Circuits fall 2020 Tutorial: analog measurements with the Feather board

INSTALL

0. Make sure you have completed installation of the Arduino IDE and added the Adafruit feather 32u4 board library: https://learn.adafruit.com/adafruit-feather-32u4-bluefruit-le/setup

HARDWARE SETUP

1. The remainder of this tutorial generally follows the Read Analog Voltage tutorial from Arduino, but we will make a few small adaptions since we are using an Adafruit feather, not an Arduino Uno: https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage
2. Plug your feather 32u4 into your breadboard.
3. Connect the 10k potentiometer configured as a voltage divider:
    a. Connect one of the 'outside' pins to the 3.3 V output pin "3V" (2nd down on left side as pictured below. This acts like the positive side of a 3.3V battery for our purposes today.
    b. Connect the other outside pin to ground, the 0V reference point. This is labeled "GND" (left side, 4th pin down).
    c. Connect the middle pin of the potentiometer to the analog input labeled "A0". Your board has a total of 6 analog inputs (like having 6 simultaneous mulit-meters!), but we only need one today.
    d.

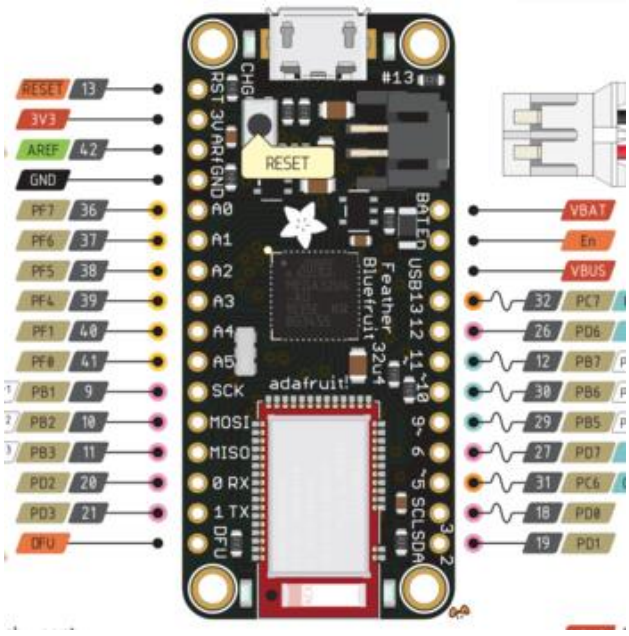

Figure 1. Adafruit feather 32u4 BLE pinout. Image credit: https://cdn-learn.adafruit.com/assets/assets/000/046/242/large1024/adafruit_products_Feather_32u4_Bluefruit_v2.3-1.png?1504885170

CODING/FIRMWARE…and a FIRST EXPERIMENT

4. Next we need to program the feather to make analog readings and tell us the answer. Turns out this is really easy to do, thanks to the wonderful open-source environment! Use it to your advantage!!!
    a. Look for the aptly titled "Code" section on the tutorial from Arduino: https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage
    b. Carefully note the description of what each line of code does in this section.
    c. Modify one line:

```
float voltage = sensorValue * (5.0/1023/0);
```

    Change the value 5.0 to 3.3V. Why do we need to do this?!
    Even better, cleaner would be to define in the code:
    const float Vin = 3.3;

    Then use the variable Vin in the equation for voltage above. You can see an example of better/cleaner code on the next page.

    d. Compile and upload the code. Just click the 'right arrow' button in the Arduino IDE. Easy ,no?
    e. Open the Serial Monitor so you can see the output voltage being measured. Select from the menu at top ribbon: "Tools > Serial Monitor"
    f. Twist the potentiometer back and forth. What do you observe in terms of the voltage readings?
    g. Finally, turn the pot all the way in one direction. Now rotate the knob to about 5 equally spaced positions. Then fill out the following table as you go. Compare and contrast your predicted voltage output to the measured output. Do they trend as expected? Are they in reasonable agreement? What would account for discrepancies noted?

R1 (Ω)	R2 (Ω)	Vout (V), predicted	Vout (V), measured	I (mA)

SAMPLE ARDUINO CODE

```
/*
  ReadAnalogVoltage

  Reads an analog input on pin 0, converts it to voltage, and prints
the result to the Serial Monitor.
  Graphical representation is available using Serial Plotter (Tools >
Serial Plotter menu).
  Attach the center pin of a potentiometer to pin A0, and the outside
pins to +5V and ground.

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/ReadAnalogVoltage
*/

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int AIN0 = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 1023) to a
voltage (0 - 3.3V):
  float Vout0 = AIN0 * (3.3 / 1023.0);
  // print out the value you read:
  Serial.println(Vout0);
}
```