STATE

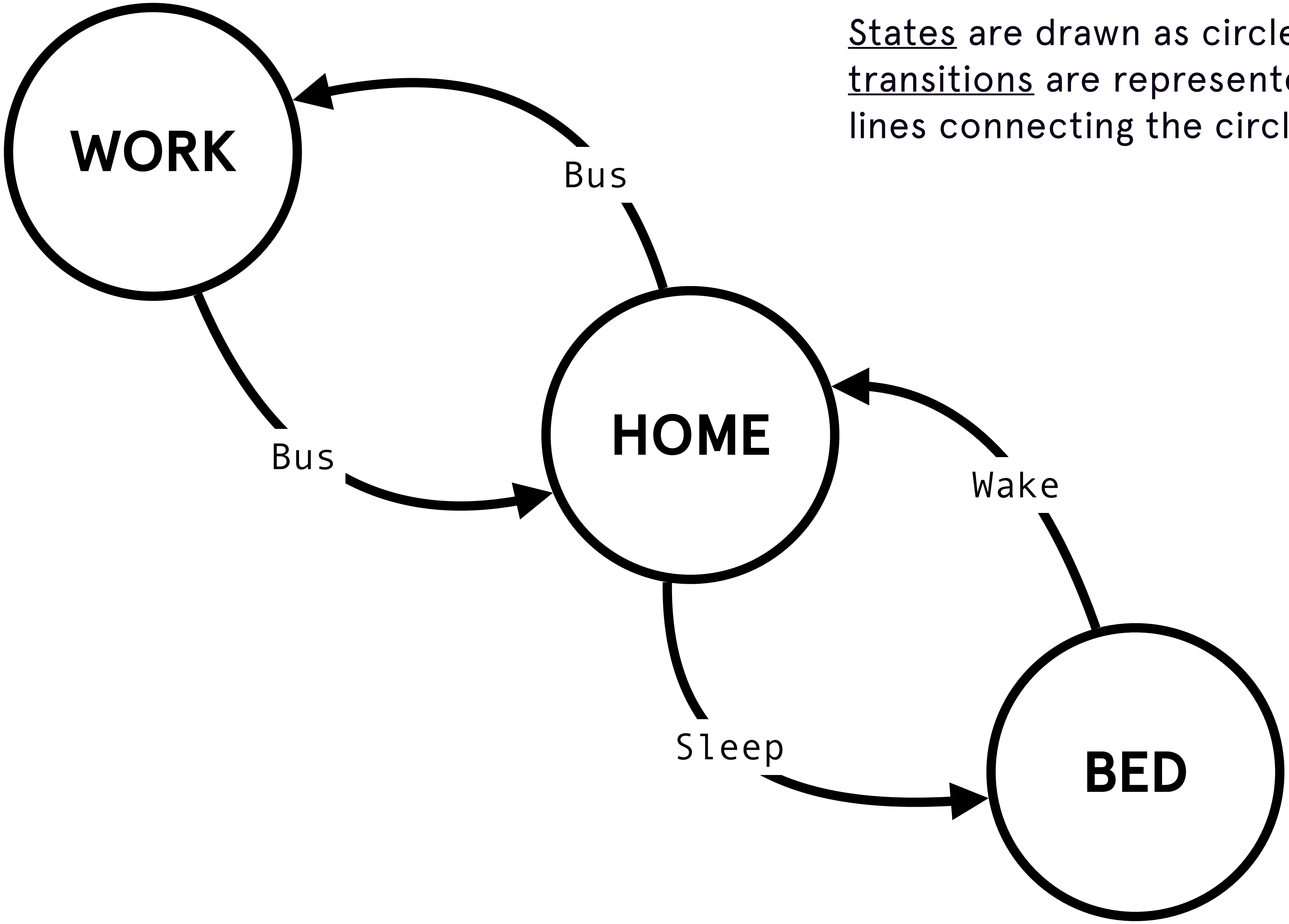
THE STATE MACHINE IS A MODEL OF COMPUTATION WHERE A SYSTEM OR PROGRAM IS IN ONE OF A FINITE NUMBER OF STATES AT ANY GIVEN TIME.

THE FUNDAMENTAL COMPONENTS OF A STATE MACHINE ARE STATES AND TRANSITIONS.

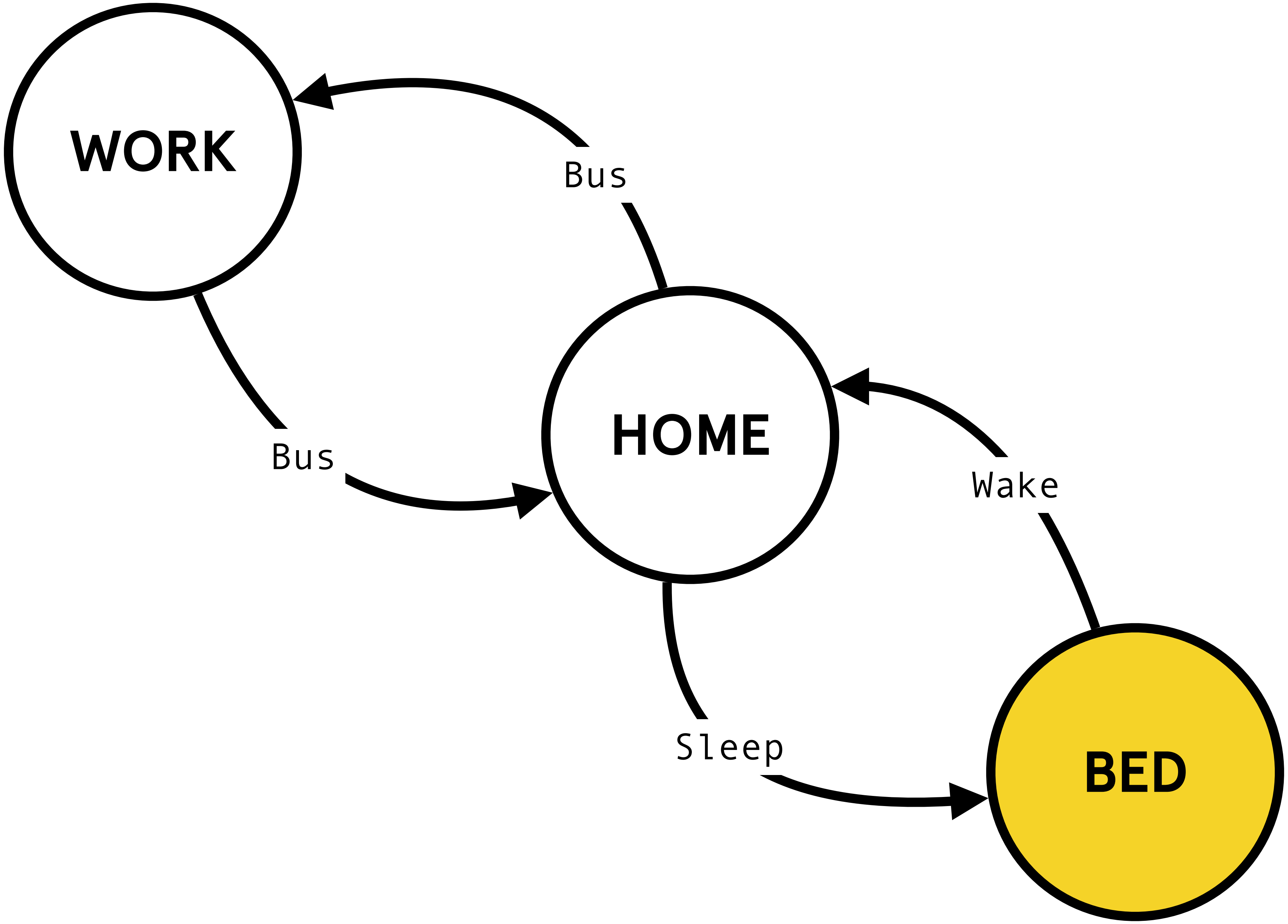
AT ANY GIVEN TIME, A STATE MACHINE CAN ONLY BE IN ONE STATE.

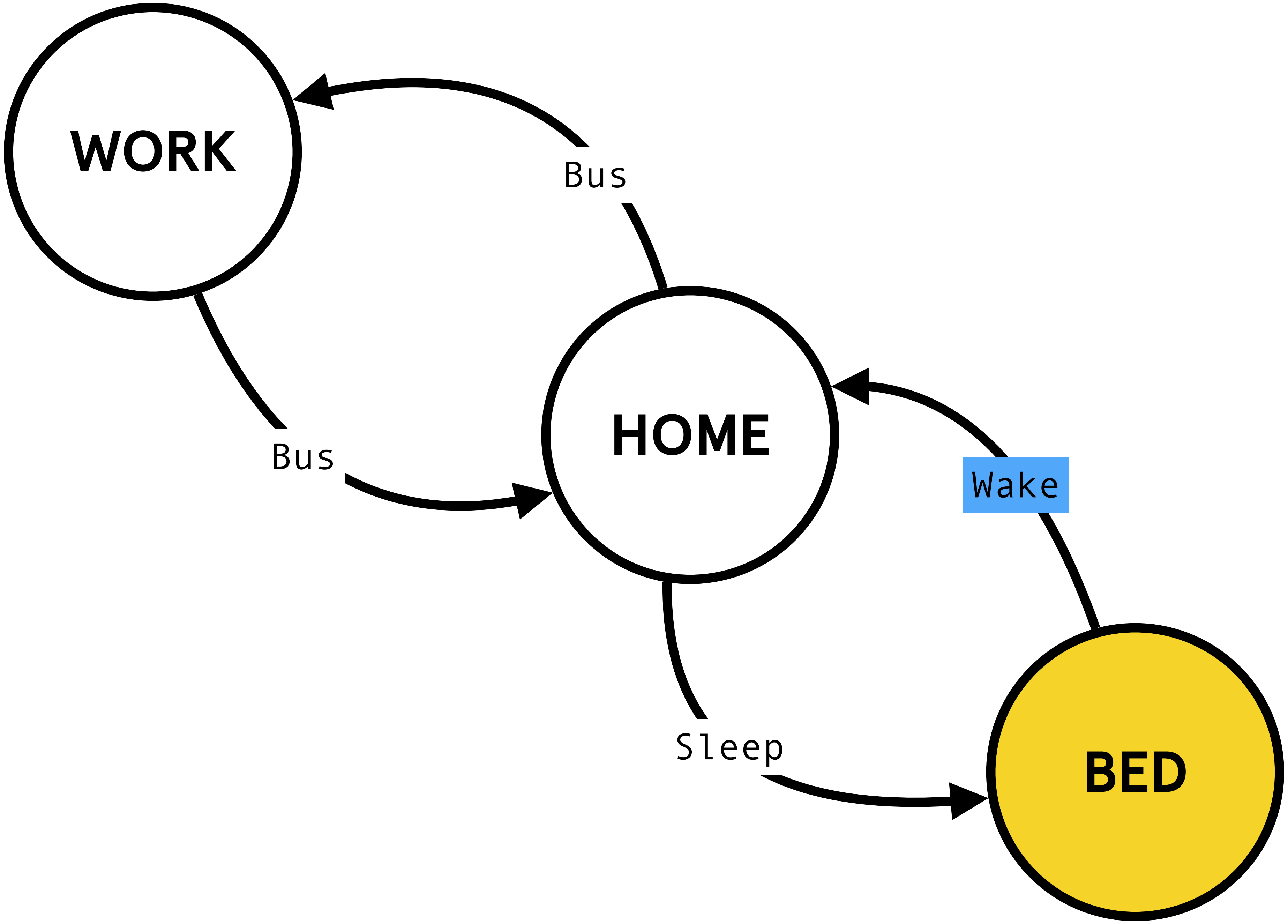
A TRANSITION IS TRIGGERED IN RESPONSE TO AN EXTERNAL INPUT OR SOME OTHER EVENT (LIKE A TIMER)

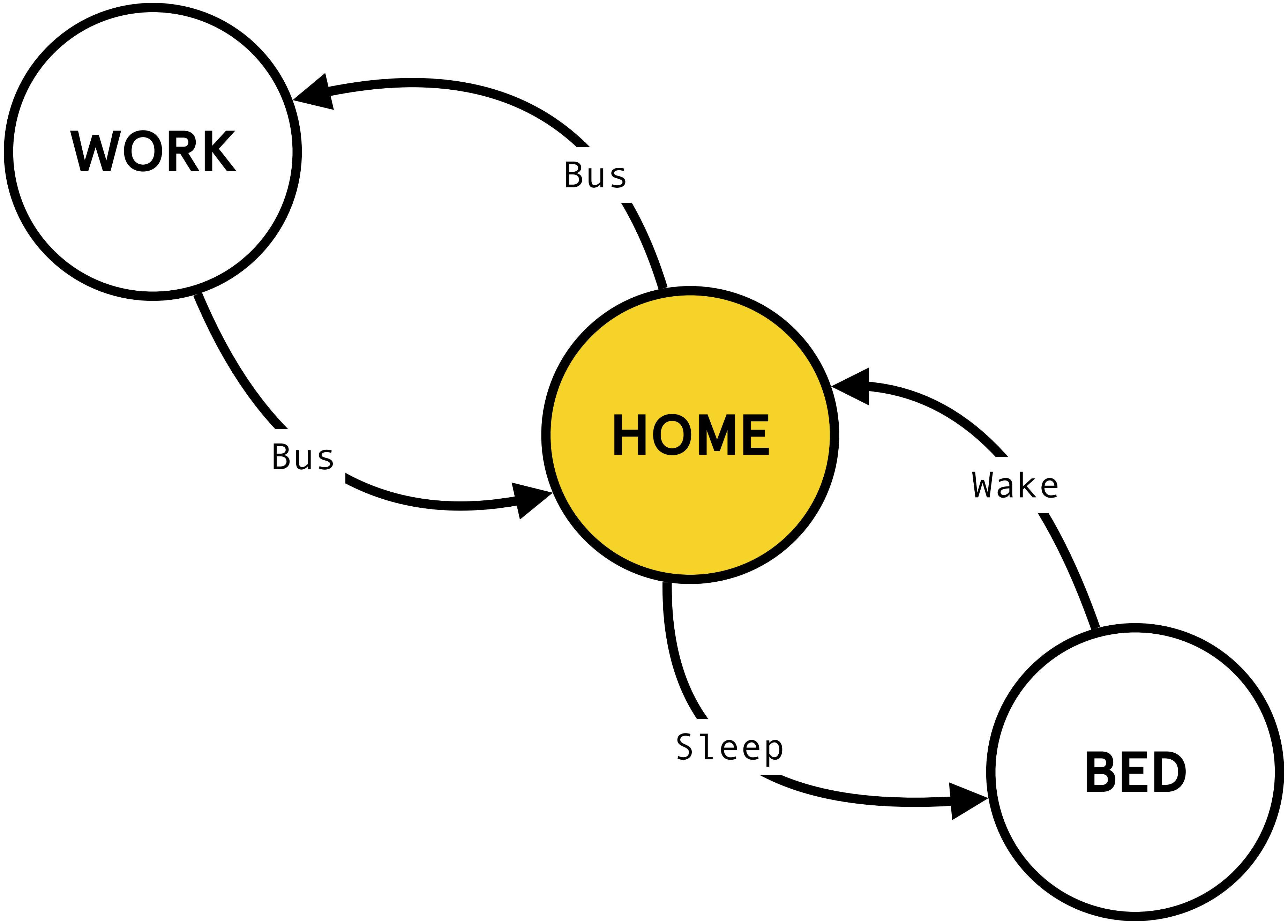
ORGANIZING PROGRAMS
USING STATE ALLOWS YOU TO
REASON MORE CLEARLY ABOUT
THE PARTS OF YOUR PROGRAM
AND HOW THEY INTERACT

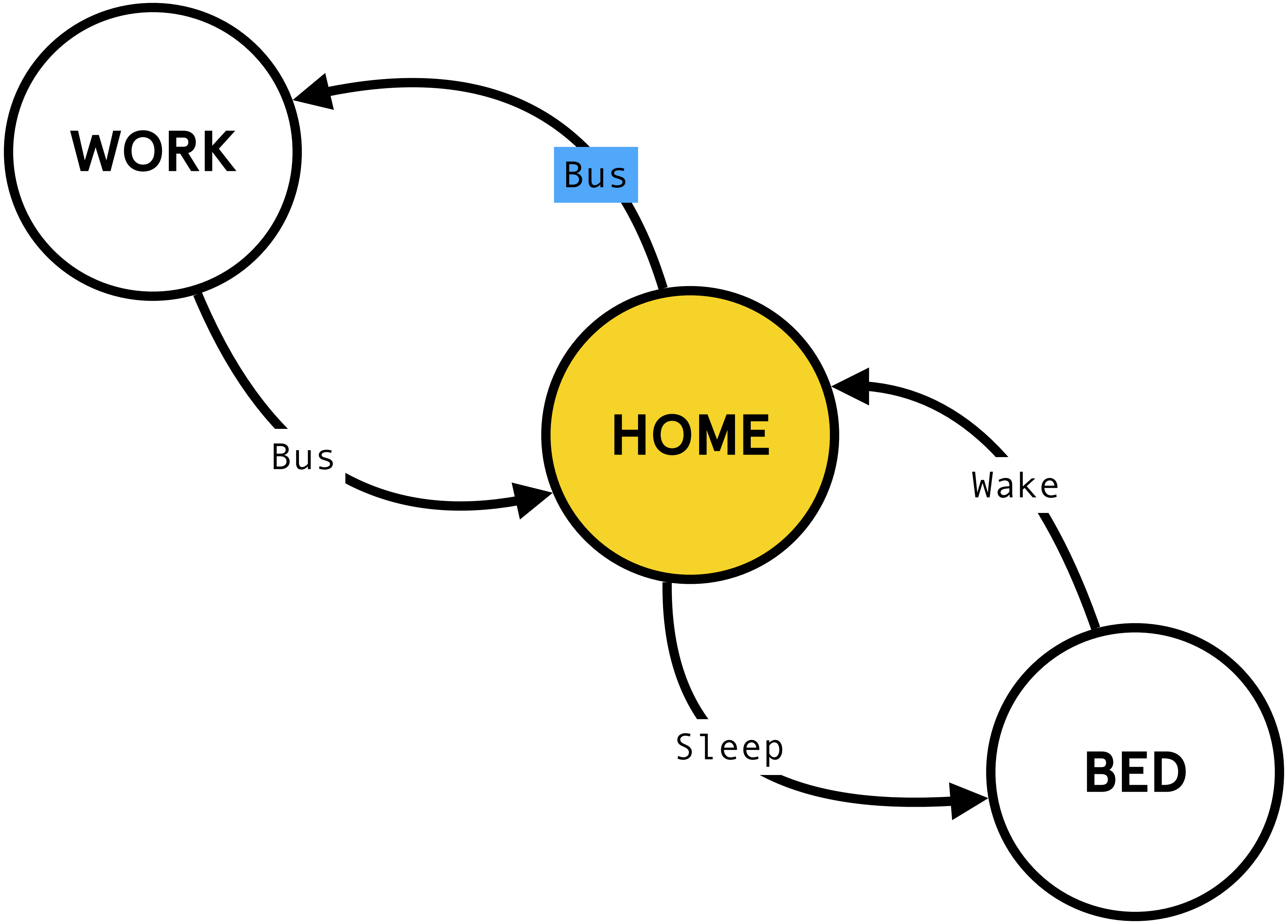


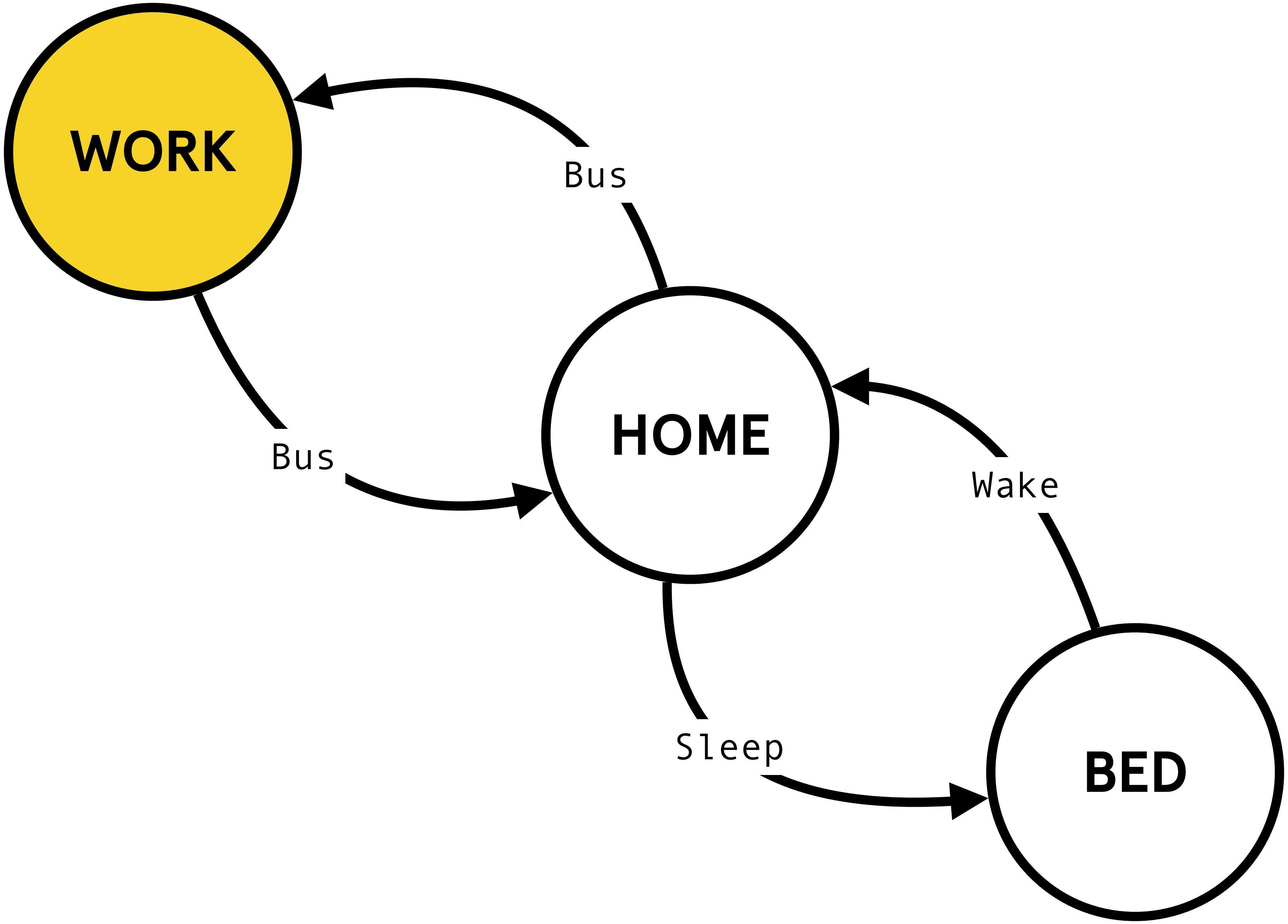
States are drawn as circles, and transitions are represented with lines connecting the circles.

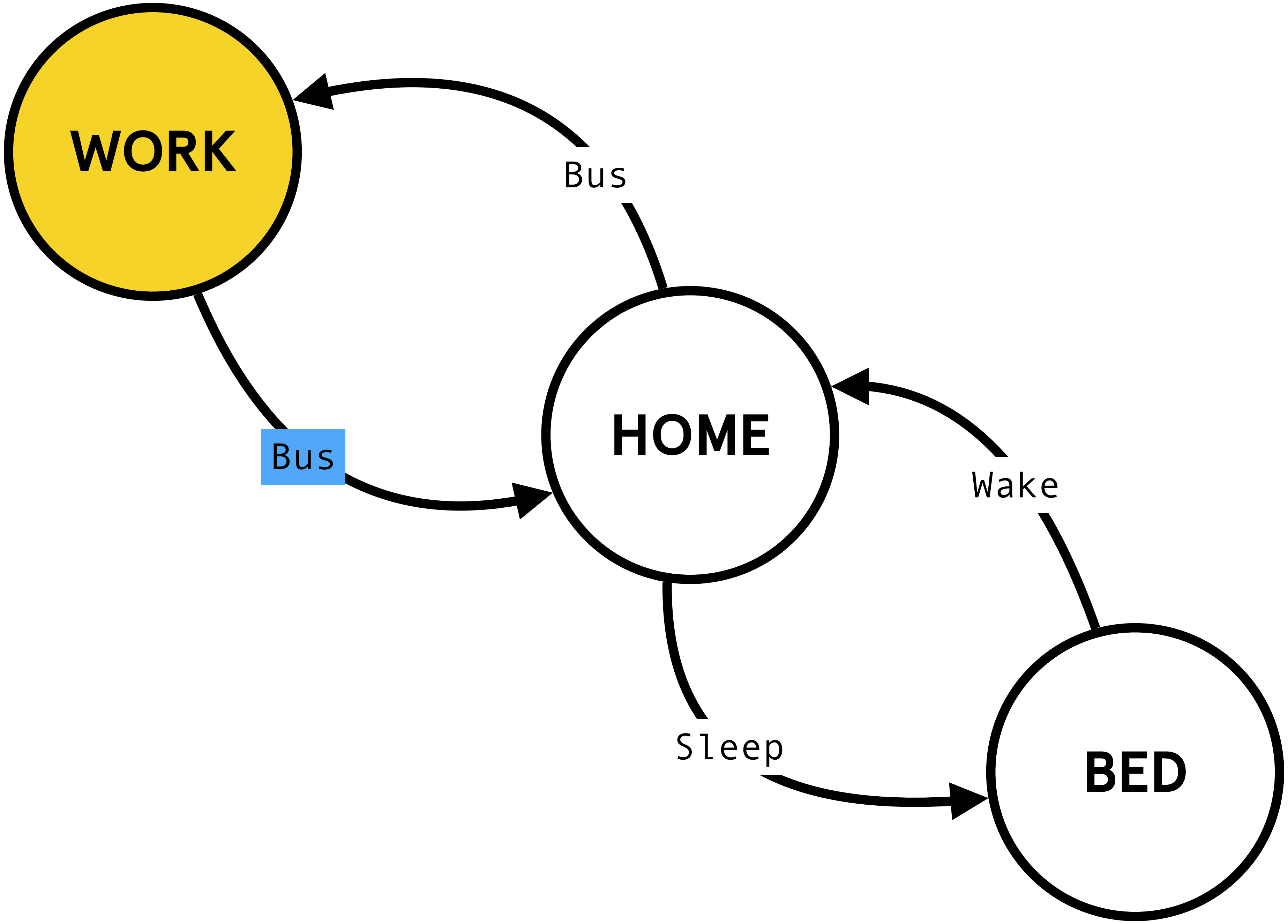

WORK
Bus
Bus
HOME
Wake
Sleep
BED

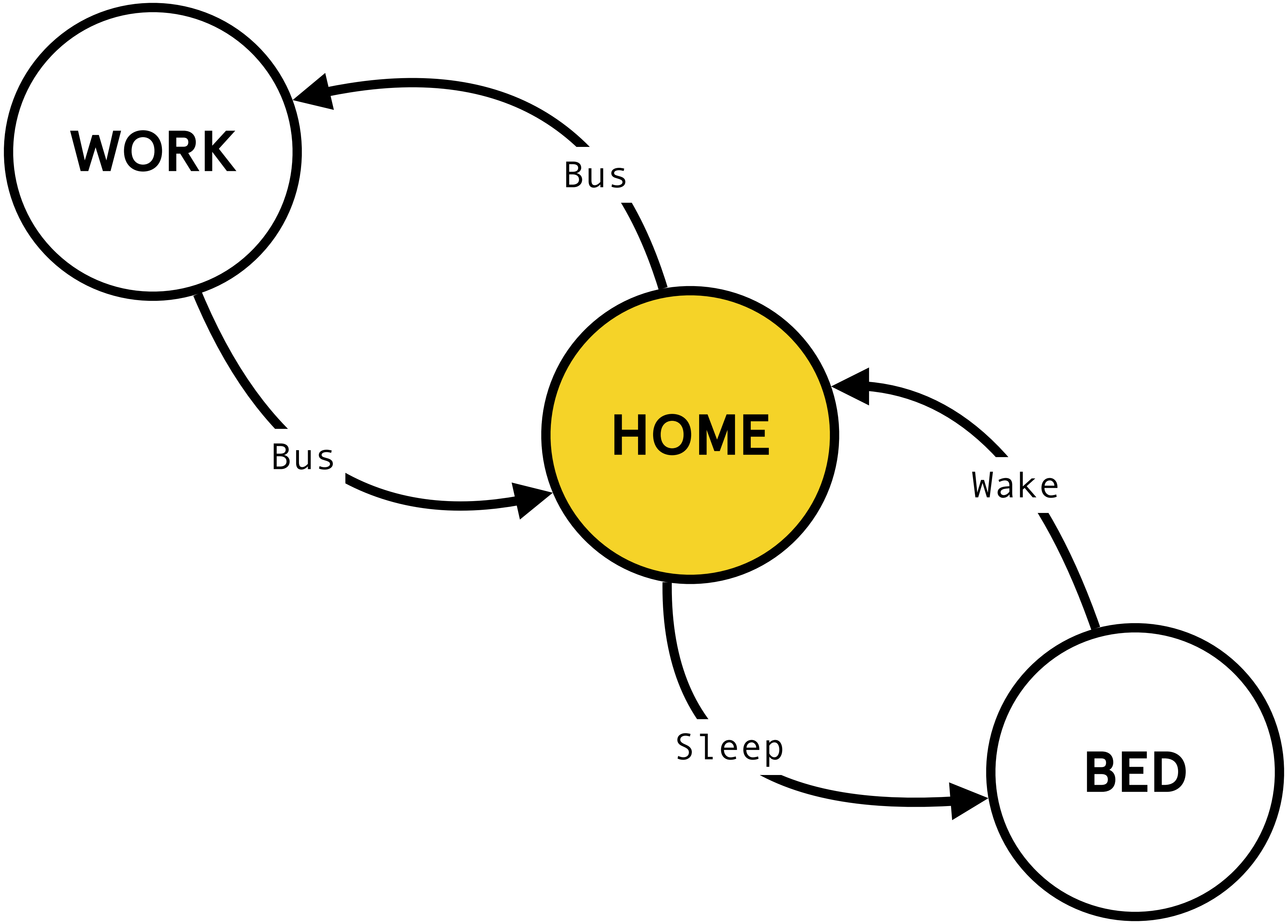

WORK
Bus
HOME
Bus
Wake
Sleep
BED

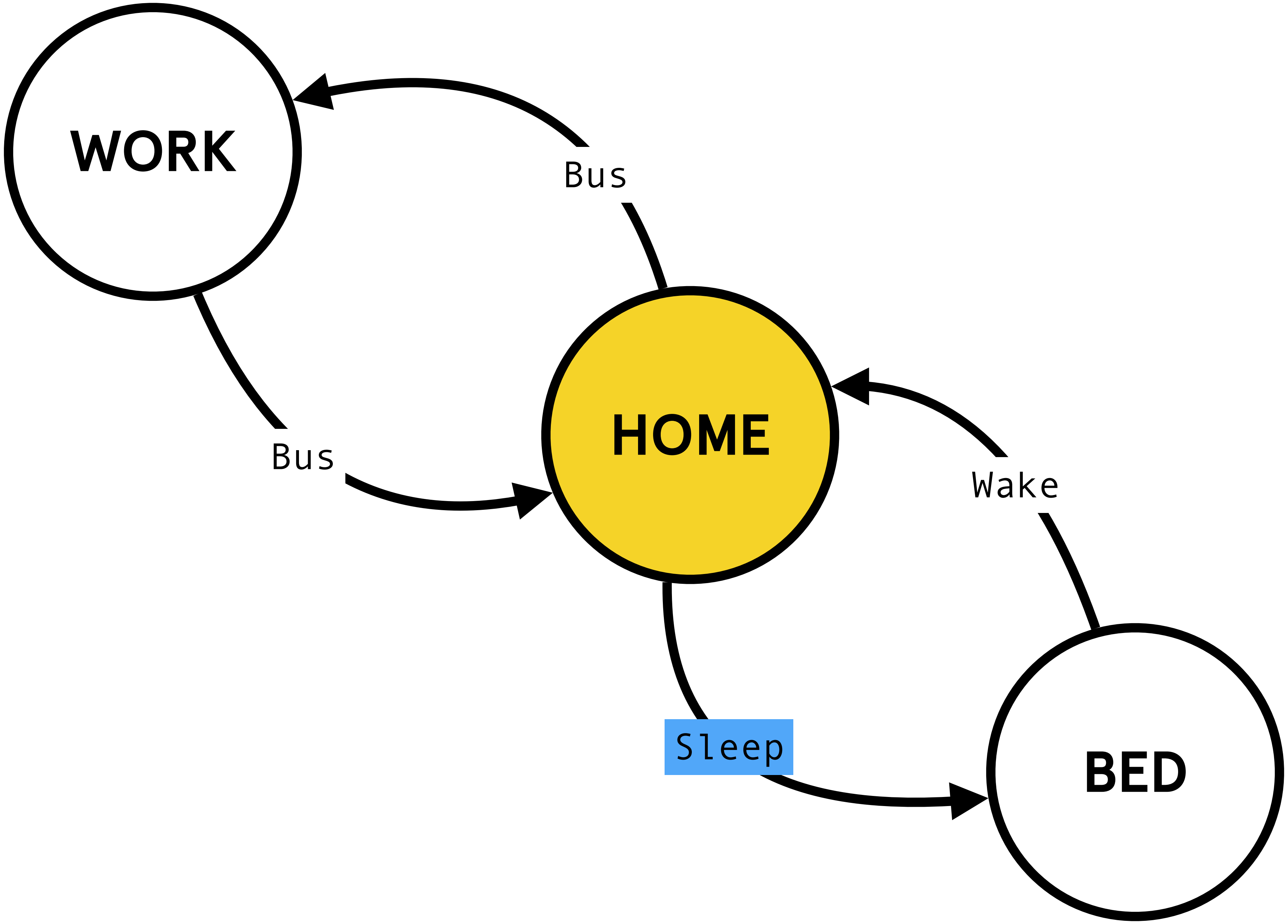

WORK
Bus
Bus
HOME
Wake
Sleep
BED

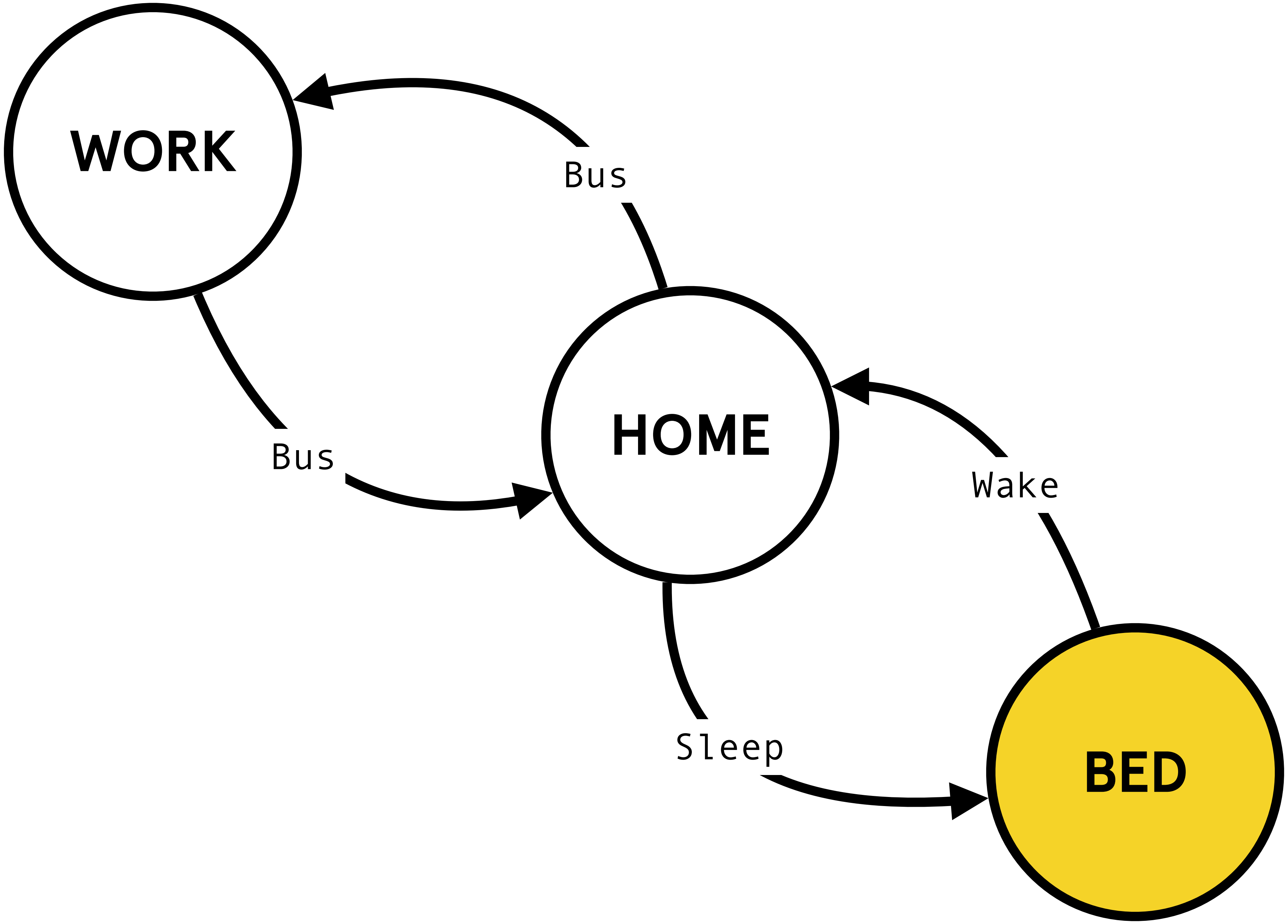

WORK
Bus
HOME
Bus
Wake
Sleep
BED


WORK
Bus
HOME
Bus
Wake
Sleep
BED


WORK
Bus
Bus
HOME
Wake
Sleep
BED


WORK
Bus
Bus
HOME
Wake
Sleep
BED


WORK
Bus
HOME
Bus
Wake
Sleep
BED

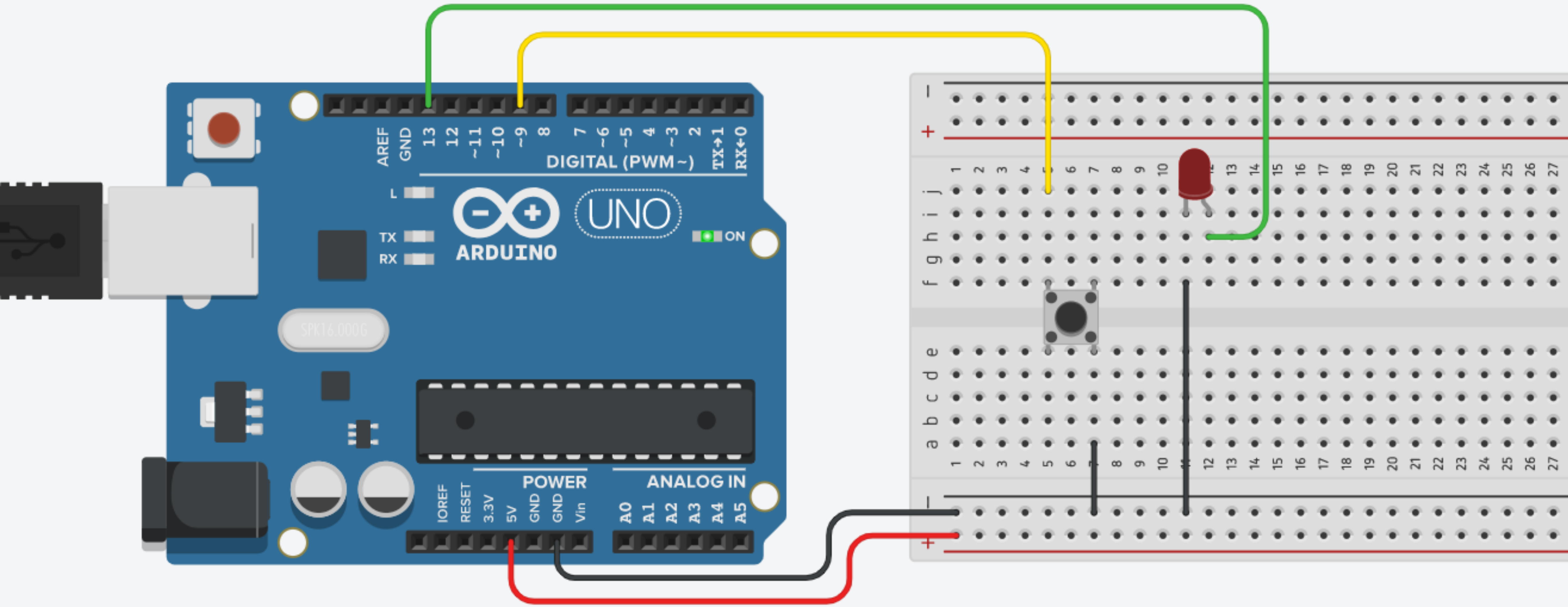


WORK
Bus
Bus
HOME
Wake
Sleep
BED

CONNECTING A BUTTON

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

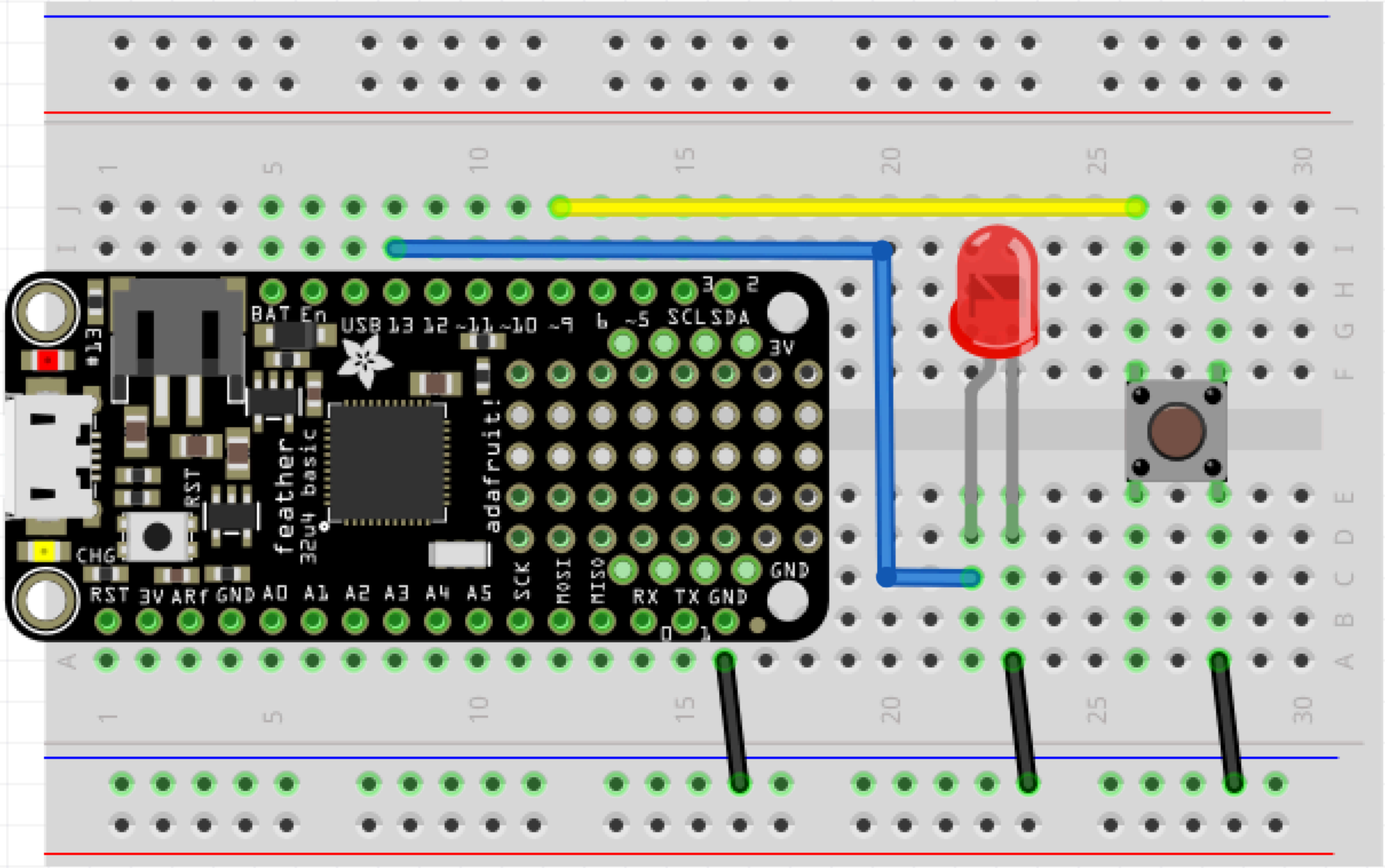
When the button is pressed, the connection to ground is completed so the pin is LOW.



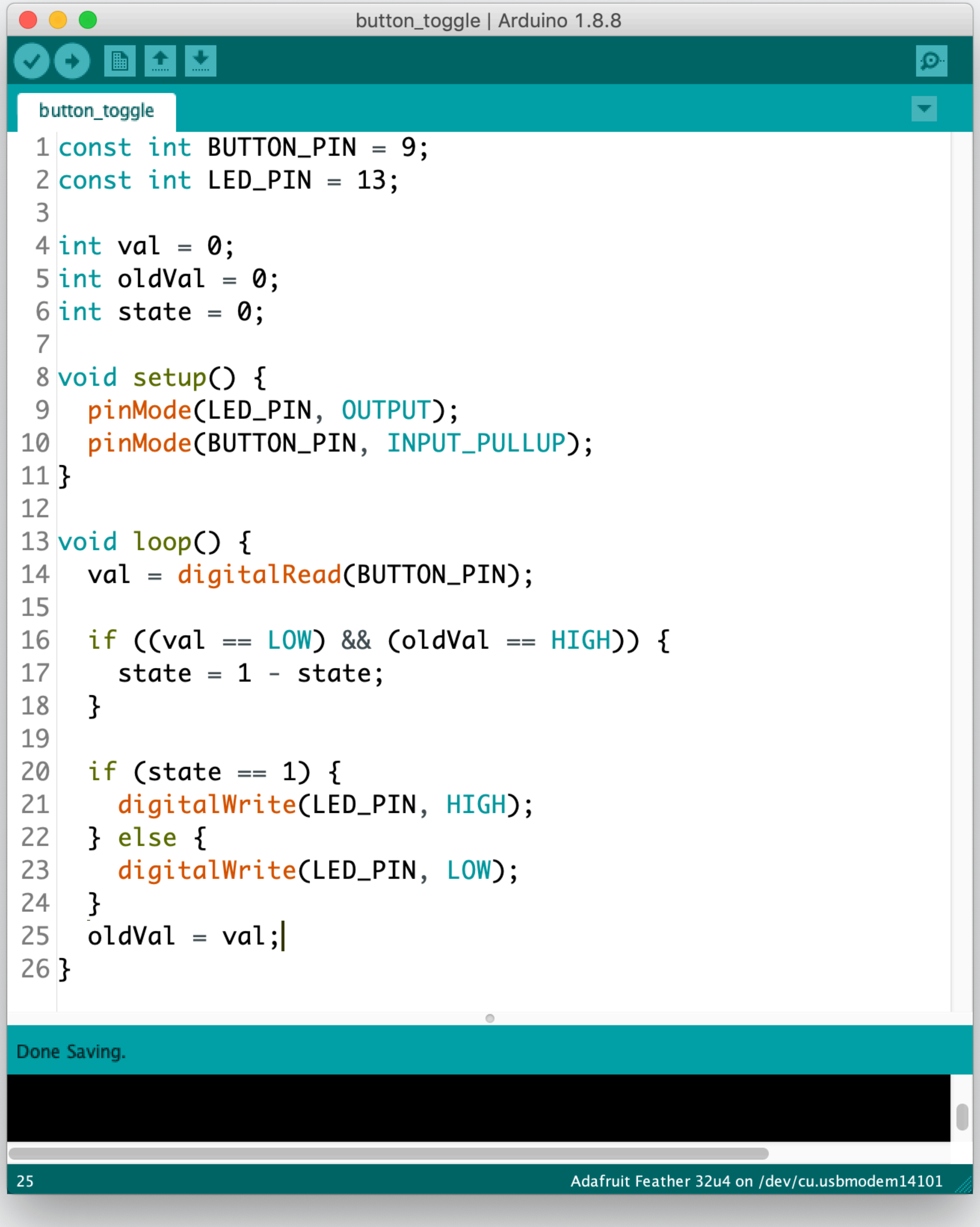
CONNECTING A BUTTON

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.



TOGGLE AN LED WITH A BUTTON



```
const int BUTTON_PIN = 9;
const int LED_PIN = 13;

int val = 0;
int oldVal = 0;
int state = 0;

void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop() {
  val = digitalRead(BUTTON_PIN);

  if ((val == LOW) && (oldVal == HIGH)) {
    state = 1 - state;
  }

  if (state == 1) {
    digitalWrite(LED_PIN, HIGH);
  } else {
    digitalWrite(LED_PIN, LOW);
  }
  oldVal = val;
}
```

Done Saving.

25 Adafruit Feather 32u4 on /dev/cu.usbmodem14101

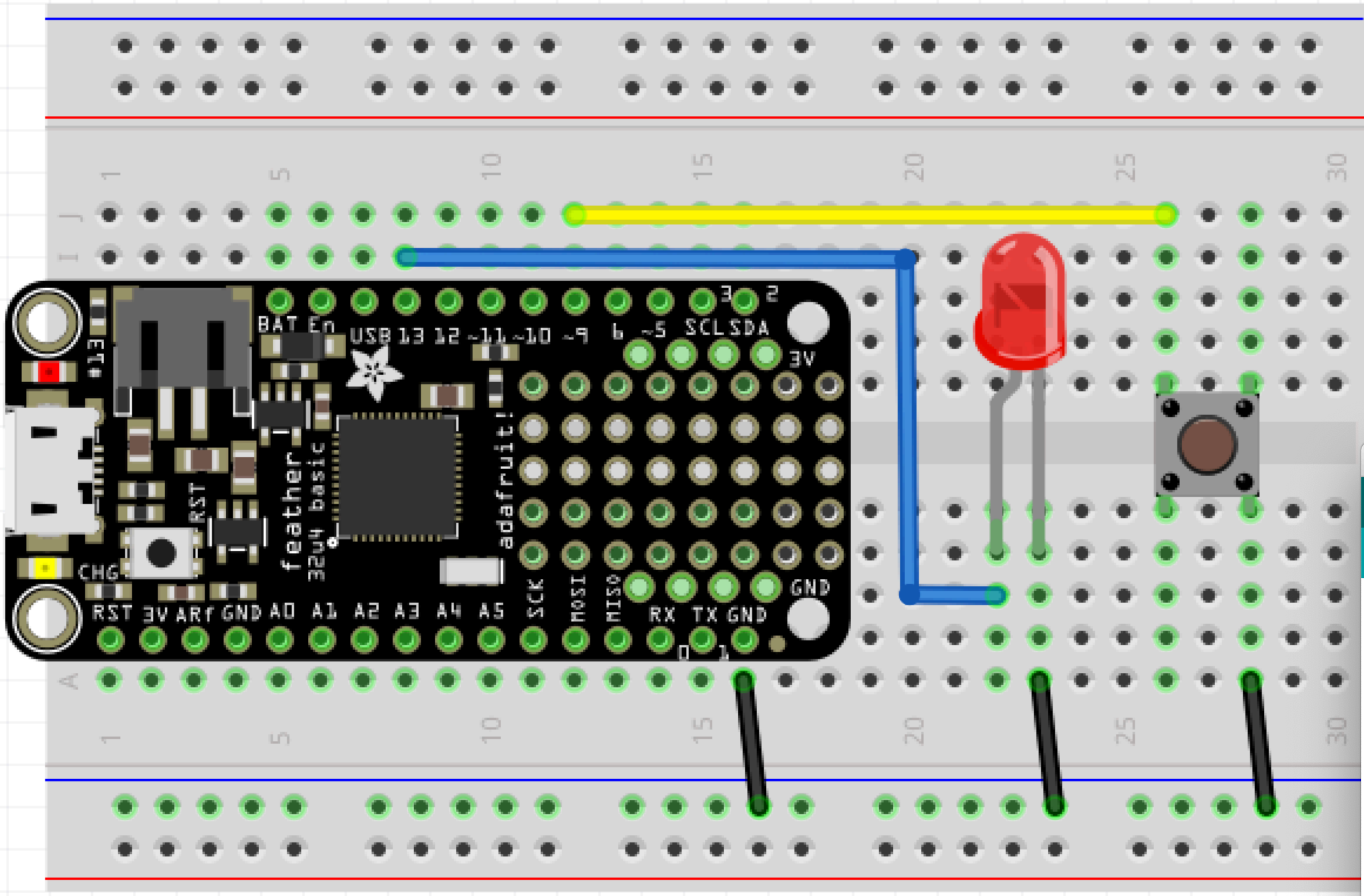
Only change the state of the LED when there is a change from LOW to HIGH on the button.

The “old_val” variable holds the state of the button the last time through loop.

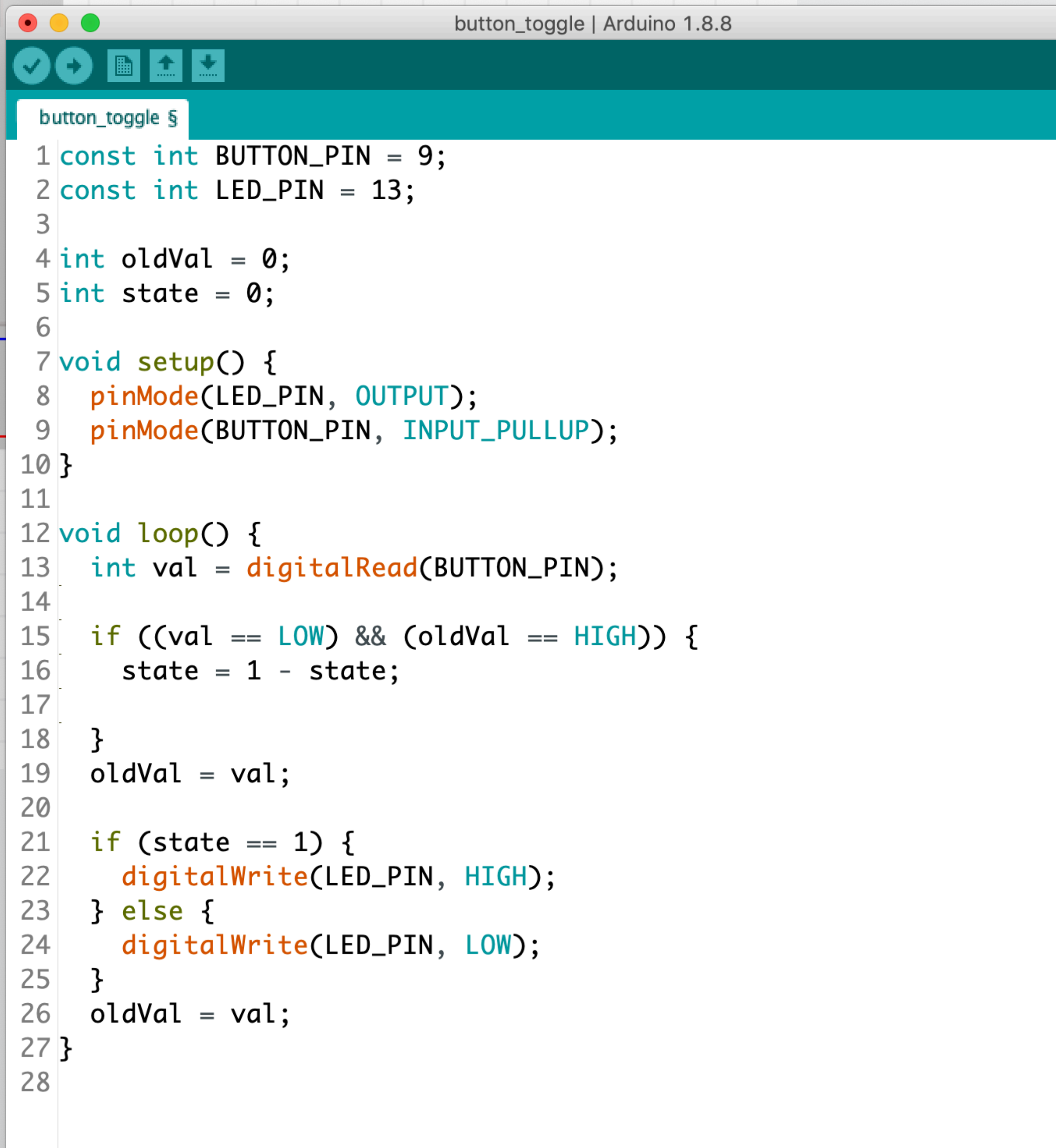
TOGGLE AN LED WITH A BUTTON

Only change the state of the LED when there is a change from LOW to HIGH on the button.

The “oldVal” variable holds the state of the button the last time through loop.



feather 32u4 basic
adafruit!
BAT En USB 13 12 ~11 ~10 ~9 6 ~5 SCL SDA
3 2
3V
GND
RST 3V ARf GND A0 A1 A2 A3 A4 A5 SCK MOSI MISO RX TX GND
0 1
#13
CHG
RST
1 5 10 15 20 25 30
J I H G F A

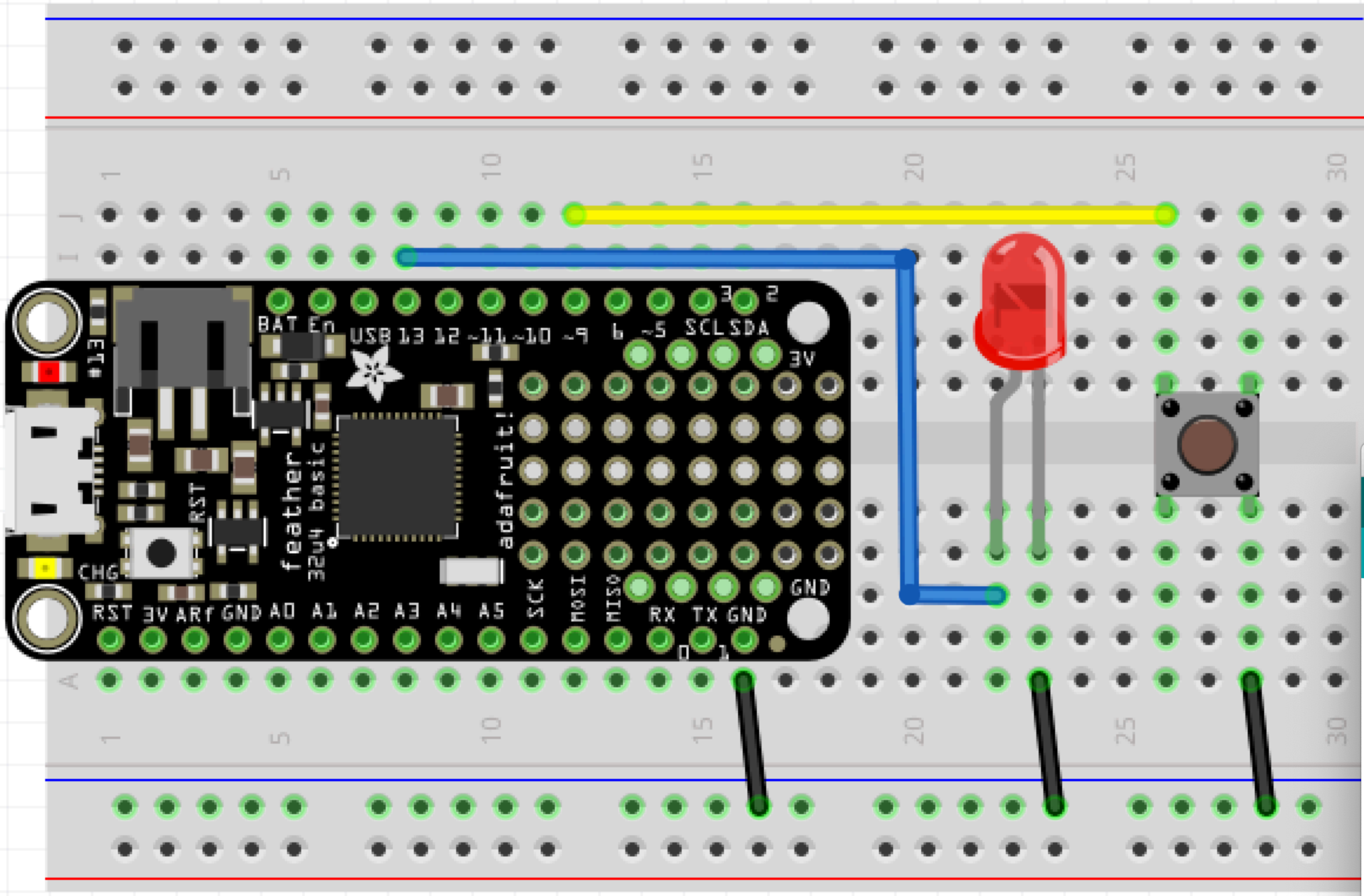


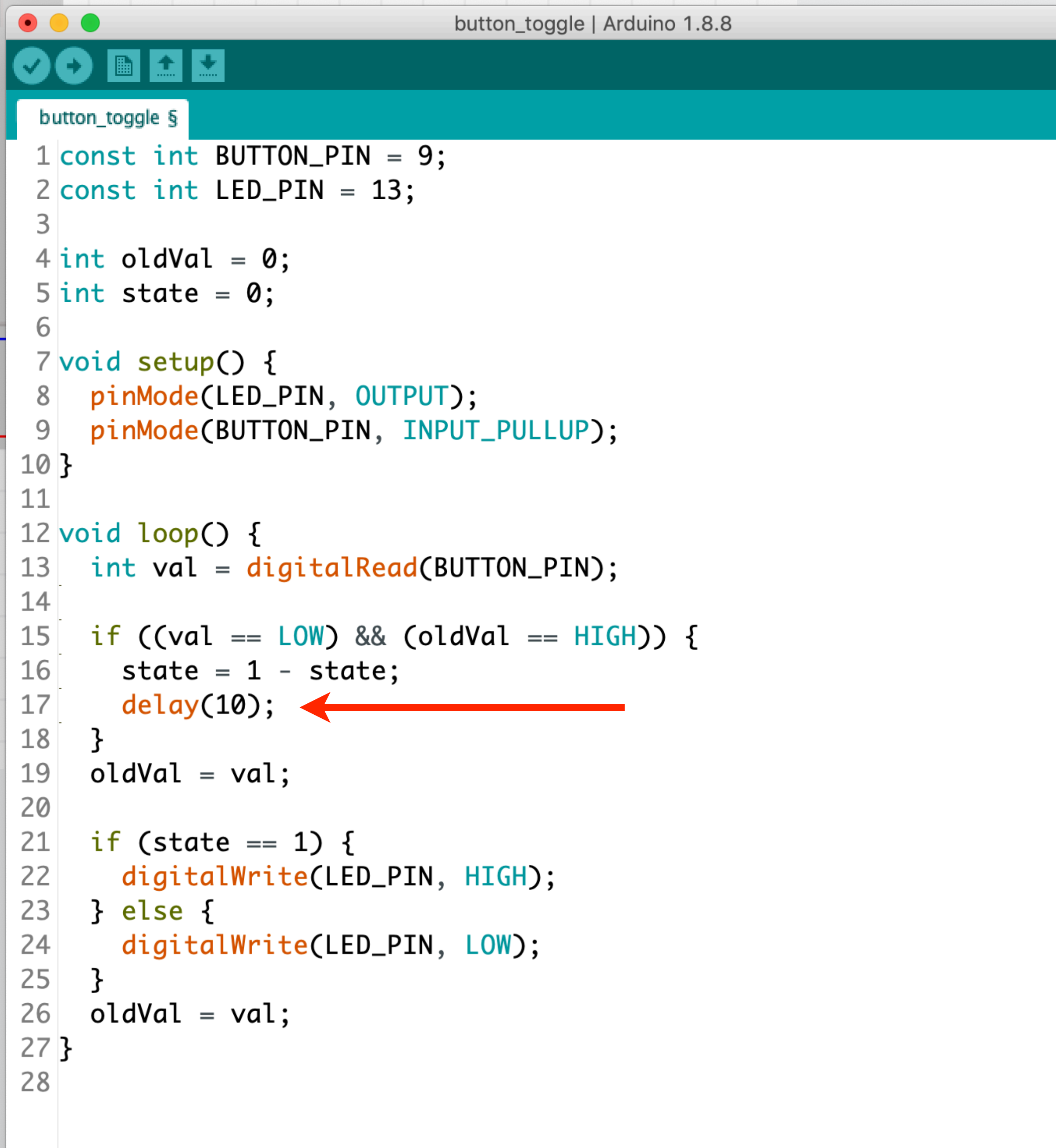
```
const int BUTTON_PIN = 9;
const int LED_PIN = 13;

int oldVal = 0;
int state = 0;

void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop() {
  int val = digitalRead(BUTTON_PIN);

  if ((val == LOW) && (oldVal == HIGH)) {
    state = 1 - state;

  }
  oldVal = val;

  if (state == 1) {
    digitalWrite(LED_PIN, HIGH);
  } else {
    digitalWrite(LED_PIN, LOW);
  }
  oldVal = val;
}

```

TOGGLE AN LED WITH A BUTTON + DEBOUNCING

Only change the state of the LED when there is a change from LOW to HIGH on the button.

The “oldVal” variable holds the state of the button the last time through loop.


#13
BAT En USB 13 12 ~11 ~10 ~9 6 ~5 SCL SDA
3V
feather 32u4 basic
adafruit!
RST
CHG
RST 3V ARf GND A0 A1 A2 A3 A4 A5 SCK MOSI MISO RX TX GND
GND
1
5
10
15
20
25
30
J
I
H
G
F
A



```
const int BUTTON_PIN = 9;
const int LED_PIN = 13;

int oldVal = 0;
int state = 0;

void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop() {
  int val = digitalRead(BUTTON_PIN);

  if ((val == LOW) && (oldVal == HIGH)) {
    state = 1 - state;
    delay(10);
  }
  oldVal = val;

  if (state == 1) {
    digitalWrite(LED_PIN, HIGH);
  } else {
    digitalWrite(LED_PIN, LOW);
  }
  oldVal = val;
}

```

Create a project with at least 3 buttons, and one analog input. The project should have 4 states and switch between states using the inputs.