

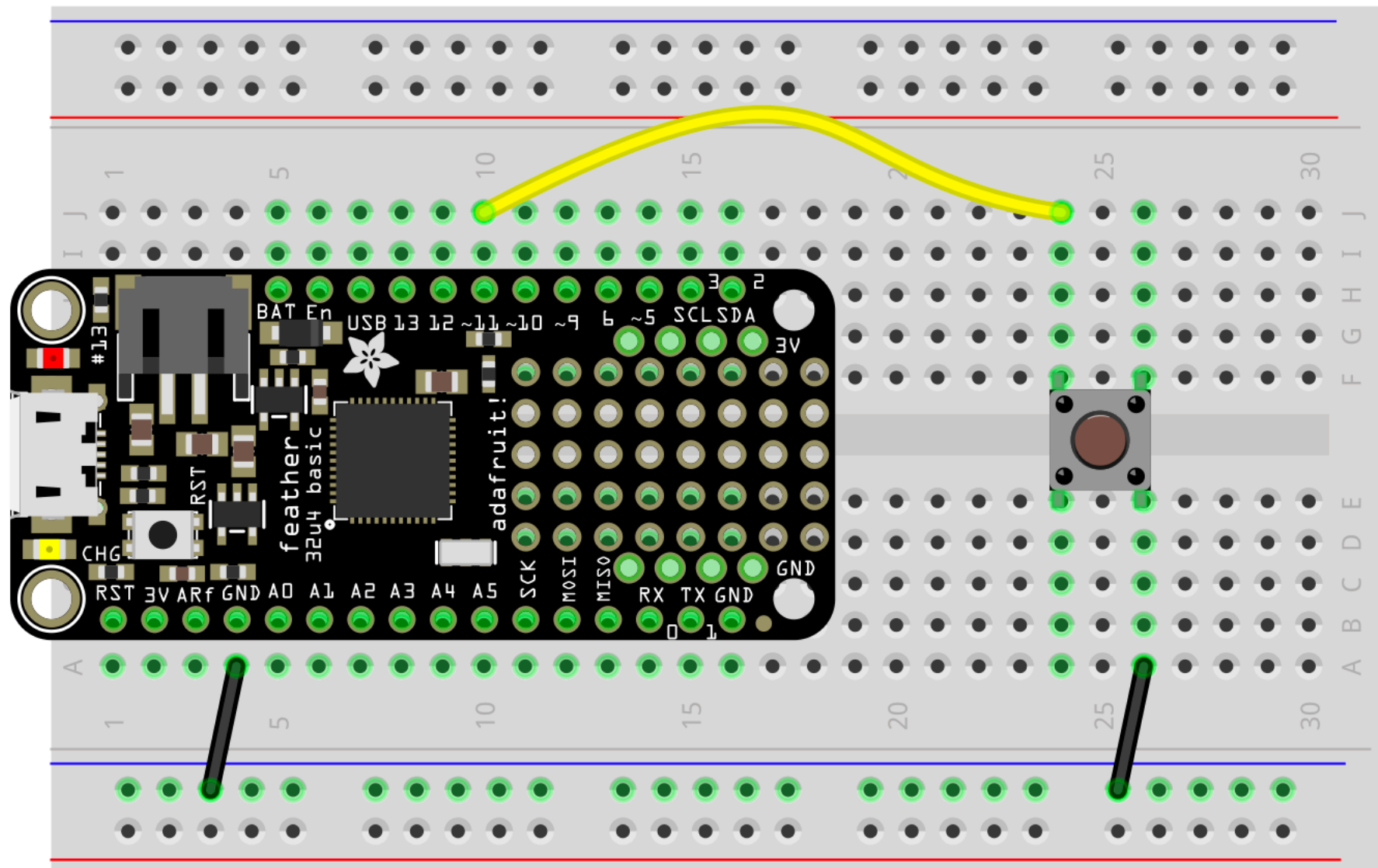
ARDUINO

+

PROCESSING

ARDUINO HID -> COMPUTER

# BUTTON INPUT



# TRANSLATE BUTTON PRESS INTO KEYBOARD PRESS

```
keyboard | Arduino 1.8.16

keyboard
1 #include "Keyboard.h"
2
3 const int BUTTON_PIN = 11;
4
5 void setup() {
6   pinMode(BUTTON_PIN, INPUT_PULLUP);
7
8   // initialize control as a keyboard
9   Keyboard.begin();
10 }
11
12 void loop() {
13   // read the pushbutton:
14   int val = digitalRead(BUTTON_PIN);
15
16   if (val == LOW) {
17     Keyboard.print("x");
18     delay(100);
19   }
20 }

Done uploading.
done in 0.019 seconds
CPU reset.

17 Adafruit Feather M0 Express, Small (-Os) (standard), Arduino, Off on /dev/cu.usbmodem14201
```

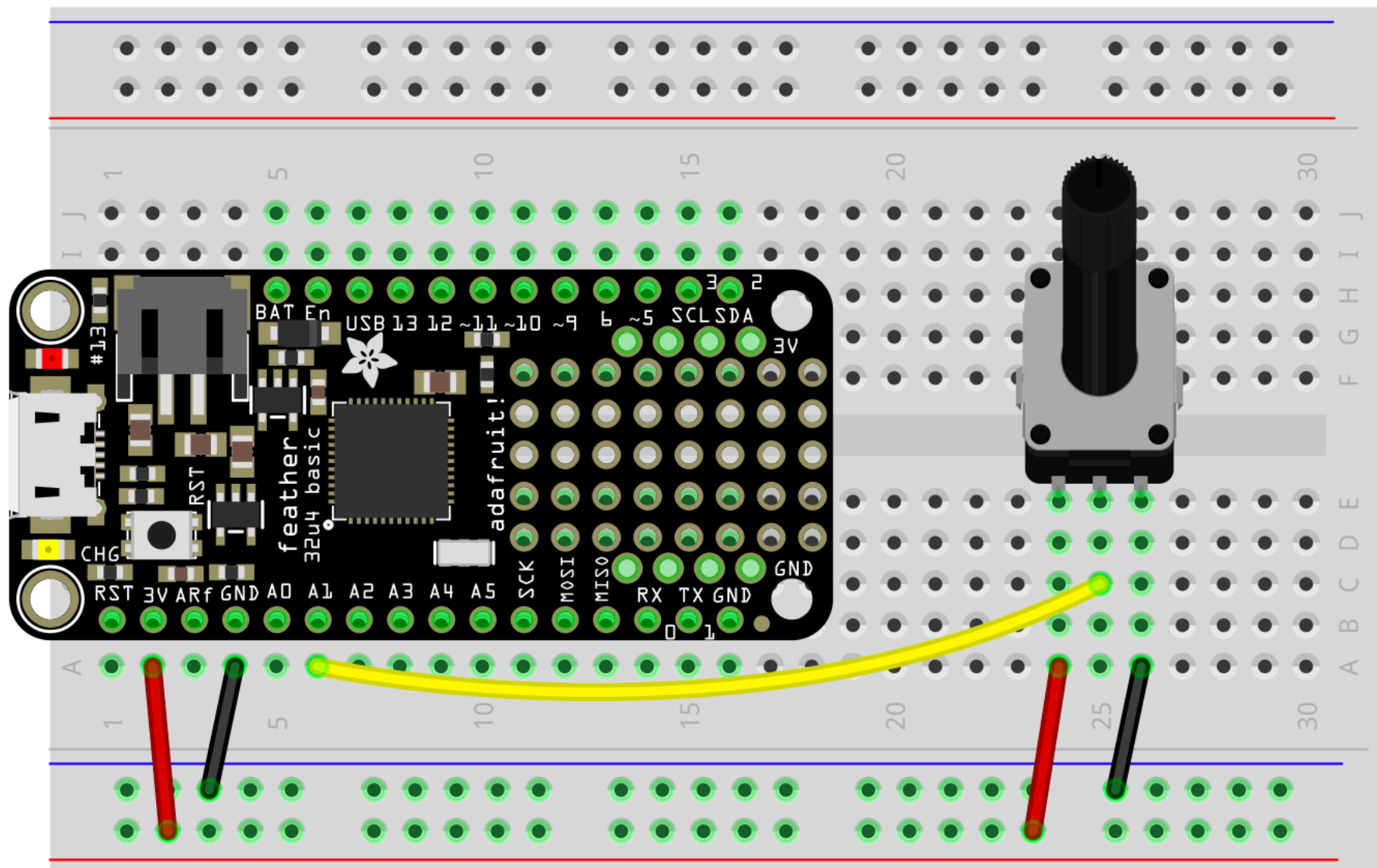
The screenshot shows a code editor window titled "sketch.js — keyboard". The Explorer panel on the left shows a project named "KEYBOARD" with files: "libraries", "index.html", "jsconfig.json", "sketch.js", and "style.css". The "sketch.js" file is selected. The main editor area shows the following JavaScript code:

```
1  let bgColor = 0;
2
3  function setup() {
4    createCanvas(windowWidth, windowHeight);
5  }
6
7  function draw() {
8    background(bgColor);
9  }
10
11 function keyPressed() {
12   bgColor = color(random(0, 255), random(0, 255), random(0, 255));
13 }
14
```

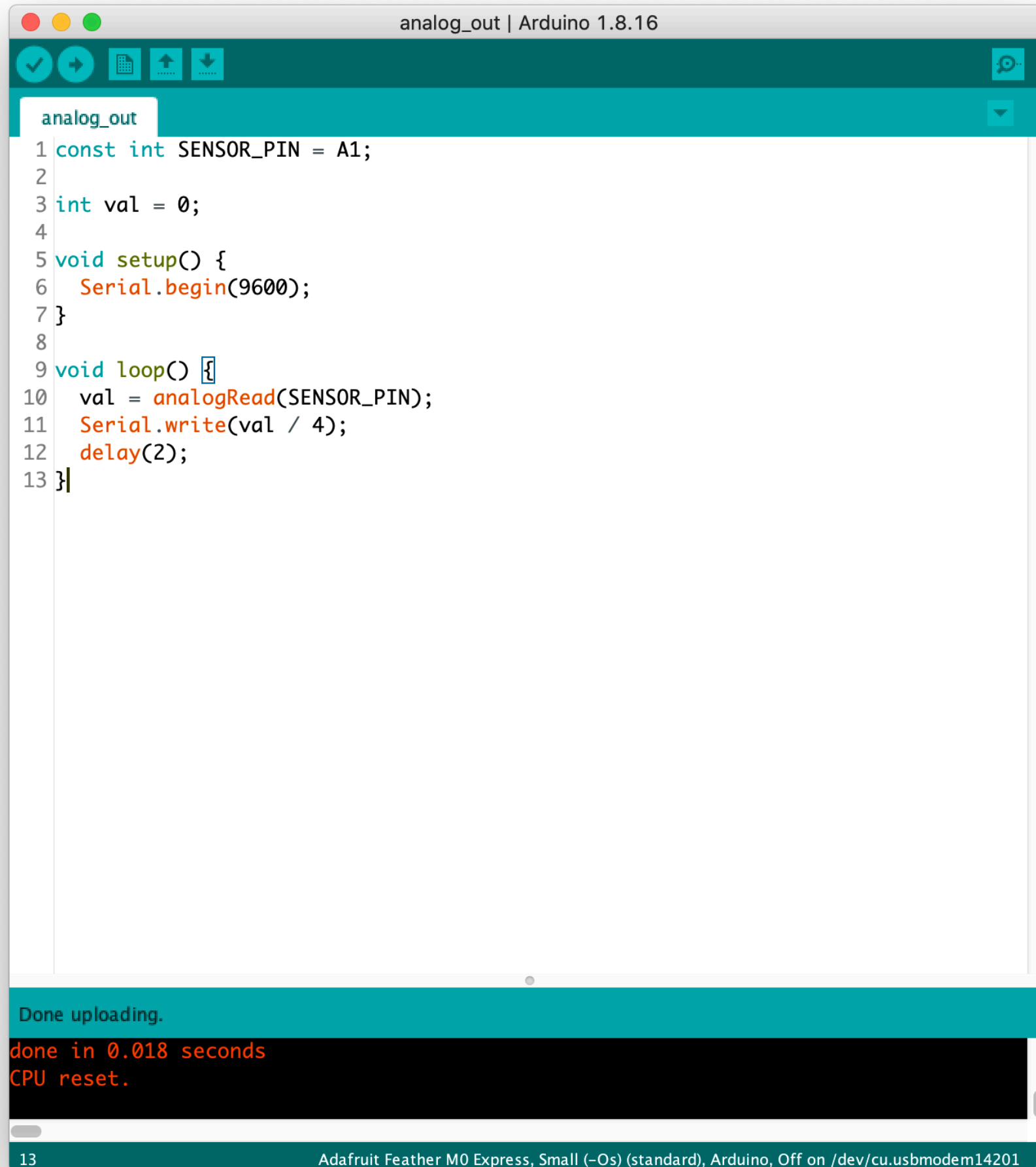
The bottom status bar shows the following information: "main\*", "0 ↓ 1 ↑", "0 0", "Spaces: 2", "UTF-8", "LF", "{}", "JavaScript", "Indents: 0", "Port : 5500", "✓ Spell", "✓ Prettier", and a user icon.

# RESPOND TO KEYBOARD PRESSES

# POTENTIOMETER (OR OTHER ANALOG SENSOR) INPUT



# WRITE VALUES VIA SERIAL WITH ARDUINO

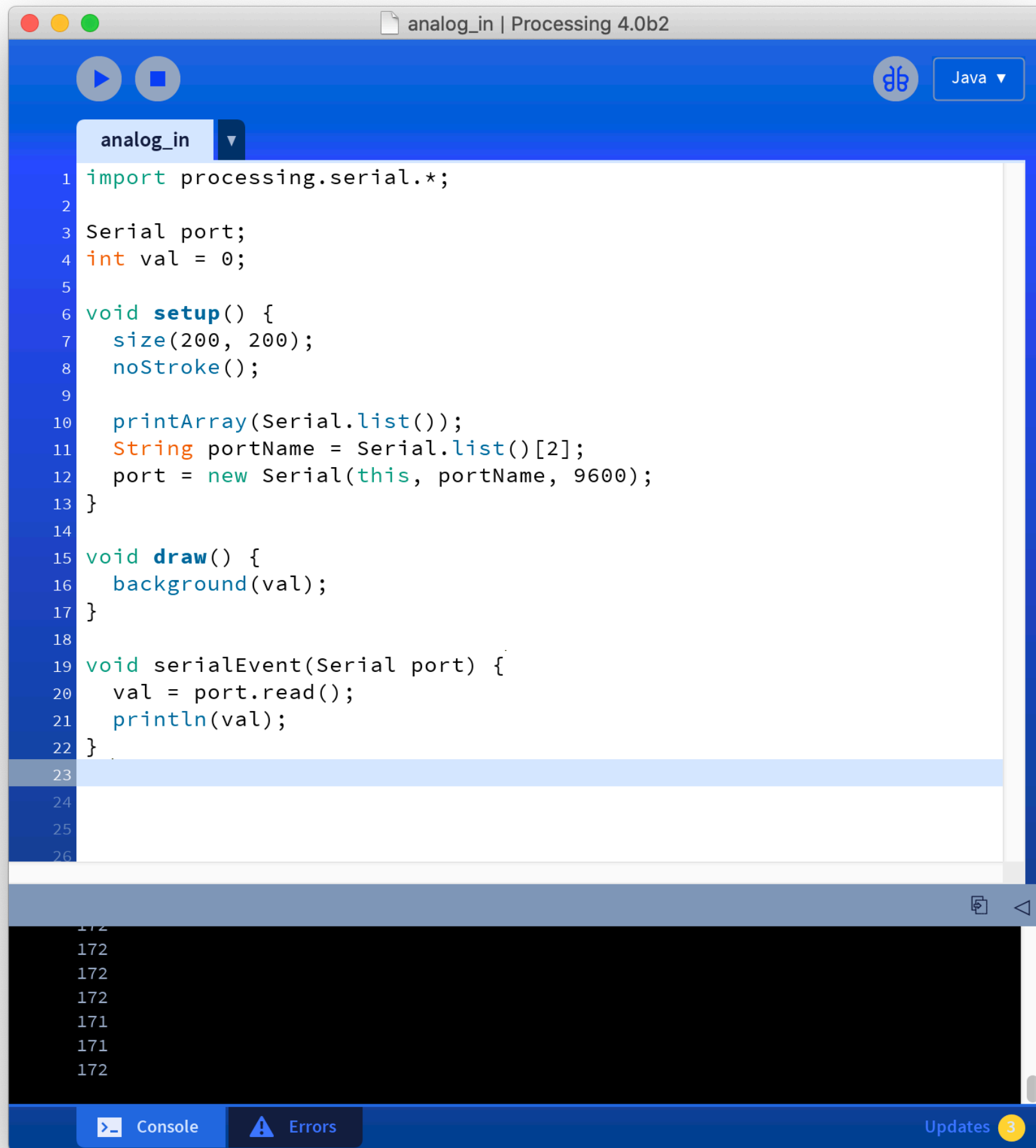


The screenshot shows the Arduino IDE interface. The title bar indicates the sketch is named 'analog\_out' and the IDE version is '1.8.16'. The code editor contains the following C++ code:

```
1 const int SENSOR_PIN = A1;
2
3 int val = 0;
4
5 void setup() {
6   Serial.begin(9600);
7 }
8
9 void loop() {
10  val = analogRead(SENSOR_PIN);
11  Serial.write(val / 4);
12  delay(2);
13 }
```

Below the code editor, a status bar shows the upload progress. A teal bar indicates 'Done uploading.' followed by a black bar showing 'done in 0.018 seconds' and 'CPU reset.' in orange text. At the bottom, the board and port are specified: '13 Adafruit Feather M0 Express, Small (-Os) (standard), Arduino, Off on /dev/cu.usbmodem14201'.





The screenshot shows the Processing IDE with a file named 'analog\_in'. The code is as follows:

```
1 import processing.serial.*;
2
3 Serial port;
4 int val = 0;
5
6 void setup() {
7   size(200, 200);
8   noStroke();
9
10  printArray(Serial.list());
11  String portName = Serial.list()[2];
12  port = new Serial(this, portName, 9600);
13 }
14
15 void draw() {
16   background(val);
17 }
18
19 void serialEvent(Serial port) {
20   val = port.read();
21   println(val);
22 }
23
24
25
26
```

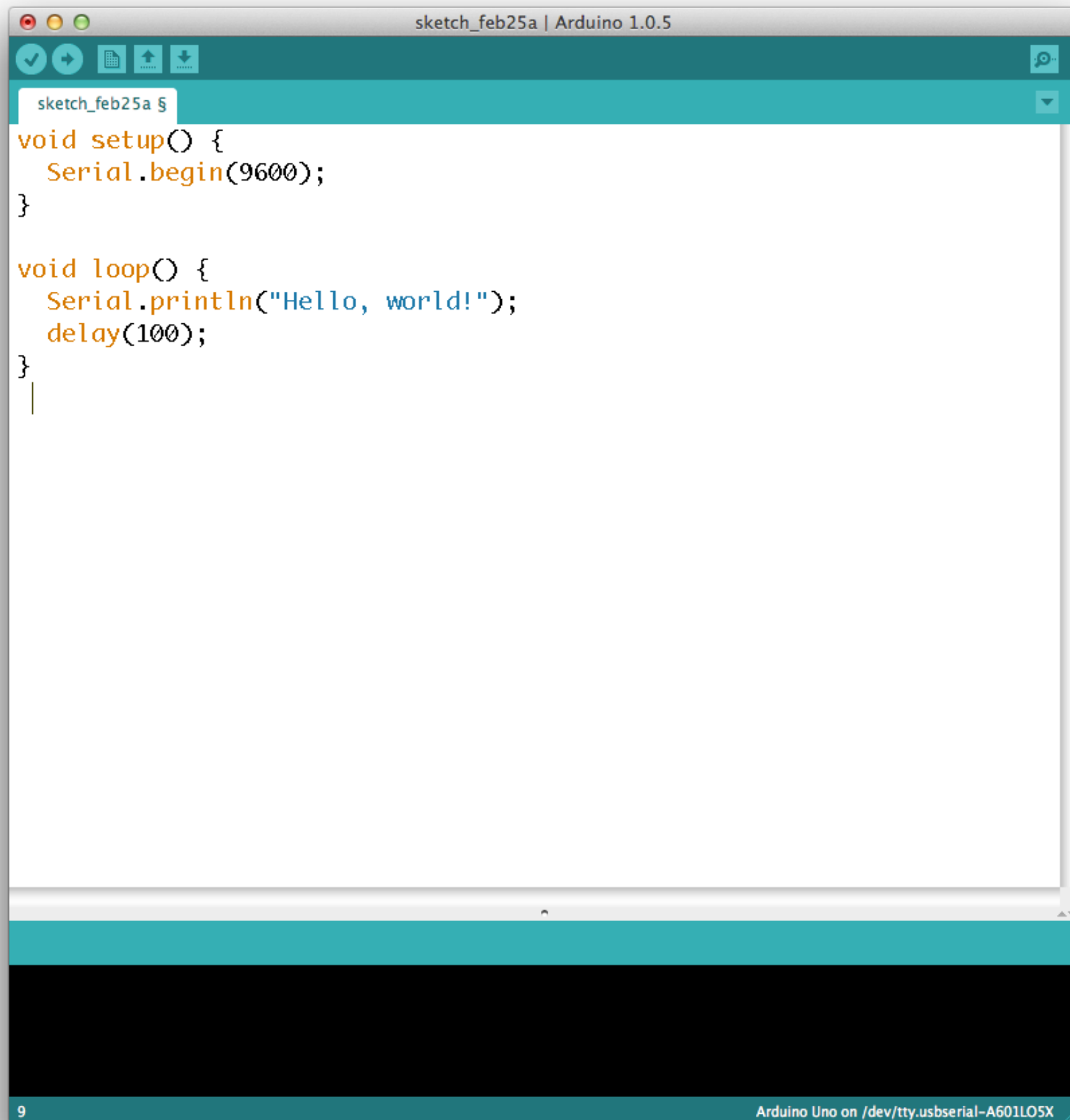
The console output shows the following values:

```
172
172
172
171
171
172
```

# READ VALUES VIA SERIAL WITH PROCESSING



ARDUINO SERIAL -> PROCESSING



The image shows a screenshot of the Arduino IDE interface. The title bar at the top reads "sketch\_feb25a | Arduino 1.0.5". The main text area contains the following C++ code:

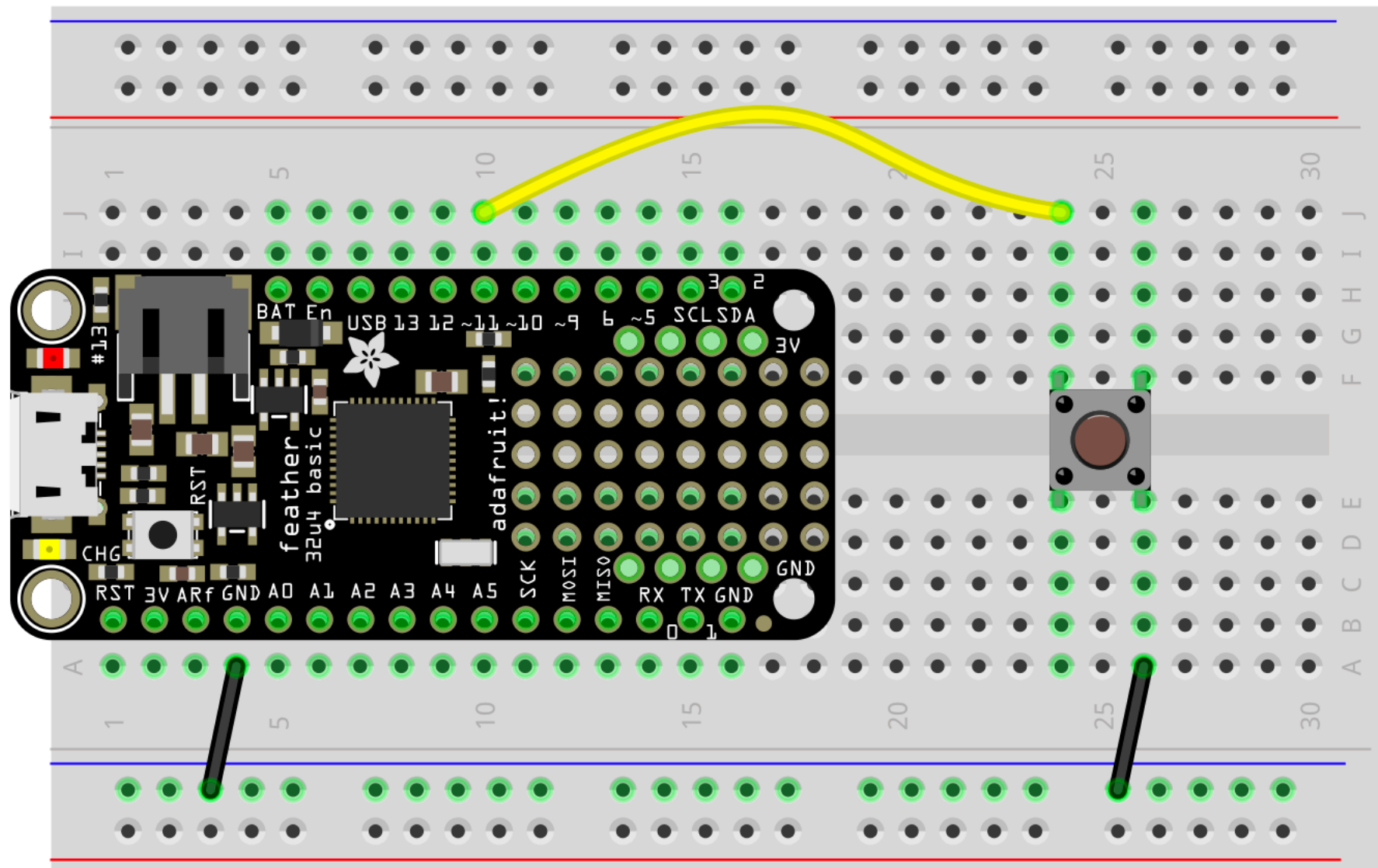
```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println("Hello, world!");  
  delay(100);  
}
```

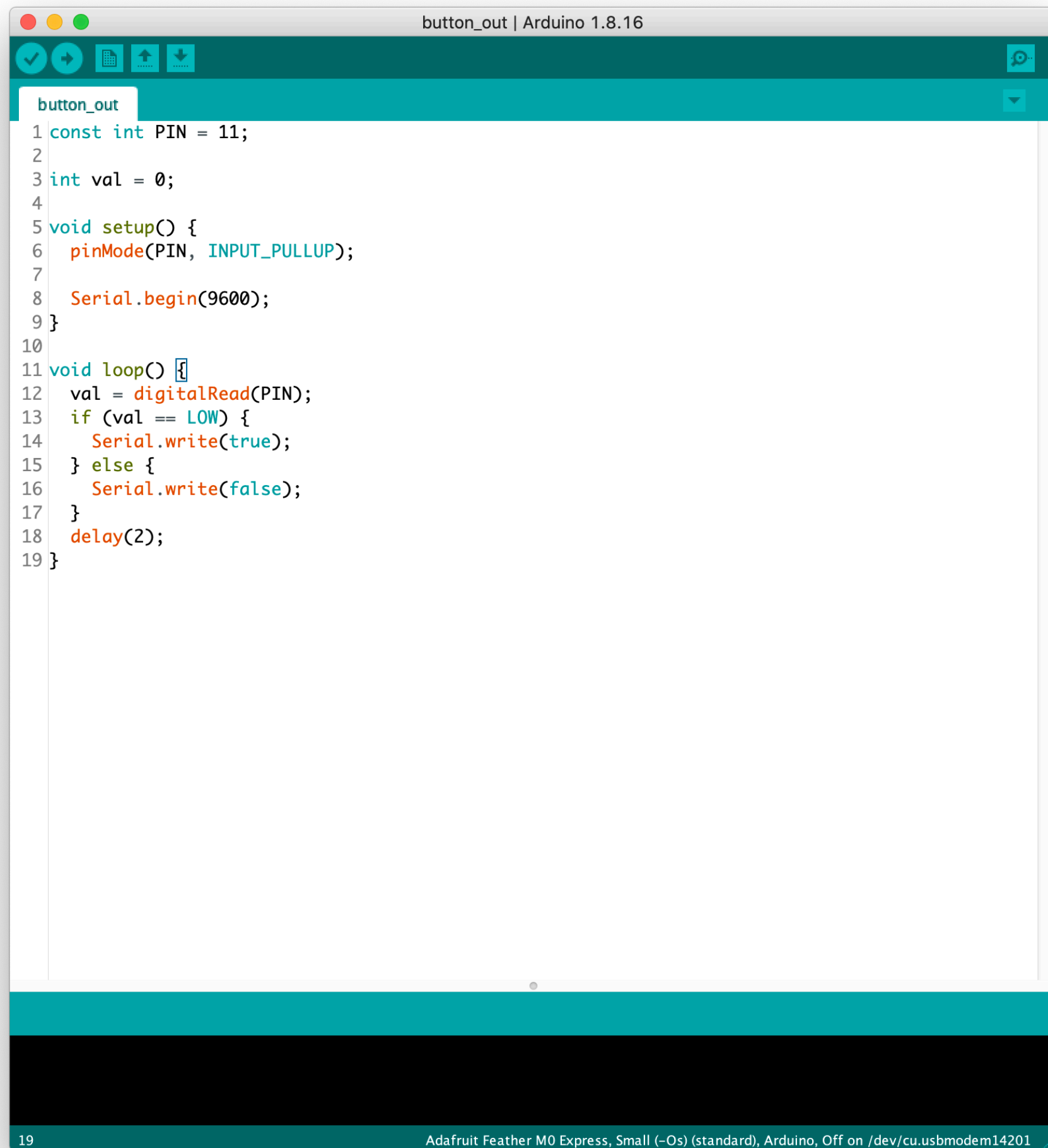
The code is written in a syntax-highlighted font. The IDE has a teal header bar with icons for checking, running, saving, and uploading. Below the code editor is a black serial monitor area, and at the very bottom is a status bar.

9 Arduino Uno on /dev/tty.usbserial-A601LO5X

# PRINT MESSAGE VIA SERIAL

# BUTTON INPUT



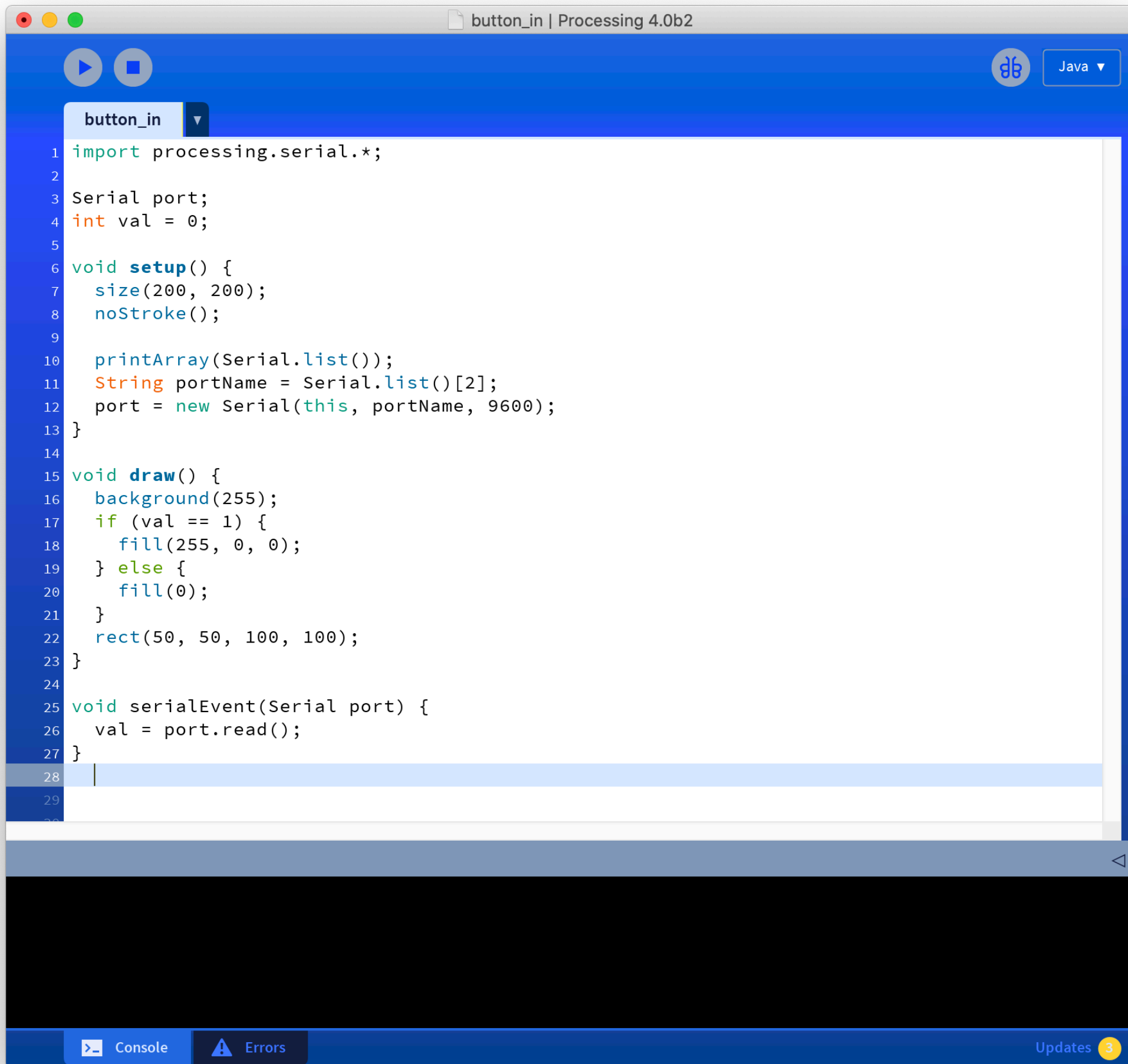


```
button_out
1 const int PIN = 11;
2
3 int val = 0;
4
5 void setup() {
6   pinMode(PIN, INPUT_PULLUP);
7
8   Serial.begin(9600);
9 }
10
11 void loop() {
12   val = digitalRead(PIN);
13   if (val == LOW) {
14     Serial.write(true);
15   } else {
16     Serial.write(false);
17   }
18   delay(2);
19 }
```

19

Adafruit Feather M0 Express, Small (-Os) (standard), Arduino, Off on /dev/cu.usbmodem14201

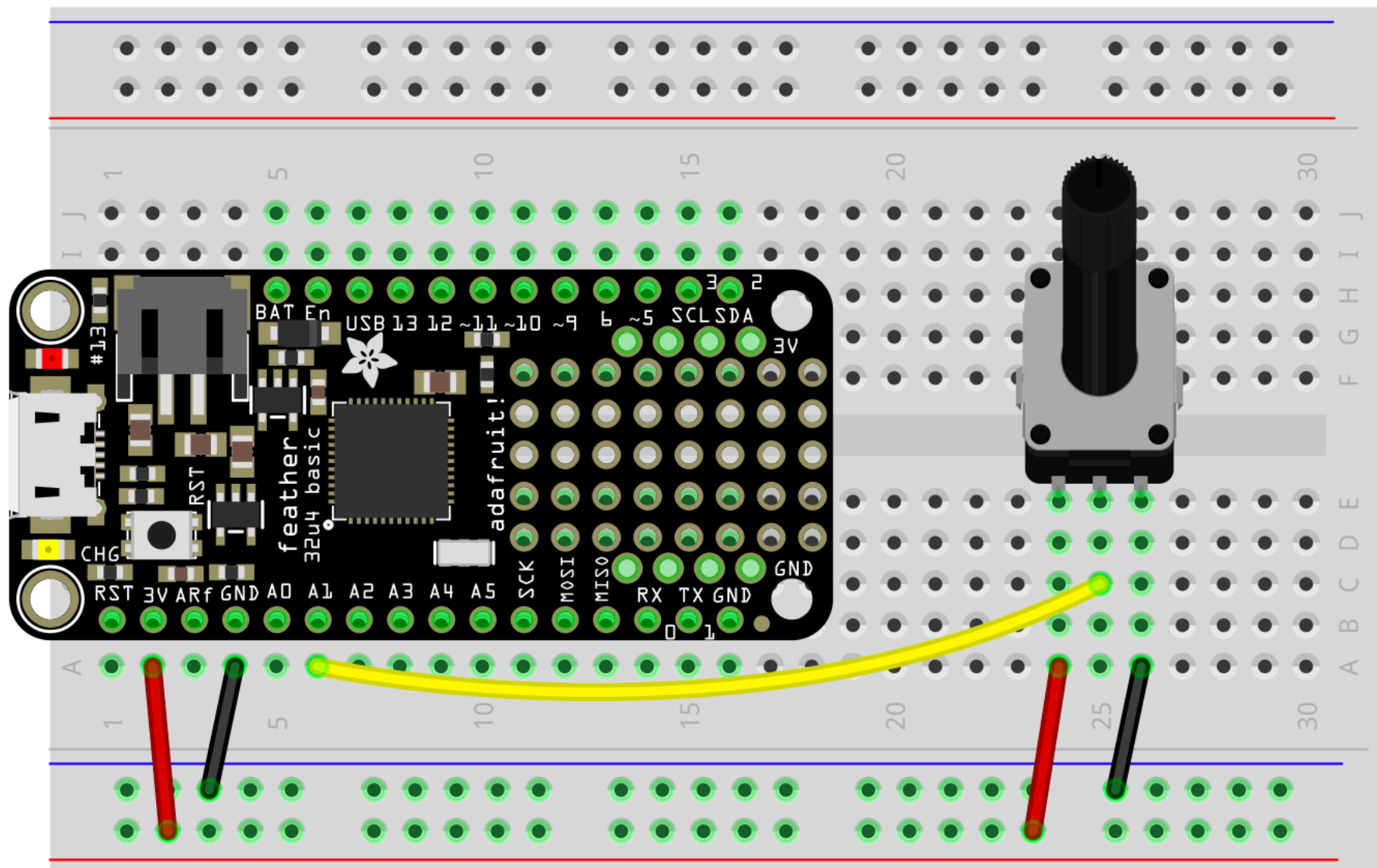
# WRITE VALUES VIA SERIAL WITH ARDUINO



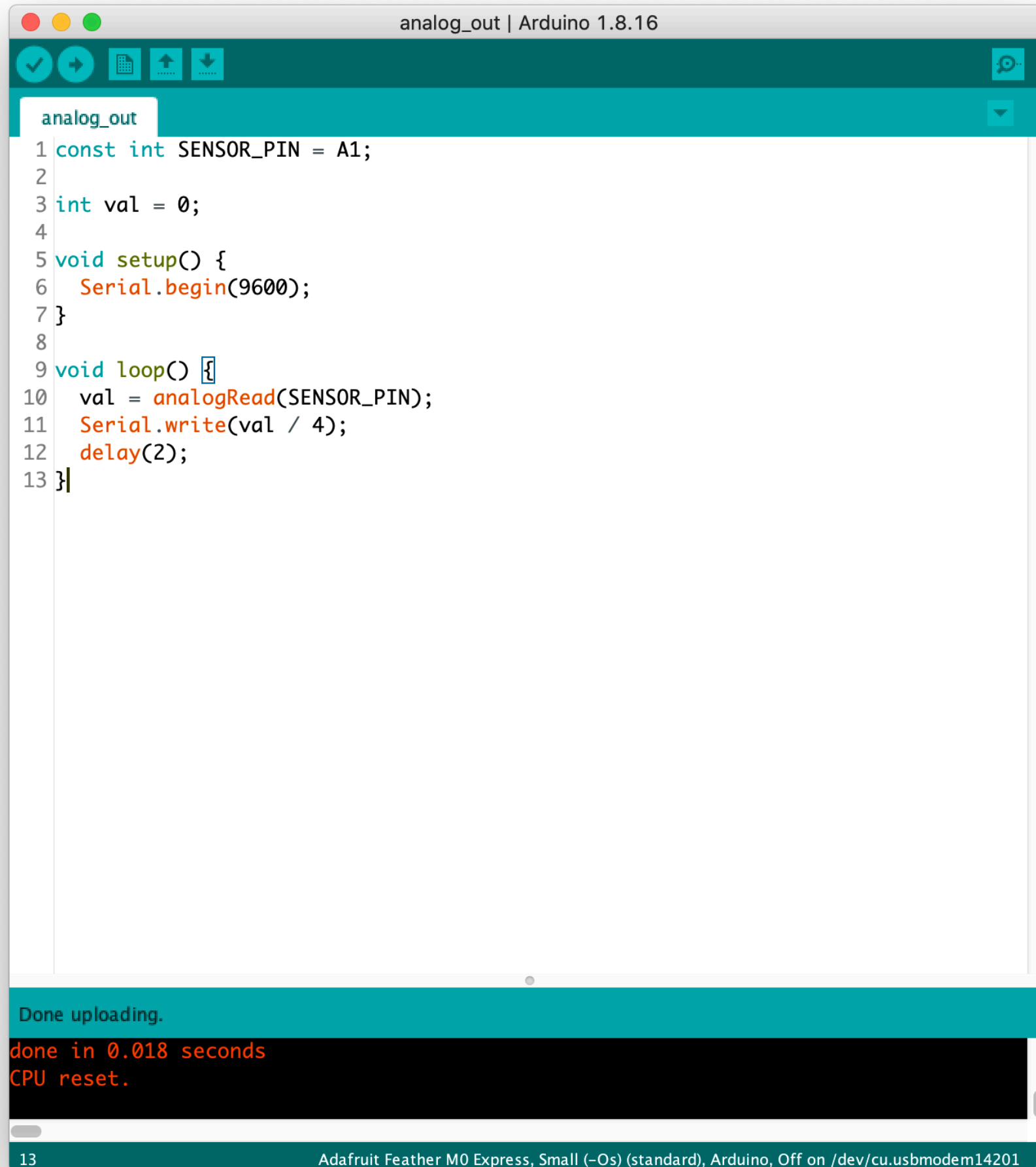
```
1 import processing.serial.*;
2
3 Serial port;
4 int val = 0;
5
6 void setup() {
7   size(200, 200);
8   noStroke();
9
10  printArray(Serial.list());
11  String portName = Serial.list()[2];
12  port = new Serial(this, portName, 9600);
13 }
14
15 void draw() {
16   background(255);
17   if (val == 1) {
18     fill(255, 0, 0);
19   } else {
20     fill(0);
21   }
22   rect(50, 50, 100, 100);
23 }
24
25 void serialEvent(Serial port) {
26   val = port.read();
27 }
28
29
30
```

# READ VALUES VIA SERIAL WITH PROCESSING

# POTENTIOMETER (OR OTHER ANALOG SENSOR) INPUT



# WRITE VALUES VIA SERIAL WITH ARDUINO



The screenshot shows the Arduino IDE interface. The title bar reads 'analog\_out | Arduino 1.8.16'. The main editor area contains the following C++ code:

```
1 const int SENSOR_PIN = A1;
2
3 int val = 0;
4
5 void setup() {
6   Serial.begin(9600);
7 }
8
9 void loop() {
10  val = analogRead(SENSOR_PIN);
11  Serial.write(val / 4);
12  delay(2);
13 }
```

Below the editor, a status bar indicates 'Done uploading.' and 'done in 0.018 seconds CPU reset.' The bottom status bar shows the board 'Adafruit Feather M0 Express, Small (-Os) (standard)' and the port '/dev/cu.usbmodem14201'.



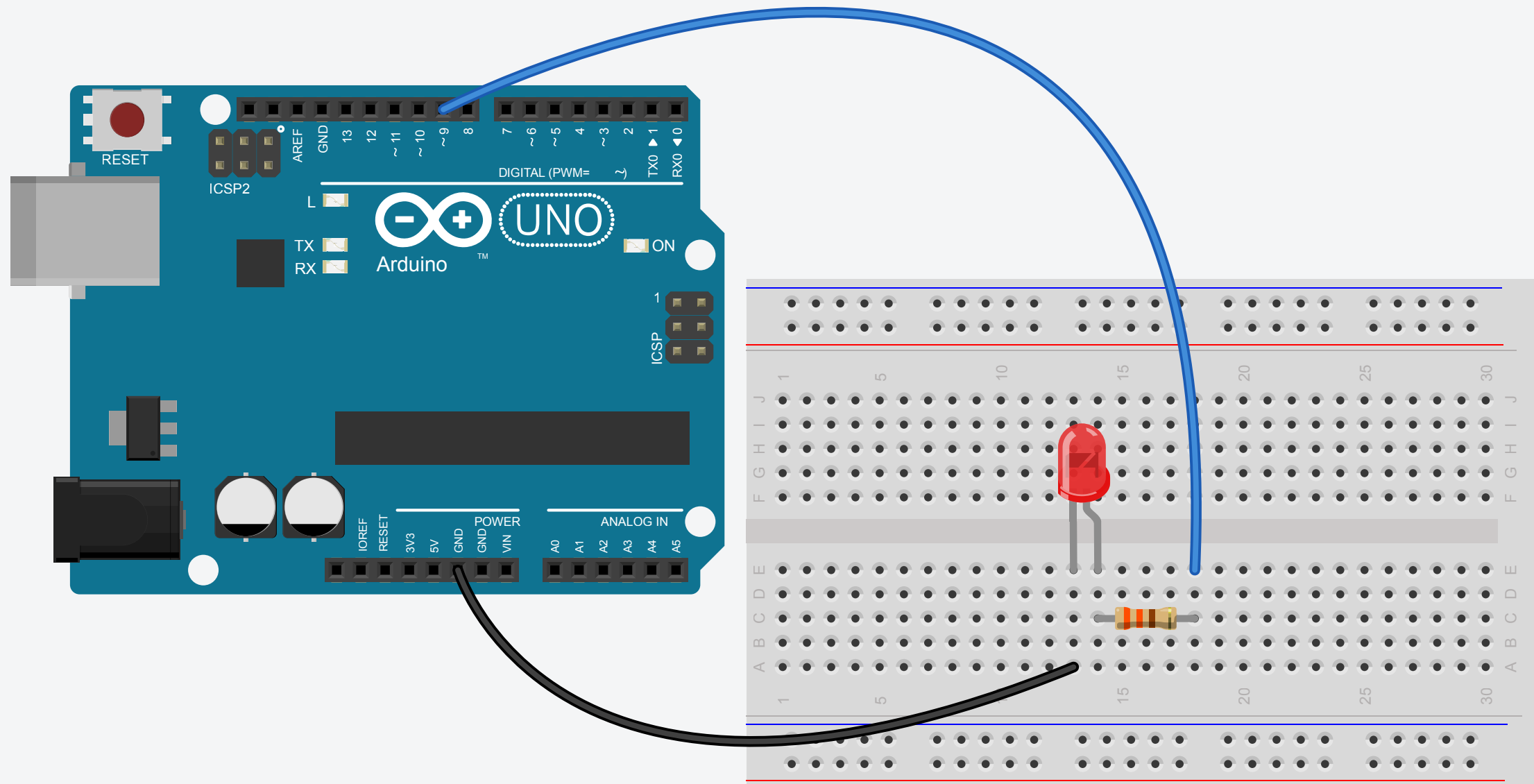
```
analog_in
1 import processing.serial.*;
2
3 Serial port;
4 int val = 0;
5
6 void setup() {
7   size(200, 200);
8   noStroke();
9
10  printArray(Serial.list());
11  String portName = Serial.list()[2];
12  port = new Serial(this, portName, 9600);
13 }
14
15 void draw() {
16   background(val);
17 }
18
19 void serialEvent(Serial port) {
20   val = port.read();
21   println(val);
22 }
23
24
25
26
```

172  
172  
172  
171  
171  
172

# READ VALUES VIA SERIAL WITH PROCESSING

PROCESSING -> ARDUINO

# PWM LED OUTPUT



processing\_serial\_out | Processing 3.3.6

processing\_serial\_out

```
1 import processing.serial.*;
2
3 Serial port;
4
5
6 void setup() {
7   size(256, 200);
8   printArray(Serial.list());
9   String portName = Serial.list()[2];
10  port = new Serial(this, portName, 9600);
11 }
12
13 void draw() {
14   for (int i=0; i<256; i++) {
15     stroke(i);
16     line(i, 0, i, height);
17   }
18   port.write(mouseX);
19 }
20
21
22
23
24
25
26
27
28
29
```

Done saving.

```
[1] "/dev/cu.SLAB_USBtoUART"
[2] "/dev/cu.SLAB_USBtoUART"
[3] "/dev/cu.SoundCoremini-SerialPor"
[4] "/dev/tty.Bluetooth-Incoming-Port"
[5] "/dev/tty.JBLXtreme-SPPDev"
[6] "/dev/tty.SLAB_USBtoUART"
[7] "/dev/tty.SoundCoremini-SerialPor"
```

Console Errors Updates 2

# WRITE VALUES VIA SERIAL WITH PROCESSING



The screenshot shows the Arduino IDE interface. The title bar indicates the sketch is named 'analog\_in' and the IDE version is 1.8.5. The code editor contains the following C++ code:

```
1 const int LED = 11;
2
3 void setup() {
4   Serial.begin(9600);
5   pinMode(LED, OUTPUT);
6 }
7
8 void loop() {
9   byte input;
10
11   if (Serial.available()) {
12     input = Serial.read();
13     analogWrite(LED, input);
14   }
15 }
```

Below the code editor, a status bar shows 'Done uploading.' and memory usage information: 'Sketch uses 1902 bytes (5%) of program storage space. Maximum is 32256 bytes. Global variables use 184 bytes (8%) of dynamic memory, leaving 1864 bytes for...'. At the bottom, the status bar indicates the board is 'Arduino/Genuino Uno' and the port is '/dev/cu.SLAB\_USBtoUART'.

# READ VALUES VIA SERIAL WITH ARDUINO