Exercise 1-4

Jonas Pedersen, Søren Graae

2025.01.07

Exercise 1

You can't use 4 "bits" as each dot or dash doesn't represent a number like in binary, which is base 2. The value for `a` after the loop is 32.

```
#define ledPin    8
#define unitTime  250 // ms
#define dotTime   unitTime
#define dashTime  unitTime * 3
#define letterTime  unitTime * 3
#define wordTime    unitTime * 7

void dot() {
  digitalWrite(ledPin, HIGH);
  delay(dotTime);
  digitalWrite(ledPin, LOW);
}

void dash() {
  digitalWrite(ledPin, HIGH);
  delay(dashTime);
  digitalWrite(ledPin, LOW);
}

void S() {
  dot();
  delay(unitTime);
  dot();
  delay(unitTime);
  dot();
}

```

```
28 void O() {
29   dash();
30   delay(unitTime);
31   dash();
32   delay(unitTime);
33   dash();
34 }
35
36 void E() {
37   dot();
38   delay(unitTime);
39 }
40
41 void R() {
42   dot();
43   delay(unitTime);
44   dash();
45   delay(unitTime);
46   dot();
47 }
48
49 void N() {
50   dash();
51   delay(unitTime);
52   dot();
53 }
54
55 void outputLetter(char letter) {
56   switch (letter) {
57     case 'S': case 's':
58       S();
59       break;
60     case 'O': case 'o':
61       O();
62       break;
63     case 'E': case 'e':
64       E();
65       break;
66     case 'R': case 'r':
67       R();
68       break;
69     case 'N': case 'n':
70       N();
```

```
      break;
    default:
      break;
  }
}

void outputWord(char *str) {
  for(; *str; str++) { // Just a while-loop in disguise
    outputLetter(*str);
    delay(letterTime);
  }
}

void setup() {
  pinMode(ledPin, OUTPUT);

  char name[] = "Soeren";

  outputWord(name);
}

void loop() {

}
```

Exercise 2

$42\%5 = 2$.

Traffic Light

```
#define greenLED  8
#define yellowLED 9
#define redLED    10
#define delayTime 2000

void red() {
  Serial.println("STOP!");
  digitalWrite(redLED, HIGH);
  digitalWrite(greenLED, LOW);
  digitalWrite(yellowLED, LOW);
```

```
}

void redyellow() {
  Serial.println("GET READY!");
  digitalWrite(redLED, HIGH);
  digitalWrite(greenLED, LOW);
  digitalWrite(yellowLED, HIGH);
}

void green() {
  Serial.println("GO!");
  digitalWrite(redLED, LOW);
  digitalWrite(greenLED, HIGH);
  digitalWrite(yellowLED, LOW);
}

void yellow() {
  Serial.println("STOP-ISH!");
  digitalWrite(redLED, LOW);
  digitalWrite(greenLED, LOW);
  digitalWrite(yellowLED, HIGH);
}

void sequence() {
  red();
  delay(delayTime);
  redyellow();
  delay(delayTime);
  green();
  delay(delayTime);
  yellow();
  delay(delayTime);
}

void setup() {
  // put your setup code here, to run once:
  pinMode(greenLED, OUTPUT);
  pinMode(yellowLED, OUTPUT);
  pinMode(redLED, OUTPUT);

  Serial.begin(115200);
  red();
}
```

```
54
55 void loop() {
56   // put your main code here, to run repeatedly:
57   sequence();
58 }
```

Counter

```
1  #define LED0 8
2  #define LED1 9
3  #define LED2 10
4
5  void updateCounter(uint8_t value) {
6    Serial.print(value);
7    Serial.print(", ");
8    Serial.println((value >> 2) & 1);
9    Serial.print((value >> 1) & 1);
10   Serial.print(value & 1);
11
12   digitalWrite(LED0, value & 1);
13   digitalWrite(LED1, (value >> 1) & 1);
14   digitalWrite(LED2, (value >> 2) & 1);
15 }
16
17 uint8_t num;
18 void setup() {
19   // put your setup code here, to run once:
20   pinMode(LED0, OUTPUT);
21   pinMode(LED1, OUTPUT);
22   pinMode(LED2, OUTPUT);
23
24   Serial.begin(9600);
25   num = 0;
26 }
27
28 void loop() {
29   // put your main code here, to run repeatedly:
30   updateCounter(num);
31
32   num++;
33   num = num % 8;
34
```

```
  delay(1000);
}
```

Exercise 3

`INPUT` is not using an internal pull-up resistor whereas `INPUT_PULLUP` is. `!` is the `NOT` operator.

Push

Done without code, just connected LED to button through resistor.

Latching

Button press is checked in every `loop()` call, and a 200ms delay is added for debouncing if pressed.

```
const int ledPin = 9;        // Pin for the LED
const int buttonPin = 7;     // Pin for the button

int ledState = LOW;          // Current state of the LED
int lastButtonState = LOW;   // Tracks the last button
   state

void setup() {
  pinMode(ledPin, OUTPUT);           // Set LED pin as
     output
  pinMode(buttonPin, INPUT_PULLUP); // Set button pin as
      input with internal pull-up resistor

  Serial.begin(9600);               // Initialize serial
     communication
}

void loop() {
  // Read the current button state
  int buttonState = digitalRead(buttonPin);

  // Check if the button was pressed (from LOW to HIGH
     transition)
  if (buttonState == HIGH && lastButtonState == LOW) {
```

```
    ledState = !ledState;          // Toggle the LED
       state
    digitalWrite(ledPin, ledState); // Update the LED
    Serial.print("LED is now: ");
    Serial.println(ledState == HIGH ? "ON" : "OFF"); //
       Print LED state to Serial Monitor
    delay(200); // Debounce delay
  }

  lastButtonState = buttonState; // Update the last
     button state
}
```

Exercise 4

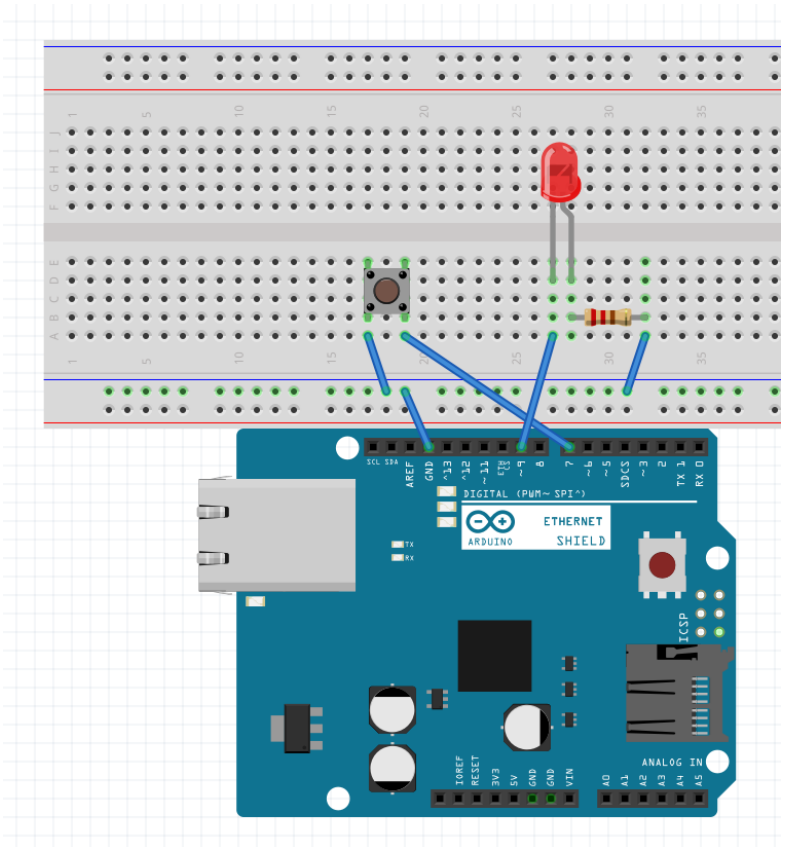

Figure 1: Ciruit for button program