

DIGITAL INPUTS (BUTTONS AND SWITCHES)

**THE INPUTS ON THE
ARDUINO READ VOLTAGE.**

**ALL INPUTS NEED TO BE
THOUGHT OF IN TERMS
OF VOLTAGE
DIFFERENTIALS.**

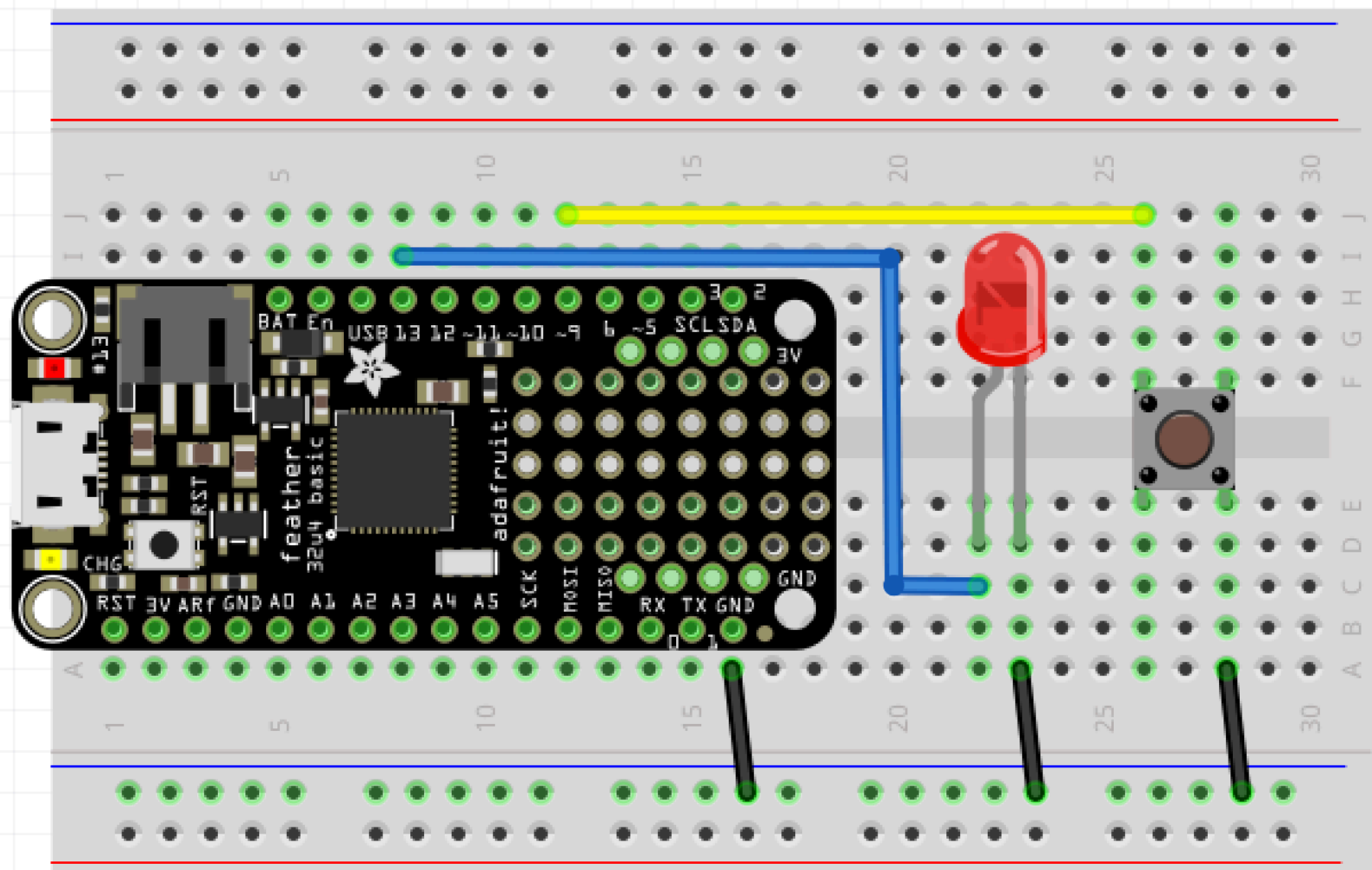
**THE ANALOG INPUTS
CONVERT VOLTAGE
LEVELS TO A NUMERICAL
VALUE.**

The Arduino always needs to have a good reference for its input values. In the case of a button (digital input) it should always be either HIGH or LOW. We can do this by setting the pinMode to INPUT_PULLUP.

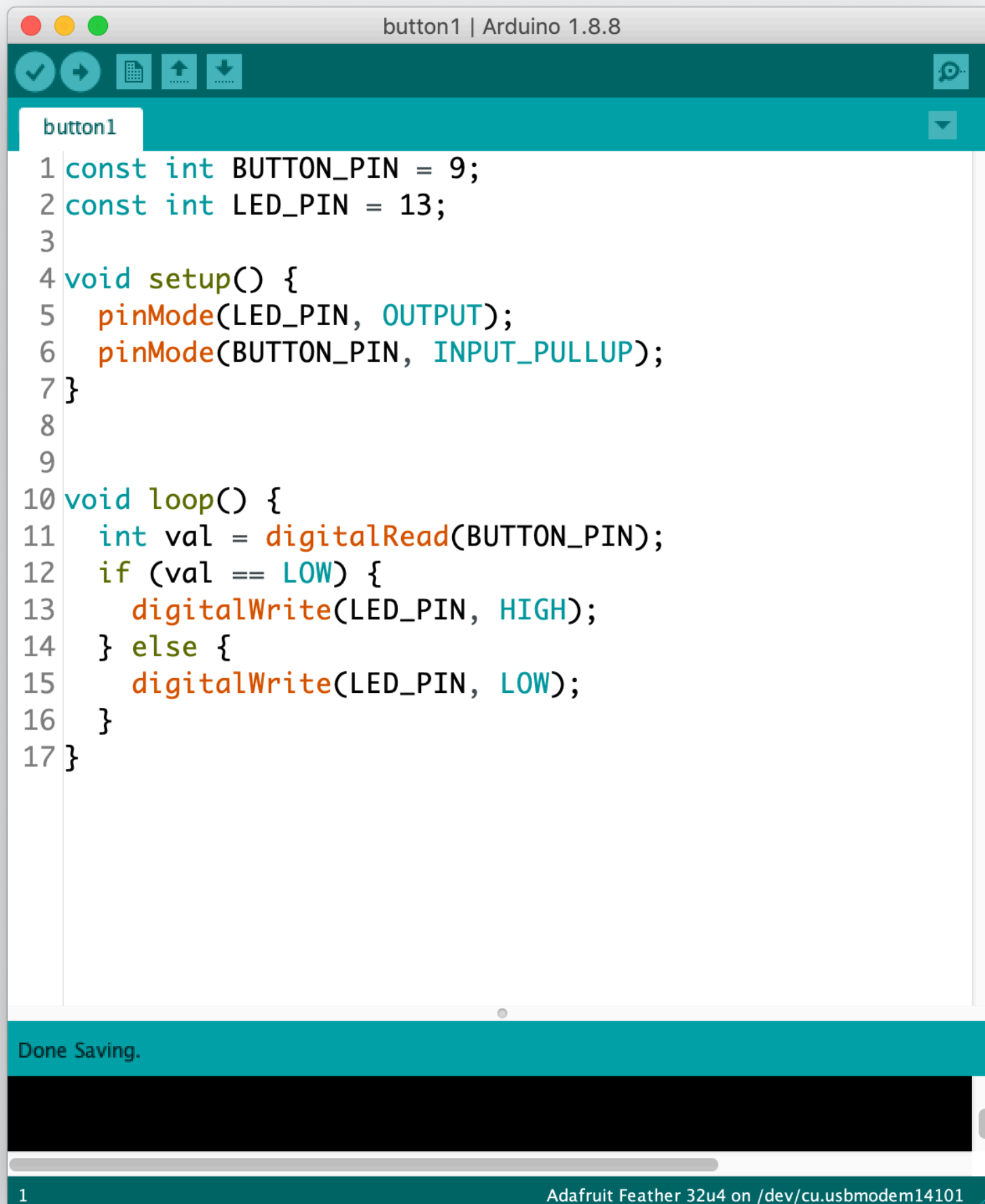
CONNECTING A BUTTON

When the button isn't pressed, the voltage going to pin is "pulled up" to HIGH.

When the button is pressed, the connection to ground is completed so the pin is LOW.



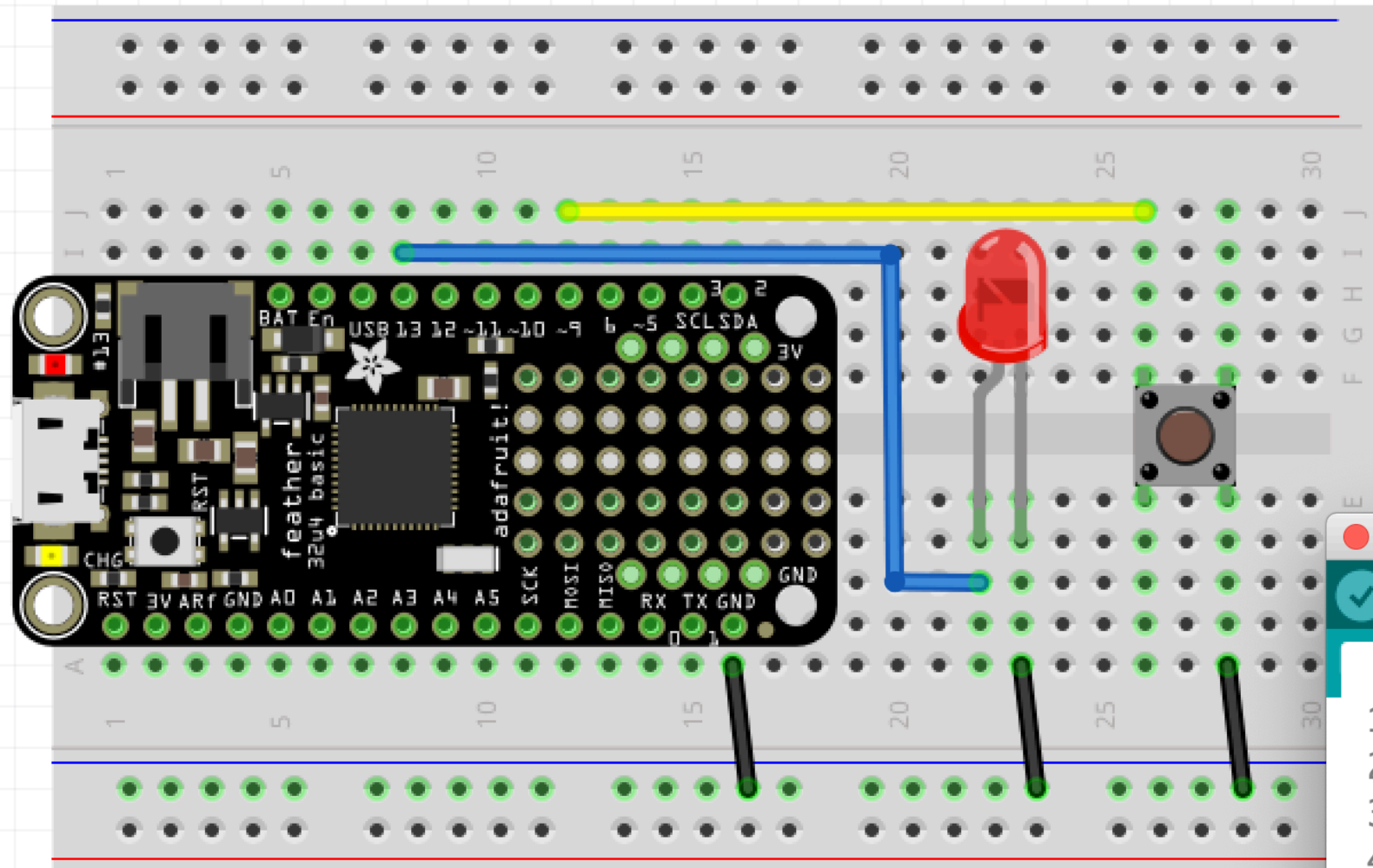
CONTROL AN LED WITH A BUTTON



The image shows a screenshot of the Arduino IDE interface. The title bar at the top reads "button1 | Arduino 1.8.8". The main editor area contains the following C++ code:

```
1 const int BUTTON_PIN = 9;
2 const int LED_PIN = 13;
3
4 void setup() {
5   pinMode(LED_PIN, OUTPUT);
6   pinMode(BUTTON_PIN, INPUT_PULLUP);
7 }
8
9
10 void loop() {
11   int val = digitalRead(BUTTON_PIN);
12   if (val == LOW) {
13     digitalWrite(LED_PIN, HIGH);
14   } else {
15     digitalWrite(LED_PIN, LOW);
16   }
17 }
```

Below the code editor, a status bar shows "Done Saving." and a black console area. At the very bottom, the footer indicates the board and port: "1 Adafruit Feather 32u4 on /dev/cu.usbmodem14101".



CONNECTING A BUTTON

When the button isn't pressed, the voltage going to pin is "pulled up" to HIGH.

When the button is pressed, the connection to ground is completed so the pin is LOW.

```
button1 | Arduino 1.8.8
button1
1 const int BUTTON_PIN = 9;
2 const int LED_PIN = 13;
3
4 void setup() {
5   pinMode(LED_PIN, OUTPUT);
6   pinMode(BUTTON_PIN, INPUT_PULLUP);
7 }
8
9
10 void loop() {
11   int val = digitalRead(BUTTON_PIN);
12   if (val == LOW) {
13     digitalWrite(LED_PIN, HIGH);
14   } else {
15     digitalWrite(LED_PIN, LOW);
16   }
17 }
```

**ANYTHING CAN BECOME A
SWITCH**