed and connected (as described above) the D13 LED on your board will light up. Click the slide switch in the block so that it turns red and click it again. The LED on your board turns off.

Extensions

The out of the box experience with MicroBlocks is fairly generic. You have to dig a bit to get access to additional functionality that the Circuit Playground Express has to offer. You do that with the [Library](#) menu.



Library opens a window that gives you access to the bundled libraries, your desktop, and anywhere on your filesystem. When you start out there will only be the bundled libraries (shown when you open the library window).

There are libraries for various supported boards and various bits of hardware. You'll want to load the Circuit Playground and NeoPixel libraries for this project. Select each one in turn and click **Okay**. You will need to reopen the library window each time.

How It Works

MicroBlocks is sort of like MakeCode and EduBlocks, and sort of like CircuitPython. But it's also unlike all of them as well.

The thing that gets installed on your board is the MicroBlocks virtual machine (VM). When you use CircuitPython its VM is installed on your board. That's what reads, compiles to bytecode, and executes your Python code.

MakeCode works a bit differently: your code (which is really JavaScript behind the blocks) is bundled with the runtime and downloaded to your computer as a file that you then copy to your board.

MicroBlocks compiles your program to bytecode, and ships that to your board. the bytecode then gets executed by the VM installed on there.

One big difference is that your block code compiles directly to bytecode, rather than being converted to another language (Javascript or Python) first. This makes it faster and more flexible since it's one step and there's no third language involved.

The other big difference is that as you make changes, they get compiled, downloaded, and executed. The board is always running the latest valid version of your code. This assumes that you clicked **Start** at the top right of the MicroBlocks window.

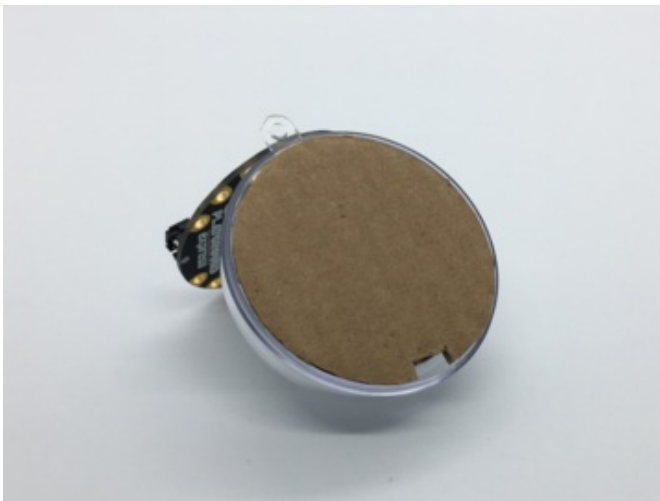
The [MicroBlocks site has a page \(https://adafru.it/D7n\)](https://adafru.it/D7n) explaining a bit about how the whole thing works.

Making an Ornament



Take a pair of the 60mm plastic half-spheres. Be sure to take a pair that fit together.

Since the Circuit Playground Express is about 50.6mm in diameter, it will be too loose on it's own. A simple solution is to trace a half sphere on cardboard and cut a disk to just fit inside. Cut a notch at one spot on the edge for battery wires and attach the Circuit Playground Express on one side and a battery on the other.





Once programming is done, this assembly can be placed inside one half and the sphere snapped together.



The code

First we have to decide what effect we want. Since the Circuit Playground Express has a ring of NeoPixels we can twinkle them in festive colors: red, green, and white.

Since we're going to work with NeoPixels, we need to hook up to them. We can use the `attach` block for that. The Circuit Playground Express has RGB NeoPixels, not RGBW ones, so be sure the `has white` switch is off/red. We can place it in a `when started` block to have it run when the program starts running. When you've done that (as shown below), click the `when started` block. That will compile, download, and execute it. Now we can play with NeoPixels.



We can randomly pick a color for each NeoPixel. We can experiment and get it working with just one pixel.



Remember what was said about clicking on any value block code to have it executed. Well, the above is just that. Make sure you're connected to your board and click the top of the above code (the `set` block in this case). LED 1 (to the left of the USB connector) should light up with one of the colors. Click repeatedly to see one of the three colors randomly appear.

We just need to do that for each pixel (1-10). We can just wrap a `for` block around what we have:



Click the `for` block and see all 10 NeoPixels get set randomly.

Now that we have all ten pixels being randomly set to one of the three colors, we can make it twinkle by doing it repeatedly with a short delay between updates:

```
repeat 50
  for i in 10
    set choice to random 1 to 3
    if choice = 1
      set NeoPixel i color [white]
    else if choice = 2
      set NeoPixel i color [green]
    else if choice = 3
      set NeoPixel i color [red]
    wait 10 millisecs
```

We don't want NeoPixels flashing constantly, that would burn through battery in no time. Since the Circuit Playground Express has a built-in accelerometer and there are blocks to access it (the `tilt` blocks), we can trigger the twinkling effect when the ornament is jostled. Since it's hanging vertically, and the Z axis extends perpendicular to the board, it's reasonable to watch it for changes.

If we keep track of readings, we can compare the current value to the previous one. If the difference is large enough we can assume that the ornament has been jostled or tapped.

```
set last_z to 0
forever
  if abs tilt z - last_z > 50
    set last_z to tilt z
  wait 100 millisecs
```

The final step is to put it all together. The constant (50 as shown) that the Z axis reading difference is compared to can be adjusted to make it more or less sensitive.

```

when started
  attach 10 LED NeoPixel strip to pin 8 has white
  set last_z to 0
  forever
    if abs tilt z - last_z > 50
      repeat 50
        for i in 10
          set choice to random 1 to 3
          if choice = 1
            set NeoPixel i color
          else if choice = 2
            set NeoPixel i color
          else if choice = 3
            set NeoPixel i color
          wait 10 millisecs
        clear NeoPixels
      set last_z to tilt z
      wait 100 millisecs
  
```

Here's the code. Unzip this and load it into MicroBlocks.

<https://adafru.it/D7o>

<https://adafru.it/D7o>