



Analog I/O

Robotics Curriculum
IT Adventures



Lesson Overview

- Learn what analog I/O is
- Learn how to use the analog I/O options with the micro:bit and littleBits
- Create a series of programs that run with analog components

Analog I/O Overview

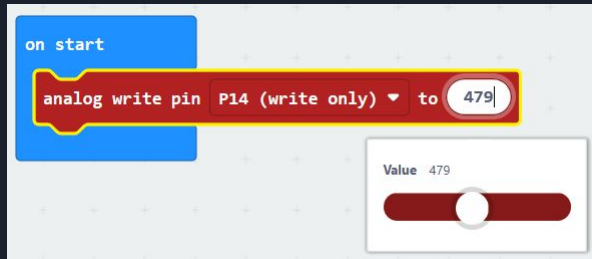
Where digital is considered on or off, analog is any signal that can be multiple values, and is usually a range of values (such as 0 to 5 volts, or 0 to 255). Within the micro:bit code, most analog signals will be from 0 to 1023. To explore this, we will use the following components:

- Slide Dimmer
 - The slide dimmer is a slider that acts as an input. This is an analog input because the incoming value will be the position of the slider within the range, and not just an on/off value.
- Servo
 - The servo is a type of motor that can be placed in a range of positions, from 0 degrees to 180 degrees. To control it, the degree position is set, making this an analog output device.



Make sure to use the power bits when using these components (p3 or p7)!

Code Example - Standard Analog I/O



To the left is an example of writing an analog value to pin P14. Similar to the digital writing of the last lesson, that line is all you need to write a value. However, the difference is that where a digital value is either 0 or 1, the analog values can range from 0 to 1023. Go ahead and mess around with reading from/writing to analog pins!

One component you should use is a slide dimmer - as a sliding scale, it can effectively act as an analog input. Write the maximum analog value to one port, connect the slider to it and the other end of the slider to a different port. Show the value on the LEDs, how does it change when you shift the slider?

Code Example - Servo Control



```
forever
  analog write pin P13 (write only) to 1023
  pause (ms) 5000
  analog write pin P13 (write only) to 256
  pause (ms) 5000
```

The image shows a Scratch-style code editor with a white background. A blue 'forever' loop block contains four blocks: two red 'analog write pin' blocks and two blue 'pause (ms)' blocks. The first red block is set to 'P13 (write only)' and '1023'. The first blue block is set to '5000'. The second red block is set to 'P13 (write only)' and '256'. The second blue block is set to '5000'.

Time to work with motors! Servo motors are special electrical motors that can swing back and forth and turn to certain positions. How fast they turn or which position they turn to are determined by an analog signal sent to them!

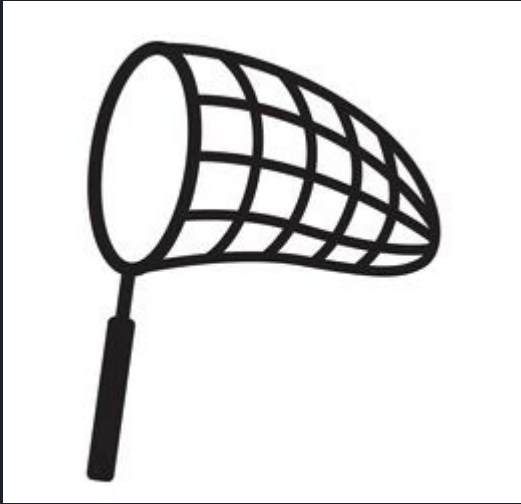
Implement the program on the left and hook up your servo to port 13. What is the difference between swing and turn mode on the component (there is a switch to flip between the two on the littleBit)? Could you control the motor with the slider from before? Try it!

Primary Learning Challenge: Steering Ahead

Using the slide dimmer to control the heading of the robot, drive the robot from one side of the room to the other, ending in a small goal area. The trick here is how you center the RVR's heading with the slider value. Have fun and don't drive too fast!



Secondary Learning Challenge: Bug delivery



Using the slide dimmer to control the heading, like in the primary challenge, drive the robot towards a ping-pong ball which will act as a bug. Once close enough, hit a button to lower the servo (which should have a net or cage) and capture the bug! Deliver the bug to the goal and score a point! Add obstacles on the ground to maneuver around. Think of a way to control your speed so grabbing the ball is easier.