



Adafruit Bonsai Buckaroo

Created by Kattni Rembor



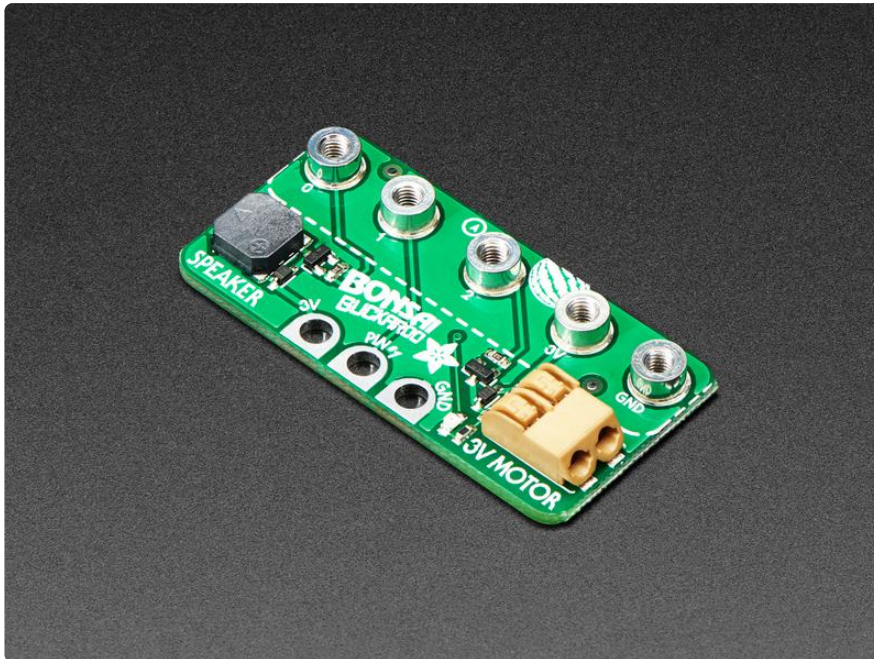
<https://learn.adafruit.com/adafruit-bonsai-buckaroo>

Last updated on 2023-08-29 04:26:22 PM EDT

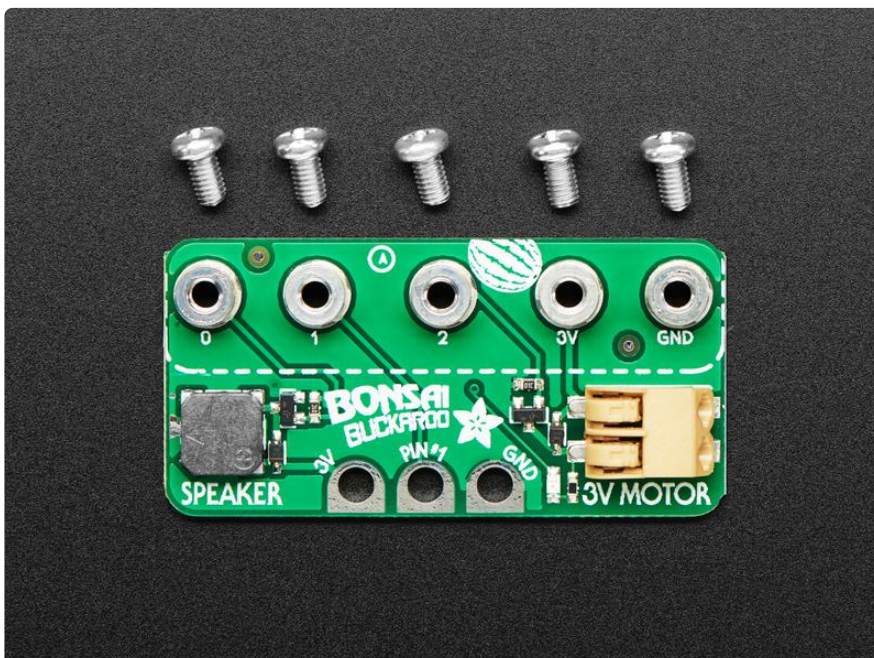
Table of Contents

Overview	3
Pinouts	5
CLUE and CircuitPython Usage	7
<ul style="list-style-type: none">• Wiring up your Bonsai Buckaroo• Wire the Pump Driver Terminal• Installing CircuitPython Libraries• Bonsai Buckaroo Plant Care Helper	
micro:bit and MakeCode Usage	11
Downloads	13
<ul style="list-style-type: none">• Files• Schematic• Fab Print	

Overview

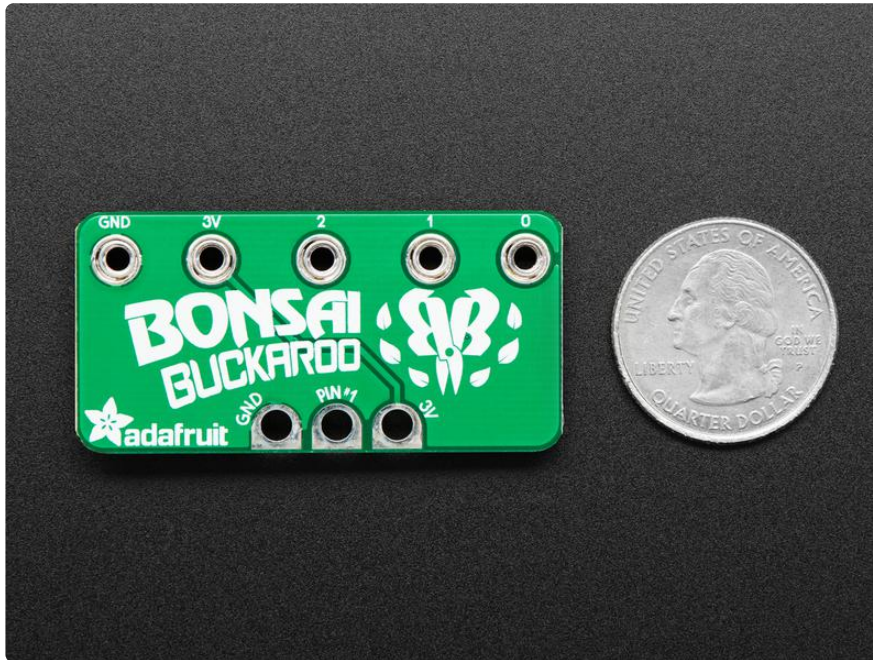


We can't wait for spring to arrive, and we're looking forward to caring for some plants! We designed this little add-on for [micro:bit \(\)](#) or [CLUE \(\)](#) boards – bolt it on with 5 screws to get a buzzer/beeper, motor driver, and breakouts for connecting a [soil sensor \(\)](#) (two alligator clips + nails work just fine). Simple, but effective!



- 8mm Buzzer/Speaker on pin P0 - have your plant kit beep when it wants watering, or have it sing a song when its nice and happy.
- 3V Motor control (on/off) on pin P2 - connect to a water pump to automatically water dry plants

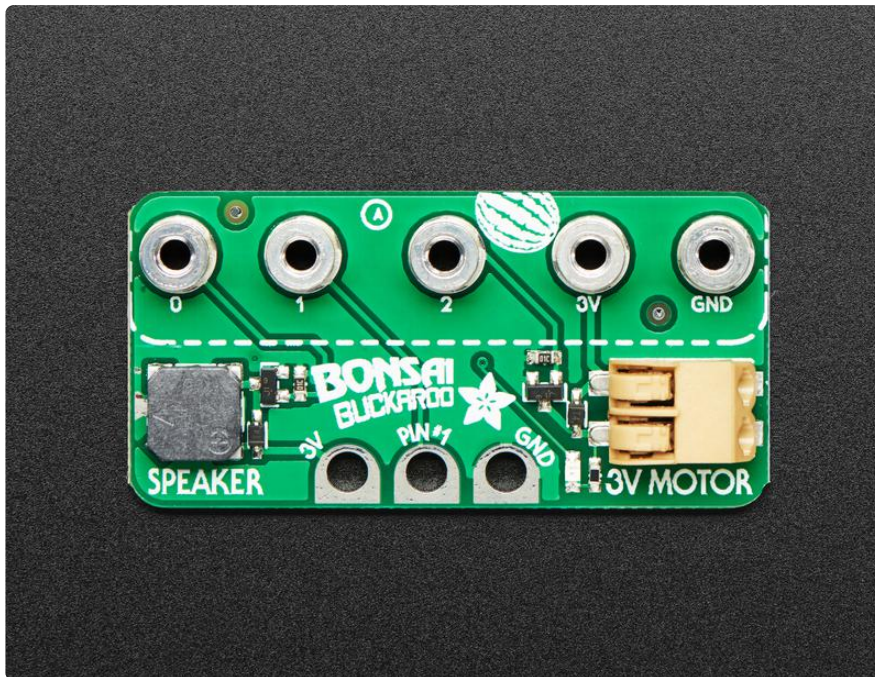
- Alligator clip pads for pin P1, 3V and GND - connect to two alligator clips and nails to measure soil resistance



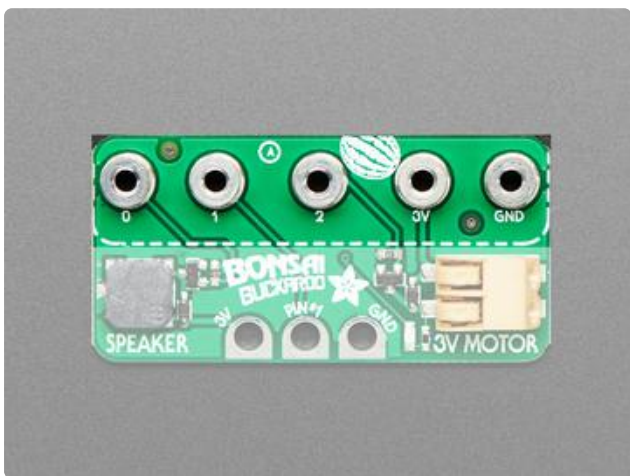
Works with micro:bit in Arduino, MicroPython or MakeCode. For CLUE, you can use CircuitPython or Arduino.



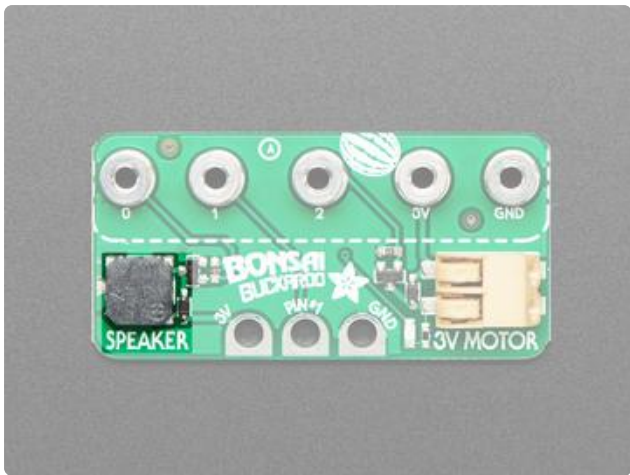
Pinouts



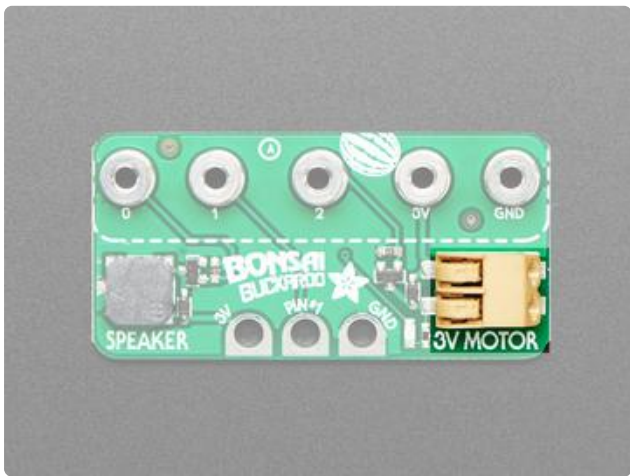
The Bonsai Buckaroo is designed to add a speaker, motor terminal, and pins for connecting a soil sensor to your micro:bit or CLUE board. Let's take a look!



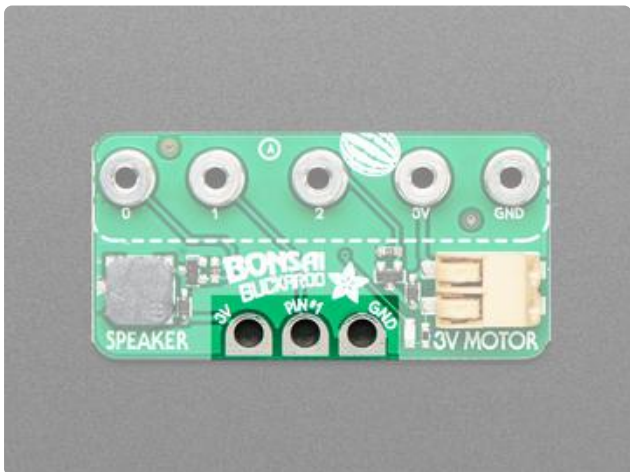
Along the top are the mount points. The Bonsai Buckaroo ships with screws for attaching to your micro:bit or CLUE.



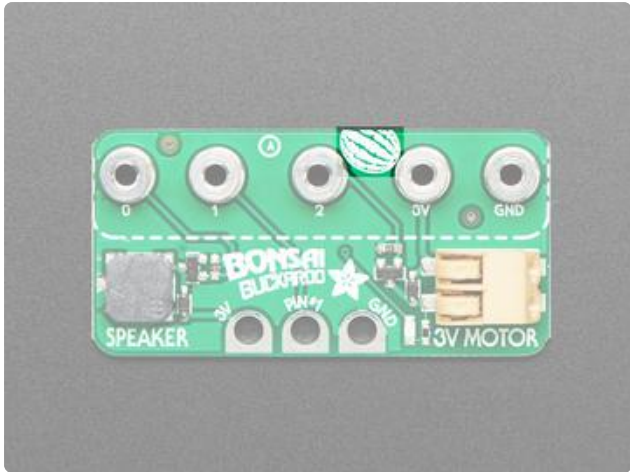
On the left side is the speaker. Use it to have your plant kit beep when it wants watering, or have it sing a song when it's nice and happy.



On the right is the 3V Motor control (on/off) on pin P2. Connect to a water pump to automatically water dry plants. The 3V connector (typically the red wire) goes into the part of the controller terminal towards the top, and the ground (typically the black wire) goes to the part of the controller terminal towards the bottom (by the 3V MOTOR label). However, for our pumps, it does not really matter which way you put the wires in



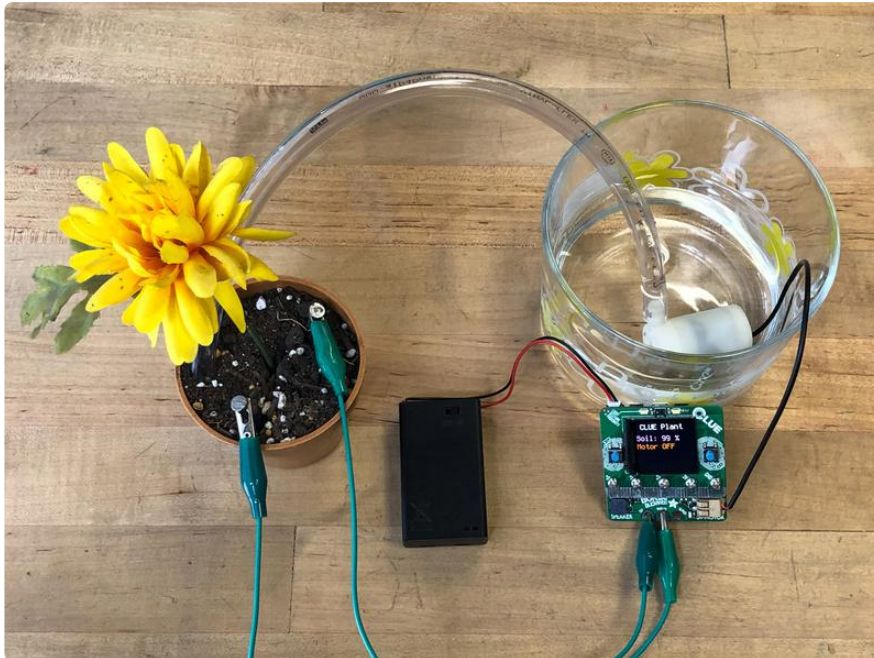
In the center are the Alligator clip pads for pin P1, 3V and GND. Connect to two alligator clips and nails to measure soil resistance as shown on the next pages



Watermelon

We'll tell you about it later.

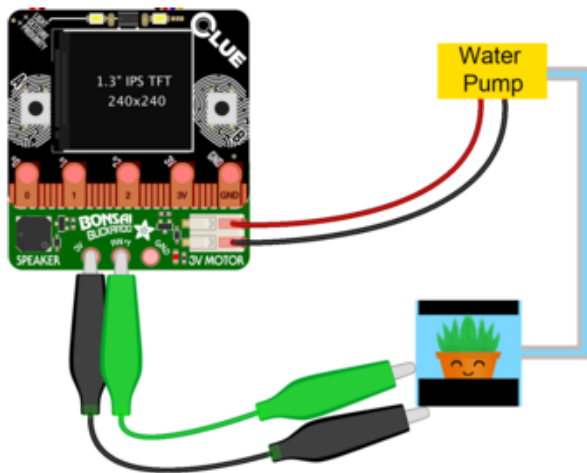
CLUE and CircuitPython Usage



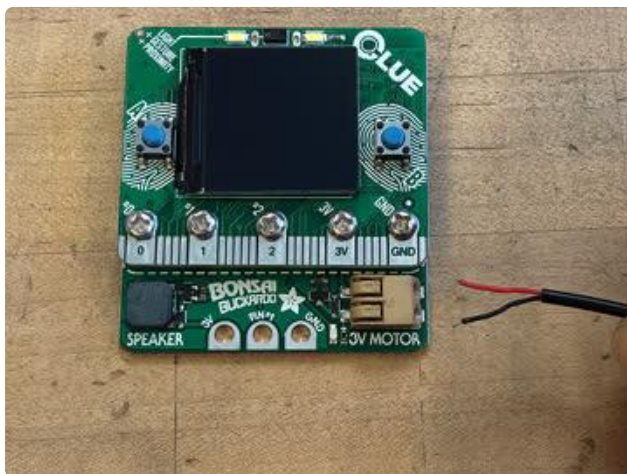
It's easy to use the Adafruit Bonsai Buckaroo with CircuitPython. This example uses the [Adafruit CircuitPython CLUE \(\)](#) library to display soil moisture level and the status of the pump motor on the CLUE display. Connect your Bonsai Buckaroo to your CLUE, and connect the motor and plant to the Bonsai Buckaroo, and you're on your way to automatic plant care. Let's take a look!

Wiring up your Bonsai Buckaroo

For this project, you'll need a Bonsai Buckaroo, a CLUE, a water pump, some tubing, alligator clips, and two nails (or something else conductive will work too!). It's super simple to connect them up.



Mount the Bonsai Buckaroo to the CLUE using the included mounting hardware. Connect the red wire of the water pump to the top of the motor controller terminal. Connect the black wire of the water pump to the bottom of the motor controller terminal. Connect an alligator clip to the 3V pad on the Bonsai Buckaroo. Connect an alligator clip to the PIN#1 pad of the Bonsai Buckaroo. Connect the other end of the alligator clip wires to nails stuck into the soil of your plant. Connect the tubing to the water pump. Place the other end of the tubing in the soil of the plant.



Wire the Pump Driver Terminal

The Bonsai Buckaroo terminal connectors are spring-loaded press-fit connectors. Push down on one tab to separate the internal connector a bit and then push in the wire. Release the pressure on the tab and the terminal will grip the wire.

Repeat for the second terminal.

The image at the top of the page shows the CLUE connected to a AA battery pack. If you find that battery pack isn't lasting long enough for your needs, consider using a larger battery, or using a USB cable plugged into the wall to provide longer usage.

Installing CircuitPython Libraries

The Adafruit CircuitPython CLUE library requires a number of other libraries to work. Please follow the instructions on the [CLUE CircuitPython Libraries \(\)](#) page to ensure you have all the necessary libraries installed.

Before continuing, ensure that your lib folder looks like the one in the CLUE guide Libraries page!

Bonsai Buckaroo Plant Care Helper

Save the following as code.py on your CIRCUITPY drive.

```
# SPDX-FileCopyrightText: 2020 Kattni Rembor for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import board
import digitalio
import analogio
from adafruit_clue import clue

# Turn off the NeoPixel
clue.pixel.fill(0)

# Motor setup
motor = digitalio.DigitalInOut(board.P2)
motor.direction = digitalio.Direction.OUTPUT

# Soil sense setup
analog = analogio.AnalogIn(board.P1)

def read_and_average(analog_in, times, wait):
    analog_sum = 0
    for _ in range(times):
        analog_sum += analog_in.value
        time.sleep(wait)
    return analog_sum / times

clue_display = clue.simple_text_display(title=" CLUE Plant", title_scale=1,
text_scale=3)
clue_display.show()

while True:
    # Take 100 readings and average them
    analog_value = read_and_average(analog, 100, 0.01)
    # Calculate a percentage (analog_value ranges from 0 to 65535)
    percentage = analog_value / 65535 * 100
    # Display the percentage
    clue_display[0].text = "Soil: {} %".format(int(percentage))
    # Print the values to the serial console
    print((analog_value, percentage))

    if percentage < 50:
        motor.value = True
        clue_display[1].text = "Motor ON"
        clue_display[1].color = (0, 255, 0)
        time.sleep(0.5)

    # always turn off quickly
    motor.value = False
    clue_display[1].text = "Motor OFF"
    clue_display[1].color = (255, 0, 0)
```



Let's take a look at the code.

First we import all the necessary libraries, and turn off the NeoPixel on the back of the board.

Next we set up the motor on Pin 2, and setup Pin 1 as an analog pin.

Then we have a helper function designed to read an analog pin, and take a number of readings over a given period of time.

The last thing we do before the loop is set up to display text on the CLUE display using the CLUE library, with the title "CLUE Plant".

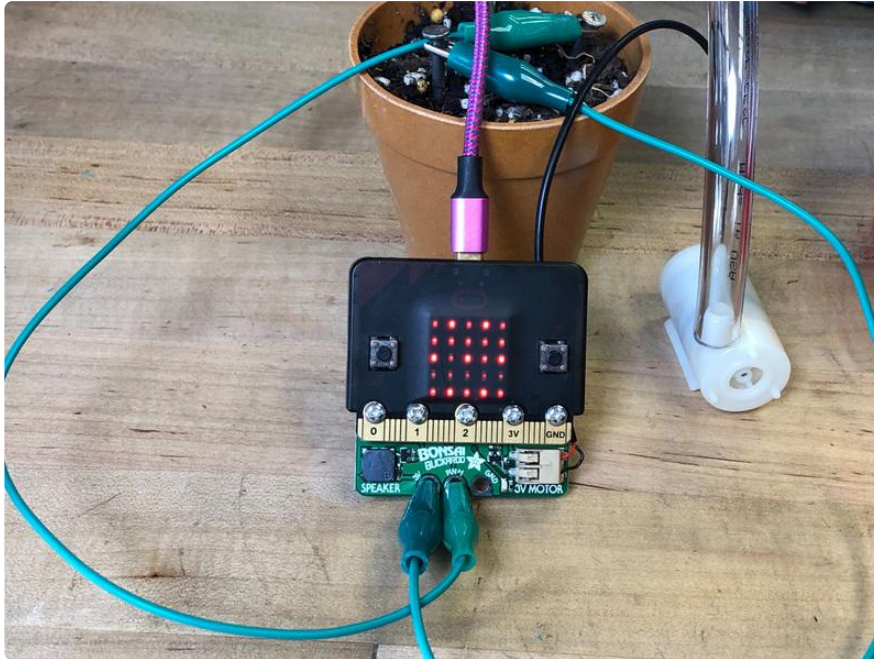
Inside the loop, we first use the helper function to take 100 readings and average them. Then we use that average to calculate a percentage, and display that percentage. We also print the average and percentage to the serial console.

If the percentage is less than 50, we turn on the motor and display "Motor ON" in green, and then pause for 0.5 seconds.

Otherwise, we turn the motor off and display "Motor OFF" in red.

That's all there is to automated plant care using CircuitPython, the Bonsai Buckaroo, and CLUE!

micro:bit and MakeCode Usage



This example uses the Bonsai Buckaroo, the micro:bit, and Microsoft MakeCode to create an automatic plant care system.

If you are new to Microsoft MakeCode, Adafruit has an excellent getting started tutorial at <https://learn.adafruit.com/makecode> ().

This project requires:

- Bonsai Buckaroo
- micro:bit
- water pump
- two alligator clips and two nails (or other conductive materials work as well).

Wire up the parts the same way as the previous page, substituting the micro:bit for the CLUE.

The following is a short program. It reads the value from Pin 1, plots a graph on the micro:bit, and displays the soil moisture value when pressing button A. If the moisture level drops below 50%, it turns on the motor for 0.5 seconds, and then turns it off.

Open the Bonsai Buckaroo Plant Care program in MakeCode

Your MakeCode screen should have the following MakeCode program loaded.

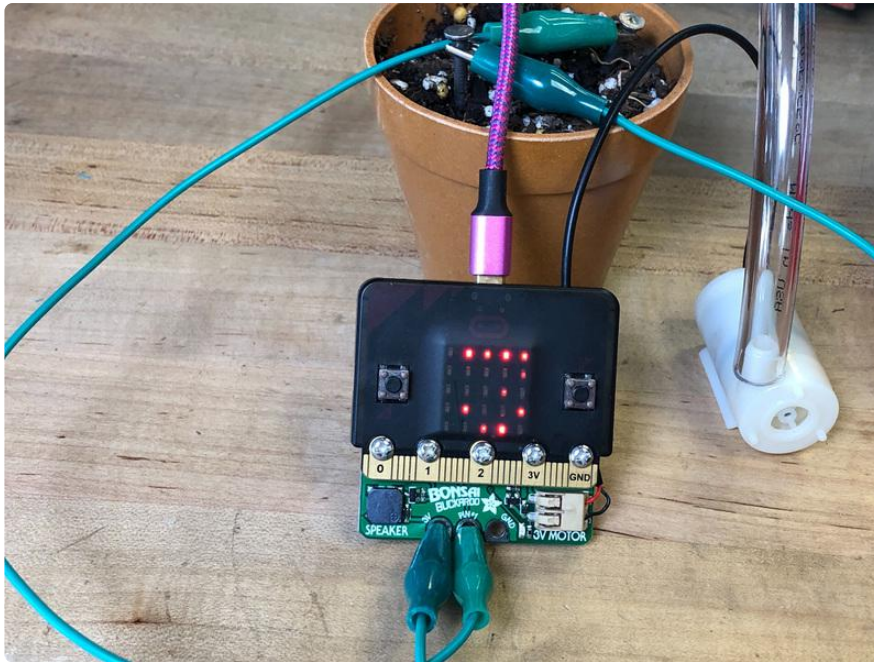
```
on start
  start melody entertainer repeating once

forever
  set reading to analog read pin P1
  plot bar graph of reading
  up to 1023
  if button A is pressed then
    show number reading
  if reading < 500 then
    play tone Middle C for 1 beat
    digital write pin P2 to 1
    pause (ms) 500
    digital write pin P2 to 0
  pause (ms) 1000
```

The program starts off with a little song. Then it begins checking the analog reading on Pin 1, and plots a bar graph on the LED array of the values. If you press button A, the LED array shows the actual reading.

If the reading drops below 500 (50%), it plays a tone, turns on the motor for 0.5 seconds, and turns it off. This repeats until the reading rises above 500.

That's all there is to plant care with Bonsai Buckaroo, micro:bit and MakeCode!

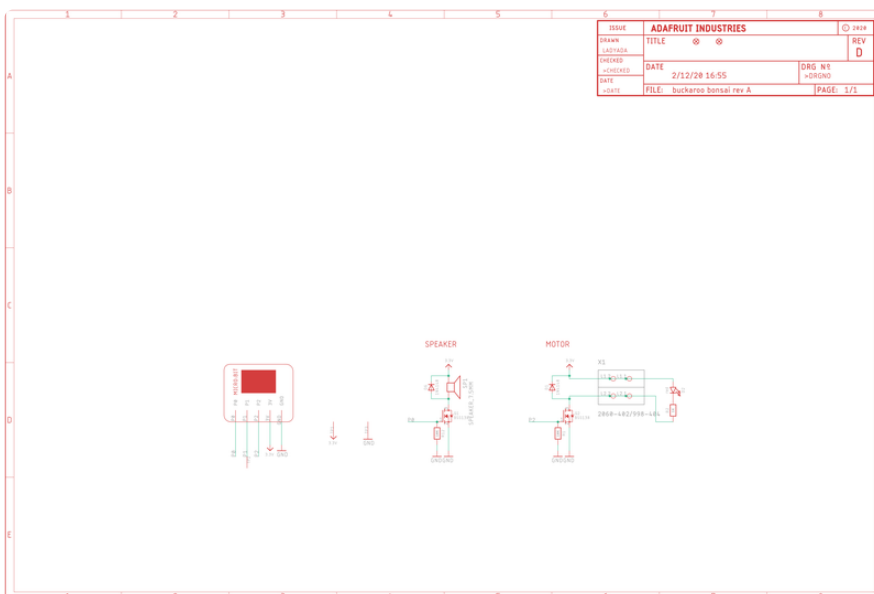


Downloads

Files

- [Fritzing object in Adafruit Fritzing Library \(\)](#)
- [EagleCAD files on GitHub \(\)](#)

Schematic



Fab Print

